On Stackelberg Pricing with Computationally Bounded Consumers[⋆]

Patrick Briest[⋆⋆], Martin Hoefer[⋆⋆⋆], Luciano Gualà[†], and Carmine Ventre[‡]

Abstract. In a Stackelberg pricing game a *leader* aims to set prices on a subset of a given collection of items, such as to maximize her revenue from a *follower* purchasing a feasible subset of the items. We focus on the case of computationally bounded followers who cannot optimize exactly over the range of all feasible subsets, but apply some publicly known algorithm to determine the set of items to purchase. This corresponds to general multi-dimensional pricing assuming that consumers cannot optimize over the full domain of their valuation functions but still aim to act rationally to the best of their ability.
We consider two versions of this novel type of Stackelberg pricing games. Assuming that items are weighted objects and the follower seeks to purchase a min-cost selection of objects of some minimum weight (the MIN-KNAPSACK problem) and uses a simple greedy 2-approximate algorithm, we show how an extension of the known single-price algorithm can be used to derive a polynomial-time $(2+\varepsilon)$-approximation algorithm for the leader's revenue maximization problem based on so-called *near-uniform* price assignments. We also prove the problem to be strongly NP-hard.
Considering the case that items are subsets of some ground set which the follower seeks to cover (the SET-COVER problem) via a standard primal-dual approach, we prove that near-uniform price assignments fail to yield a good approximation guarantee. However, in the special case of elements with frequency 2 (the VERTEX-COVER problem) it turns out that exact revenue maximization can be done in polynomial-time. This stands in sharp contrast to the fact that revenue maximization becomes APX-hard already for elements with frequency 3.

1 Introduction

The problem of *multi-dimensional pricing* consists of assigning revenue maximizing prices to a set of distinct items given information about the preferences of potential customers. A natural way to describe consumer preferences is via *valuation functions* that

[⋆] Work supported by DFG grant Kr 2332/1-2 within Emmy Noether Program.

[⋆⋆] Department of Computer Science, University of Paderborn, Germany. E-mail: patrick.briest@upb.de. Work done in part while the author was at Cornell University, supported by a scholarship of the German Academic Exchange Service (DAAD).

[⋆⋆⋆] Lehrstuhl Informatik I, RWTH Aachen University, Germany. E-mail: mhoefer@cs.rwth-aachen.de. The author is supported by UMIC Research Center at RWTH Aachen University.

[†] Dipartimento di Matematica, Universitá di Roma "Tor Vergata", Italy. E-mail: guala@mat.uniroma2.it.

[‡] Department of Computer Science, University of Liverpool, UK. E-mail: Carmine.Ventre@liverpool.ac.uk. Work also supported by EPSRC grant EP/F069502/1 and EU through IP AEOLUS.

map subsets of items to non-negative real numbers describing how much a set is valued by a certain customer. Given fixed item prices a rational customer acting according to *quasi-linear utilities* chooses to purchase the subset of items maximizing her *utility*, i.e., the difference between her value for the set and the sum of prices of items contained in it. It is known that general multi-dimensional pricing with unlimited supply of each item allows polynomial-time approximation algorithms achieving ratios that are logarithmic in the number of customers or linear in the number of distinct items via considering only *single-price solutions* [4, 8]. Lower bounds of the same order of magnitude have also been proven for approximation guarantees in both parameterizations [7, 14].

For the known algorithmic results it is sufficient that valuation functions can be accessed via *demand oracles*, i.e., consumers are treated as black boxes that can answer the question: "Given some vector of prices, which set of items would you choose to buy?" But how much can it help us to have more detailed information about the structure of customer preferences? If the number of customers is large, unfortunately, the answer is "not at all" as follows readily from the known lower bounds, which hold for special cases of the problem in which the exact preferences can be elicited via demand queries. However, the situation is different when the number of distinct customers is small.

In a 2-player *Stackelberg Pricing Game*, named after the underlying market model due to Heinrich Freiherr von Stackelberg [27], we are given a collection of items, some of which have fixed-costs. A so-called *leader* may assign prices to the remaining items. A *follower* then purchases a *feasible* set of items of minimum cost and pays the leader for the priceable items in the set. This problem is equivalent to multi-dimensional pricing with a single follower if the follower's feasible sets are unrestricted. However, a standard assumption when considering Stackelberg games is that the follower has to be able to optimize in polynomial time over her feasible sets. As an example, think of items as edges in a graph and the follower's feasible sets as all possible paths connecting some vertices s and t. For some kinds of followers, e.g., buying a min-cost vertex cover in a bipartite graph, it has been shown that improved approximation guarantees are possible [8].

In this paper we initiate the study of a closely related question: What if the follower is unable to exactly optimize over her feasible sets, because the problem is computationally hard, but is still guaranteed to act rationally to the best of her ability? More formally, we will assume that when prices have been fixed the follower applies a publicly known approximation algorithm to find a near-optimal feasible set of items to purchase. This assumption is quite reasonable when followers are actually software agents with known implementation details. To the best of our knowledge, this is the first analysis of multi-dimensional pricing with follower preferences that are neither *single-minded* or *unit-demand*, nor expressible as exact optimization over the full domain of the valuation function. Before describing our results in detail, we review some important related work and introduce the notation used throughout the paper.

Related Work. Algorithmic aspects of multi-dimensional pricing problems, which are important in the context of pure optimization as well as the design of revenue-maximizing auction mechanisms [3], were first studied by Aggarwal et al. [1] and Guruswami et al. [20]. Subsequently, quite a number of improved algorithmic results for special cases of the problem [2, 9, 12, 13, 15, 16, 21] and complexity theoretic lower bounds [7, 14, 10] have been derived.

Our introductory example of shortest-path Stackelberg pricing was first introduced by Labbé et al. [23], who derived a bi-level LP formulation of the problem and proved NP-hardness. Subsequently, Roch et al. [24] presented a first polynomial time approximation algorithm with a provable (logarithmic) approximation guarantee. More recently, Cardinal et al. [11] investigated the corresponding minimum spanning tree game, proving that this version of the problem is APX-hard and that the very simple *single-price algorithm* achieves a logarithmic approximation guarantee. Finally, Briest et al. [8] extended the analysis of the single-price algorithm to Stackelberg pricing in general. Stackelberg pricing in which the follower purchases a single-source shortest path tree has been considered in [6].

Stackelberg pricing can also be considered with objectives other than revenue maximization. When prices are tolls on network arcs the problem of congestion minimization has received considerable attention. Karakostas and Kolliopoulos [22], Fleischer, Jain and Mahdian [17], Fleischer [18] and Swamy [25] show how tolls can be used to enforce low-congestion Nash equilibria in selfish network routing games.

Preliminaries. In this paper we consider games falling in the following general class. There are two players in the game, one *leader* and one *follower*. There is also a set of items $\mathcal{I}$ that is partitioned into fixed-cost items $\mathcal{F}$ and priceable items $\mathcal{P}$. Each fixed-cost item $i \in \mathcal{F}$ has a fixed-cost $c(i) \geq 0$. For each priceable item $i \in \mathcal{P}$ the leader can specify a price $p(i) \geq 0$. The follower has a set $\mathcal{S} \subset 2^{\mathcal{I}}$ of *feasible subsets* and is interested in buying some subset in $\mathcal{S}$. The cost of a subset $S \in \mathcal{S}$ is given by the cost of fixed-cost items and the price of priceable items: $cost(S) = \sum_{i \in S \cap \mathcal{F}} c(i) + \sum_{i \in S \cap \mathcal{P}} p(i)$. The *revenue* of the leader from subset S is given by the prices of the priceable items that are included in S, that is, $r(S) = \sum_{i \in S \cap \mathcal{P}} p(i)$. We let $S_A(p)$ be the feasible subset in $\mathcal{S}$ chosen by the follower when she uses polynomial-time algorithm A given prices p. Naturally, the follower would like A to return the minimum-cost subset in $\mathcal{S}$, but this could be a hard task to solve in polynomial-time. We capture this intuition by making no assumption on optimality of the algorithm: A can return a suboptimal subset in $\mathcal{S}$. Our interest is to find the pricing function p^* for the leader that generates maximum revenue when the follower uses algorithm A, i.e., $p^* \in \arg\max_p r(S_A(p))$. We denote this maximum revenue by r^*. To guarantee that the revenue is bounded and the optimization problem is non-trivial, we assume that there is at least one feasible subset that is composed only of fixed-cost items and that the follower algorithm outputs it under certain circumstances. Towards this aim, we further assume that for each priceable item there is a threshold price above which no subset including it will be output by the follower algorithm. This last assumption holds for every follower algorithm with bounded approximation ratio.

In the above class of games, we will consider the MIN-KNAPSACK pricing problem and the SET-COVER pricing problem. In the knapsack pricing problem, the set of items is a set of weighted objects $\mathcal{O}$. A subset of $\mathcal{O}$ is feasible if the total weight of the object comprising it is at least a given bound W. We will refer to the revenue optimization problem for the knapsack pricing problem by STACKKP. In the set cover pricing problem, following our terminology, given some ground set to be covered, every item corresponds to a given subset of the ground set. A subset of items is feasible for the follower whenever it covers all the elements of the ground set. We will refer to the revenue optimization problem for the set cover pricing problem by STACKSC and denote the special case of vertex cover pricing by STACKVC.

Contributions. The focus of this paper are followers applying approximation algorithms that are (i) structurally simple and (ii) sufficiently suboptimal to ensure that revenue maximization has to take into account the algorithm's exact structure. We first consider STACKKP and assume that the follower uses the well known greedy algorithm (see Section 2) to compute a 2-approximate solution to the minimization version of the knapsack problem she needs to solve. Even though structurally quite simple, this problem seems to capture many of the fundamental problems of Stackelberg pricing with computationally limited followers.

We show that in this case a careful adaptation of the known single-price strategy termed *near-uniform pricing*, which essentially assigns a single price to a subset of items and removes the remaining ones from the market by assigning a sufficiently high price (see Section 2 for a formal definition), can be used to approximate optimal revenue in most of the solution space. Adding some fairly standard enumeration techniques we are able to derive a polynomial-time $(2 + \varepsilon)$-approximation for the revenue maximization problem. The main technical difficulty lies in the fact that our analysis needs to argue about the exact computation done by the follower for a given price vector rather than using some global optimality condition. We point out that our algorithm is best possible among all algorithms based on near-uniform price assignments. Finally, we show that the revenue maximization problem in this setting is strongly NP-hard.

We then turn our attention to STACKSC and assume that the follower is using the primal-dual schema based approximation algorithm (see Section 3) to find a selection of sets to purchase. We view the problem in its equivalent formulation of VERTEX-COVER in hypergraphs and start by investigating the special case of regular VERTEX-COVER in standard graphs. We prove that while near-uniform price assignments cannot achieve better than logarithmic approximation guarantee in this case, exploiting the algorithm's structure nevertheless allows for exact revenue maximization in polynomial time. To the best of our knowledge, this is the second example of polynomial-time revenue maximization being possible for a class of Stackelberg pricing games. Previously, it was shown that games with a follower purchasing a min-cost vertex cover in a bipartite graph in the special case that all priceable vertices are located on one side of the bipartition allows for polynomial-time revenue maximization [8]. It would be very interesting to see whether there is a deeper connection between these two problems. Turning to general hypergraphs it turns out that revenue maximization (STACKSC) becomes hard already with edges of cardinality 3 (or elements of frequency 3 in the SET-COVER view) and is APX-hard in general. This is quite surprising given that the follower's primal-dual algorithm achieves approximation guarantee f for any frequency f, i.e., the approximation complexity of the underlying problem scales quite smoothly. We also argue that in this general case neither near-uniform price vectors nor our algorithm from the VERTEX-COVER case can guarantee any sub-exponential approximation ratio.

Knowing the Follower's Algorithm. A central assumption in our analysis is that the leader has full knowledge of the follower's algorithm. This assumption might appear quite strong. It turns out, however, that this non-black-box attitude on the leader's side is necessary to achieve any reasonable approximation guarantees. Suppose the leader only knows the approximation guarantee of the follower's algorithm A and is given black-box-access to it, but no specific details are revealed. In this case it is easy to argue that for both StackKP and StackVC no algorithm can achieve a finite approximation guarantee. We omit the simple proof of the following fact due to space limitations.

Proposition 1. *For any constant $\rho > 1$, there are instances of* StackKP *and* StackVC *in which no leader's algorithm can achieve a finite approximation guarantee given only information about ρ and black-box-access to the follower's ρ-approximation algorithm.*

Similarly, it is necessary to assume that the follower decides on the algorithm to be used in advance. If the follower is allowed to choose the algorithm (from a known set of alternatives) once the leader has set the prices, then an impossibility result similar to the one above applies.

2 Knapsack Pricing

In the Min-Knapsack problem we are given a set $\mathcal{O}$ of n *objects*, some of them with fixed cost and some priceable. Each object $o \in \mathcal{O}$ has weight $w(o) \in \mathbb{N}$ and we are given an integer weight bound W. Following the general framework given above, each subset X of $\mathcal{O}$ has a cost which is defined as the sum of the cost of the fixed-cost objects in X and the prices of the priceable objects in X. The follower wants to purchase a set of objects of minimum cost whose weight is at least W. We assume that the follower uses the standard greedy algorithm outlined below to find an approximation of such a minimum-cost set.

The Follower's Algorithm. An object's cost-efficiency (below referred to as efficiency for brevity) is defined as $\phi(o) = c(o)/w(o)$ or $\phi(o) = p(o)/w(o)$, depending on whether it is fixed-cost or priceable. Algorithm 1 below proceeds as follows. First, order all objects by non-decreasing efficiency (breaking ties by decreasing weight). Then add objects to the knapsack in this order. If an object makes the weight of the solution it completes at least W, remember this (feasible) solution and discard the object. Finally, return the best solution found. Note that we assume that ties are broken according to decreasing weight, i.e., larger objects are considered first given identical efficiency. This is a natural tie breaking rule, as it aims at minimizing the *overlap* of objects that exceed the knapsack's remaining capacity when they are considered.

Transforming the Optimal Solution. Let p^* be the optimal price assignment and $\mathcal{P}^*$ be the set of priceable objects that are selected by Algorithm 1 given these prices.

The key ingredient for our approximation algorithm for knapsack pricing is the observation that price assignments that result in a large number of priceable objects being bought by the follower can be approximated by almost uniform price assignments at the expense of reducing overall revenue by no more than a constant factor.

Definition 1. *We call a price assignment p* near-uniform*, if there exists a single efficiency $\phi > 0$, such that $p(o) = w(o) \cdot \phi$ or $p(o) = +\infty$ for every $o \in \mathcal{P}$.*

Algorithm 1: The greedy approximation algorithm for MIN-KNAPSACK

```
Let $o_1, o_2, \dots, o_n$ be the objects ordered by non-decreasing efficiency, i.e.,
$\phi(o_1) \leq \cdots \leq \phi(o_n)$.
$X \leftarrow Y \leftarrow \emptyset$.
$c_Y \leftarrow +\infty$.
for $i = 1, \dots, n$ do
    $X \leftarrow X \cup \{o_i\}$.
    if $w(X) \geq W$ then
        if $cost(X) < c_Y$ then
            $Y \leftarrow X$.
            $c_Y \leftarrow cost(X)$.
        $X \leftarrow X \setminus \{o_i\}$.
Return $Y$.
```

We call an object b *blocking*, if the weight of the current solution exceeds W when it is added by the greedy algorithm. Let $\mathcal{B} = \{b_1, \dots, b_l\}$ denote the blocking objects with $\phi(b_1) \leq \cdots \leq \phi(b_l)$. Since every blocking object corresponds to a unique solution checked by the greedy algorithm, our approach to relating the algorithm's behavior on two different price vectors is to relate the sets of blocking objects in both cases.

Theorem 1. *Let p^* be the optimal price assignment for a given* STACKKP *instance and r^* the resulting revenue. Furthermore, assume that given prices p^* the follower purchases at least $k \in \mathbb{N}$ priceable objects, i.e., $|\mathcal{P}^*| \geq k$. Then there exists a near-uniform price assignment $\tilde{p}$ with revenue at least $r^*(k-1)/(2k)$.*

Proof. Define $w^* = \sum_{o \in \mathcal{P}^*} w(o)$, $r^* = \sum_{o \in \mathcal{P}^*} p^*(o)$ and let $\phi^*_{ave} = r^*/w^*$. Let c^* denote the total cost of the solution bought by the follower given prices p^*. We define a near-uniform price assignment $\tilde{p}$ by $\tilde{p}(o) = (1/2)w(o)\phi^*_{ave}$ for all $o \in \mathcal{P}^*$, and $\tilde{p}(o) = +\infty$ else.

There is a one-to-one correspondence between the blocking objects $\mathcal{B} = \{b_1, \dots, b_l\}$ given prices p^* and solutions checked by the greedy algorithm. Because of the fact that blocking objects do not influence the set of solutions checked by the greedy algorithm (beyond the one they belong to themselves), it is w.l.o.g. to assume that there is at most a single priceable blocking object given prices p^*, and if it exists it belongs to $\mathcal{P}^*$.

Proving the claimed revenue guarantee for $\tilde{p}$ consists of two parts. First, we show that with prices $\tilde{p}$ the blocking objects of efficiency less than $\phi^*_{ave}/2$ are still blocking with with prices $\tilde{p}$ and their corresponding solutions have cost greater than $c^* - (1/2)r^*$. We then show that among the solutions with blocking objects of efficiency greater or equal than $\phi^*_{ave}/2$ there exist some with cost at most $c^* - (1/2)r^*$ and the cheapest among these contains priceable objects of value at least $r^*(k-1)/(2k)$.

We first deal with blocking objects with efficiency $\phi(b_j) < \phi^*_{ave}/2$ and argue that they remain blocking. Consider b_j with $\phi(b_j) = c(b_j)/w(b_j) < \phi^*_{ave}/2$. Since we have at most one blocking priceable object (of efficiency at least ϕ^*_{ave}), b_j must be fixed-cost. Given prices p^*, let $s_{<j}$ be the remaining unfilled capacity of the knapsack when b_j is considered and let $w_{<j}$ and $r_{<j}$ denote the weight of priceable objects in

the knapsack at this point and their total price, respectively. Similarly, let $w_{>j}$ and $r_{>j}$ denote the weight and price of priceable objects from $\mathcal{P}^*$ that are considered after b_j and let $\phi_{>j} = r_{>j}/w_{>j}$ be their average efficiency. Finally, define $f_{>j}$ to be the total cost of fixed-cost objects in the optimal solution with higher efficiency than b_j.

We are going to argue that moving all priceable objects to a position behind b_j (as happens with near-uniform pricing $\tilde{p}$) will not cause b_j to become non-blocking. First observe that $r_{<j} < w_{<j}\phi^*_{ave}/2 \leq w^*\phi^*_{ave}/2 = (1/2)r^*$ and, thus, we have that $w_{>j}\phi_{>j} > (1/2)r^*$. By the fact that b_j is not part of the cheapest solution, we know that $f_{>j} + w_{>j}\phi_{>j} \leq c(b_j)$. It follows that

$$\begin{aligned} w_{>j}\phi_{>j} &\leq c(b_j) - f_{>j} \leq c(b_j) - (s_{<j} - w_{>j})\phi(b_j), \text{ since } f_{>j} \geq (s_{<j} - w_{>j})\phi(b_j) \\ &= (w(b_j) - s_{<j} + w_{>j})\phi(b_j). \end{aligned}$$

Assume now that b_j becomes non-blocking if we remove total weight $w_{<j}$ from the knapsack, i.e., $w(b_j) - s_{<j} \leq w_{<j}$. We may then write that

$$\phi_{>j} \leq \left(1 + \frac{w(b_j) - s_{<j}}{w_{>j}}\right)\phi(b_j) \leq \left(1 + \frac{w_{<j}}{w_{>j}}\right)\phi(b_j).$$

Using that $w_{>j}\phi_{>j} > (1/2)r^*$ and $\phi(b_j) < (1/2)\phi^*_{ave}$ we obtain $\frac{1}{2}r^* < w_{>j}\phi_{>j} \leq (w_{>j} + w_{<j})\phi(b_j) < \frac{1}{2}w^*\phi^*_{ave} = \frac{1}{2}r^*$, a contradiction. We conclude that b_j remains a blocking element.

Note, that the fact that blocking objects of efficiency at most $\phi^*_{ave}/2$ remain blocking also implies that no non-blocking objects with efficiency below $\phi^*_{ave}/2$ can become blocking, since under prices $\tilde{p}$ the knapsack contains less total weight when such an object is considered. Also observe that given prices $\tilde{p}$, the fact that $r_{<j} < (1/2)r^*$ implies that every solution found by the greedy algorithm with a blocking object b_j of efficiency less than $\phi^*_{ave}/2$ has total cost greater than $c^* - (1/2)r^*$.

We continue by considering blocking objects with efficiency $\phi(b_j) \geq \phi^*_{ave}/2$. If $\mathcal{P}^*$ did not contain a blocking object given prices p^*, then changing prices to $\tilde{p}$ does not change the set of blocking objects of efficiency at least $\phi^*_{ave}/2$. To see this, note that with prices $\tilde{p}$ the knapsack's remaining capacity is smaller than before at any point, so no previously blocking object can become non-blocking. On the other hand, all the non-blocking objects left enough room to pack the objects from $\mathcal{P}^*$, so changing the order in which they are considered cannot create new blocking objects either. It follows that all solutions found by the greedy algorithm with blocking objects of efficiency at least $\phi^*_{ave}/2$ contain priceable objects with a total price of $r^*/2$. The previously cheapest solution is still found and has cost $c^* - (1/2)r^*$.

If $\mathcal{P}^*$ did contain a blocking element with prices p^*, but does not given prices $\tilde{p}$, we consider two cases. If the previously optimal solution is still found by the greedy algorithm, the same argumentation as above guarantees revenue $r^*/2$. If it is not, it must be the case that some previously non-blocking element has now become blocking. Consider the first such element b_j. Since all objects in the knapsack at the time b_j is considered and b_j itself were part of the cheapest solution given prices p^*, we have again found a solution of total cost at most $c^* - (1/2)r^*$. Since all priceable objects are in the knapsack at this point and remain in it, a lower bound of $r^*/2$ on the revenue follows immediately.

Finally, assume that $\mathcal{P}^*$ contains a blocking element with both prices p^* and $\tilde{p}$. In this case, the solution with priceable blocking object is guaranteed to have cost at most $c^* - (1/2)r^*$. By the algorithm's tie-breaking rule every solution with a blocking object of efficiency larger than $\phi^*_{ave}/2$ found at a later time contains all but the smallest object from $\mathcal{P}^*$ and, by the fact that $|\mathcal{P}^*| \geq k$, contains priceable objects of total value at least $(1 - 1/k)w^*(1/2)\phi^*_{ave} = r^*(k-1)/(2k)$. □

Algorithm 2: A $(2+\varepsilon)$-approximation algorithm for STACKKP

```
Let $1$ be the minimum, $\Phi$ the maximum efficiency of fixed-cost objects, $W$ the knapsack
capacity, $\sqrt{1+\varepsilon/2} - 1 \geq \delta > 0$, $r_{max} \leftarrow 0$.
Choose $k \geq (2(2+\varepsilon))/(2+\varepsilon-2(1+\delta)^2)$ and let
$\Lambda = \{\phi(o) \mid o \in \mathcal{F}\} \cup \{(1+\delta)^j \mid j = 0, \ldots, \lceil \log_{1+\delta} \Phi \rceil\}$.
foreach set $S \subseteq \mathcal{P}$ of priceable objects with $|S| \leq k-1$ do
    foreach price assignment p with $p(o)/w(o) \in \Lambda$ for all $o \in S$ and $p(o) = +\infty$ else
    do
        Let $r$ be the resulting revenue.
        $r_{max} \leftarrow \max\{r_{max}, r\}$.
foreach $0 \leq i \leq \lceil \log_{1+\delta} \Phi \rceil$ do
    Let $\phi_i = (1+\delta)^i$.
    foreach $0 \leq j \leq \lceil \log_{1+\delta} W \rceil$ do
        Let $S$ be a set of at least $k$ priceable objects with total weight between $(1+\delta)^j$
        and $(1+\delta)^{j+1}$, if such set exists, else $S = \emptyset$.
        Set $p(o) = w(o)\phi_i$ for all $o \in S$ and $p(o) = +\infty$ for all other priceable objects.
        Let $r$ be the resulting revenue.
        $r_{max} \leftarrow \max\{r_{max}, r\}$.
Return $r_{max}$.
```

Approximation Algorithm for Revenue Maximization and Hardness Result. We are now ready to present our $(2+\varepsilon)$-approximate algorithm for STACKKP. Algorithm 2 proceeds in two stages. First it checks for some given constant $k \in \mathbb{N}$ all possible price assignments to sets of at most $k-1$ priceable objects. Then for each possible weight w, it finds a set with k or more priceable objects with total weight (roughly) w (if such a set exists), and considers all near-uniform price assignments to that set.

Note that in both stages, checking all possible efficiencies cannot be done in polynomial time. Instead we restrict our attention to the efficiencies of the fixed-cost objects plus all powers of $(1+\delta)$ for some $\delta > 0$ to guarantee that we efficiently find a near-optimal price assignment in which all objects are considered by the greedy algorithm in the right order. Similarly, in the second stage we cannot enumerate all possible weights for the objects in our near-uniform price assignments, but again have to settle for powers of $(1+\delta)$. The proof of Theorem 2 is omitted due to space limitations.

Theorem 2. *Algorithm 2 computes in polynomial time a $(2+\varepsilon)$-approximation for* STACKKP.

We mention that we can provide an instance showing that the above analysis is essentially tight. Finally, it turns out that revenue maximization against a follower using Algorithm 1 is strongly NP-hard. Details are left for the full version of this paper.

Theorem 3. STACKKP *is strongly NP-hard.*

3 Set Cover Pricing

In this section we consider the vertex and set cover pricing problems. In the simplest case the follower wants to purchase a minimum vertex cover of a graph $G = (V, E)$. Let $V = \mathcal{P} \cup \mathcal{F}$, where as above $\mathcal{F}$ and $\mathcal{P}$ denote the set of fixed-cost and priceable vertices respectively. More generally, for set cover we consider an equivalent formulation of vertex cover in hypergraphs. Namely, given the universe and its subsets of the set cover instance, we define an hypergraph where the vertices are the universe subsets and hyperedges are the elements of the universe. In this case there can be hyperedges in E connecting more than two vertices. We assume that the follower uses the standard primal-dual algorithm to find an approximation to the minimum vertex cover.

The Follower's Algorithm. The algorithm iteratively considers uncovered edges and raises budgets at the incident vertices until (at least) one of the vertices becomes tight. The algorithm is given for the case of regular vertex cover as Algorithm 3, for more details see [26, Ch. 15].

Algorithm 3: Primal-dual algorithm of the follower

```
$\gamma_e \leftarrow 0$ for all edges $e \in E$
For all $v \in V$ let $c'(v) \leftarrow c(v)$ if $v \in \mathcal{F}$ and $c'(v) \leftarrow p(v)$ otherwise
Fix an order of edges $e_1, \dots, e_m$
for $i = 1, \dots, m$ do
    Let $e_i = (u, v)$ and $\sigma_u \leftarrow c'(u) - \sum_{e\,:\,e=(u,v')} \gamma_e$
    $\sigma_v \leftarrow c'(v) - \sum_{e\,:\,e=(u',v)} \gamma_e$
    $\gamma_e \leftarrow \min(\sigma_u, \sigma_v)$
Add every vertex $v$ with $c'(v) = \sum_{e\,:\,e=(u,v)} \gamma_e$ to the cover.
```

Revenue Maximization for Vertex Cover. At first, let us consider the approximation ratio of single-price and near-uniform price assignments. For the latter, as from Definition 1, we have a $\phi \in \mathbb{R}_0^+$ and $p(v) \in \{\phi, \infty\}$ for all $v \in \mathcal{P}$. It turns out that for these pricings there is a simple logarithmic lower bound (proof omitted).

Theorem 4. *Single-price and near-uniform price assignments yield an approximation factor of $\Omega(\log n)$ for* STACKVC*, where n is the number of priceable vertices.*

Instead, we present a natural greedy algorithm to compute optimal prices for the seller. It simulates a run of the primal-dual algorithm and raises prices of the priceable vertices in the same manner as the dual budgets γ are raised by the follower. In this way the algorithm greedily tries to sell a vertex to the follower as soon as she is willing to pay for it.

Algorithm 4: Greedy pricing for STACKVC

```
$\gamma_e \leftarrow 0$ for all edges $e \in E$
for each edge $e = e_1, e_2, \ldots, e_m$ in the order of the follower do
    if $e = (u, v)$ is incident to a priceable vertex $v$ then
        $\gamma_e \leftarrow c(u) - \sum_{e':e'=(u,v')} \gamma_{e'}$
        $p(v) \leftarrow \sum_{e':e'=(u',v)} \gamma_{e'}$
    else
        $\sigma_u \leftarrow c(u) - \sum_{e':e'=(u,v')} \gamma_{e'}$
        $\sigma_v \leftarrow c(v) - \sum_{e':e'=(u',v)} \gamma_{e'}$
        $\gamma_e \leftarrow \min(\sigma_u, \sigma_v)$
```

Theorem 5. *Algorithm 4 solves* STACKVC *in polynomial time.*

Proof. Consider an optimum pricing p^*, which yields a strictly larger revenue than the *greedy pricing* p_g computed with our algorithm. For such a given set of prices p^*, we denote by γ_e^* the dual contribution of edge e, which we call the *budget* of edge e. This contribution is the result of applying the follower's algorithm to the instance using p^*. We restrict our attention to an optimal pricing p^* for which the follower purchases all priceable vertices. The existence of such a pricing follows from the next lemma (proof omitted).

Lemma 1. *Given any pricing p, there is a pricing p_l with $p_l(v) \leq p(v)$ for all $v \in \mathcal{P}$, for which the leader obtains at least as much revenue as for p and the follower purchases every vertex $v \in \mathcal{P}$.*

Now consider the smallest i', for which edge $e' = e_{i'} = (u, v')$ has $\gamma_{e'}^* \neq \gamma_{e'}$. It is easy to note that this edge must be incident to a priceable vertex v', and the difference in budgets is a result from setting different prices. As in both p^* and p_g all vertices are bought by the follower, it must be the case that $p^*(v) < p_g(v)$, and hence $\gamma_{e'}^* < \gamma_{e'}$. We now compare the revenue of pricing p^* to a pricing p' with $p'(v) = p^*(v)$ for every vertex $v \neq v'$, and for which $p'(v') = p^*(v') + \gamma_{e'} - \gamma_{e'}^*$. In p' the budgets γ_e' are equivalent to γ_e (i.e., the budgets generated by the greedy pricing p_g) for every edge $e_1, \ldots, e_{i'-1}$ and also $e_{i'} = e'$.

We call $\delta^j(v) = \sum_{e_i:e_i=(v,u),i\leq j} \gamma_{e_i}' - \gamma_{e_i}^*$ the *reservation* that is created by p^* at vertex v at the end of processing edge e_j. The budget of e' is raised to a smaller amount in p^* than in p', so after processing e' there is positive reservation at the other endpoint u of e', i.e. $\delta^{i'}(u) = \delta^{i'}(v')$ at vertex u. No other vertex except u and v' has reservation at this point, so it holds that $\sum_{v\neq v'} |\delta^{i'}(v)| \leq \delta^{i'}(v')$. This will be our invariant, and in the following we prove it for the remaining edges $j > i'$ and the remaining iterations of the algorithm with pricing p' (proof omitted).

Lemma 2. *For any iteration $j \geq i'$ we have that $\sum_{v\neq v'} |\delta^j(v)| \leq \delta^{i'}(v')$.*

The lemma above shows that the sum of absolute values of reservation at all vertices except v' at any point during the remaining runs of the followers algorithm is at most the initial reservation $\delta^{i'}(v')$. Note that v' is bought in both cases. In p^* all priceable vertices are bought by the follower, but this might not be true for p' and vertices $v \neq v'$. p' might lose revenue there. By Lemma 1, this can be fixed by reducing the price in p' of every priceable vertex to the sum of the budgets of incident edges. Note that in p^* every vertex was bought, which implies every price $p^*(v) = \sum_{e:e=(u,v)} \gamma_e^*$ is also the sum of budgets of incident edges. As the total absolute reservation in the end of the algorithm is at most $\delta^{i'}(v')$, the total decrease in revenue that is lost on vertices $v \neq v'$ in this step is at most $\delta^{i'}(v')$. This is exactly the revenue surplus that p' generates over p^* at vertex v'. Thus, pricing p' yields at least as much revenue as p^*. This implies that we can transform any pricing by iteratively adjusting the prices without decreasing revenue, such that all budgets of the edges are equal to those generated by the greedy pricing p_g. In particular, this implies that p_g is an optimal pricing. □

The next proposition, whose proof is left for the full version of this paper, shows that knowing the order of the edges when pricing for the primal-dual algorithm in STACKVC is essential.

Proposition 2. *For every constant $\varepsilon > 0$, there exists an instance of* STACKVC *such that if the order of the edges in the follower's primal-dual 2-approximation algorithm is unknown to the leader, then every pricing p yields an approximation ratio of $\Omega(1/\varepsilon)$.*

Hardness Results. The main argument in the previous section works only for the case of regular vertex cover. Let us turn to the case of set cover with elements contained in at least three sets, i.e. elements with *frequency* at least three. We understand them as hyperedges incident to more than two vertices. In this case it might be profitable to reduce the price for a vertex from the value in the greedy pricing. Indeed, we show that set cover pricing problem is much harder to solve. The proof is omitted due to space limitations.

Theorem 6. STACKSC *is APX-hard even if all elements have maximum frequency 3 and all fixed-cost sets have cost 1.*

Finally, we derive a devastating lower bound on greedy, single-price, and near-uniform price assignments in set cover pricing. Once again, we leave the proof for the full version of this paper.

Theorem 7. *The greedy, single-price, and near-uniform price assignments yield an approximation factor of $2^{\Omega(|I|)}$ for* STACKSC*, where $|I|$ is the size of the representation.*

Acknowledgments. We wish to thank Guido Proietti for the helpful discussions in the starting phase of this study. We are very grateful to Piotr Krysta for several insightful discussions relating to a number of results presented in this paper during a productive week of research in Liverpool.

References

1. G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for Multi-Product Pricing. In *Proc. of 31st ICALP*, 2004.

2. M. Balcan and A. Blum. Approximation Algorithms and Online Mechanisms for Item Pricing. In *Proc. of 7th EC*, 2006.
3. N. Balcan, A. Blum, J. Hartline, and Y. Mansour. Mechanism Design via Machine Learning. In *Proc. of 46th FOCS*, 2005.
4. M. Balcan, A. Blum, and Y. Mansour. Item Pricing for Revenue Maximization. In *Proc. of 9th EC*, 2008.
5. P. Berman and M. Karpinski. Improved Approximation Lower Bounds on Small Occurrence Optimization. ECCC-Report TR03-008, 2003.
6. D. Bilò, L. Gualà, G. Proietti, and P. Widmayer. Computational Aspects of a 2-Player Stackelberg Shortest Paths Tree Game. In *Proc. of 4th WINE*, 2008.
7. P. Briest. Uniform Budgets and the Envy-Free Pricing Problem. In *Proc. of 35th ICALP*, 2008.
8. P. Briest, M. Hoefer, and P. Krysta. Stackelberg Network Pricing Games. In *Proc. of 25th STACS*, 2008.
9. P. Briest and P. Krysta. Single-Minded Unlimited-Supply Pricing on Sparse Instances. In *Proc. of 17th SODA*, 2006.
10. P. Briest and P. Krysta. Buying Cheap is Expensive: Hardness of Non-Parametric Multi-Product Pricing. In *Proc. of 18th SODA*, 2007.
11. J. Cardinal, E. Demaine, S. Fiorini, G. Joret, S. Langerman, I. Newman, and O. Weimann. The Stackelberg Minimum Spanning Tree Game. In *Proc. of 10th WADS*, 2007.
12. S. Chawla, J. Hartline, and R. Kleinberg. Algorithmic Pricing via Virtual Valuations. In *Proc. of 8th EC*, 2007.
13. N. Chen, A. Ghosh, and S. Vassilvitskii. Optimal Envy-free Pricing with Metric Substitutability. In *Proc. of 9th EC*, 2008.
14. E.D. Demaine, U. Feige, M.T. Hajiaghayi, and M.R. Salavatipour. Combination Can Be Hard: Approximability of the Unique Coverage Problem. In *Proc. of 17th SODA*, 2006.
15. K. Elbassioni, R. Raman, and S. Ray. On Profit-Maximizing Pricing for the Highway and Tollbooth Problems. Technical report *arXiv:0901.1140v1*, 2009.
16. K. Elbassioni, R. Sitters, and Y. Zhang. A Quasi-PTAS for Envy-Free Pricing on Line Graphs. In *Proc. of 15th ESA*, 2007.
17. L. Fleischer, K. Jain, M. Mahdian. Tolls for Heterogeneous Selfish Users in Multicommodity Networks and Generalized Congestion Games. In *Proc. of 45th FOCS*, 2004.
18. L. Fleischer. Linear tolls suffice: New bounds and algorithms for tolls in single source networks. *Theoretical Computer Science*, 348(2-3): 217-225, 2005.
19. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, 1979.
20. V. Guruswami, J.D. Hartline, A.R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On Profit-Maximizing Envy-Free Pricing. In *Proc. of 16th SODA*, 2005.
21. J. Hartline and V. Koltun. Near-Optimal Pricing in Near-Linear Time. In *Proc. of WADS*, 2005.
22. G. Karakostas and S. Kolliopoulos. Edge Pricing of Multicommodity Networks for Heterogeneous Users. In *Proc. of 45th FOCS*, 2004.
23. M. Labbé, P. Marcotte, and G. Savard. A Bilevel Model of Taxation and its Application to Optimal Highway Pricing. *Management Science*, 44(12): 1608–1622, 1998.
24. S. Roch, G. Savard, and P. Marcotte. An Approximation Algorithm for Stackelberg Network Pricing. *Networks*, 46(1): 57–67, 2005.
25. C. Swamy. The Effectiveness of Stackelberg Strategies and Tolls for Network Congestion Games. In *Proc. of 18th SODA*, 2007.
26. V. Vazirani. Approximation Algorithms. Springer Verlag, 2001.
27. H. von Stackelberg. *Marktform und Gleichgewicht (Market and Equilibrium).* Verlag von Julius Springer, Vienna, 1934.