How to Sell a Graph: Guidelines for Graph Retailers

Alexander Grigoriev[1], Joyce van Loon[1,⋆], René Sitters[2], and Marc Uetz[1]

[1] Maastricht University, Quantitative Economics,
P.O. Box 616, NL–6200 MD Maastricht, The Netherlands
{a.grigoriev, j.vanloon, m.uetz}@ke.unimaas.nl
[2] Max-Planck Institute für Informatik, Algorithms and Complexity,
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
sitters@mpi-inf.mpg.de

Abstract. We consider a profit maximization problem where we are asked to price a set of m items that are to be assigned to a set of n customers. The items can be represented as the edges of an undirected (multi)graph G, where an edge multiplicity larger than one corresponds to multiple copies of the same item. Each customer is interested in purchasing a bundle of edges of G, and we assume that each bundle forms a simple path in G. Each customer has a known budget for her respective bundle, and is interested only in that particular bundle. The goal is to determine item prices and a feasible assignment of items to customers in order to maximize the total profit. When the underlying graph G is a path, we derive a fully polynomial time approximation scheme, complementing a recent NP-hardness result. If the underlying graph is a tree, and edge multiplicities are one, we show that the problem is polynomially solvable, contrasting its APX-hardness for the case of unlimited availability of items. However, if the underlying graph is a grid, and edge multiplicities are one, we show that it is even NP-complete to approximate the maximum profit to within a factor $n^{1-\varepsilon}$.

Keywords: Pricing problems, tollbooth problem, highway problem, computational complexity, dynamic programming, fully polynomial time approximation scheme.

1 Introduction

We consider a profit maximization problem that is defined on a multi graph. Given is a simple undirected graph $G = (V, I)$ with $|I| = m$ edges, and given are (integral) edge multiplicities c_i, $i \in I$. Each edge can be thought of as an *item* for sale, and the edge multiplicity determines how many copies of the corresponding item are available. We will mainly discuss problems where all c_i's are finite, but notice that most of the related work addresses problems with unlimited availability of items. There is a set of n *customers* $J = \{1, \ldots, n\}$ each of which

[⋆] Supported by METEOR, the Maastricht Research School of Economics of Technology and Organizations.

F.V. Fomin (Ed.): WG 2006, LNCS 4271, pp. 125–136, 2006.

is interested in purchasing exactly one *bundle* $I_j \subseteq I$ of items. This is also known as *single-minded* customers [13]. We assume that each bundle I_j forms a simple *path* in the graph G. Each customer $j \in J$ has an integral *budget* (or valuation) b_j, which is the largest amount that a customer is willing to pay for her bundle I_j.

A feasible solution to the problem consists of an allocation of items to customers and a vector of prices $p = (p_1, \ldots, p_m)$, one for each item, such that

1. each item $i \in I$ is sold to no more than c_i customers,
2. each customer $j \in J$ is either assigned the entire bundle I_j, or nothing,
3. a bundle I_j assigned to a customer j must be affordable, $\sum_{i \in I_j} p_i \leq b_j$.

Within a given solution, let us call a customer a *winner* if she gets assigned her bundle I_j. Let us denote by W the set of winners. We call a set of winners W *feasible* with respect to the availability of items whenever $\sum_{j \in W} |I_j \cap \{i\}| \leq c_i$, for all $i \in I$. Note that for any feasible set of winners W, there exists a price vector p such that all customers in W can indeed afford their respective bundles. The optimization problem consists of finding a feasible set of winners and a vector of prices such that none of the constraints $(1) - (3)$ is violated, and such that the total profit $\Pi(W, p) = \sum_{j \in W} \sum_{i \in I_j} p_i$ is maximized.

One usually distinguishes between solutions that are *envy-free* and those that are not. In the setting with single minded customers considered here, envy-freeness requires that if a customer is not a winner, then the total price of that bundle must exceed her budget. However, we mostly address problems without requiring envy-freeness.

Notice that, in contrast to the classical auction literature, we assume that we *know* the budget of every customer. At first sight this assumption may seem infeasible, yet there are good reasons for studying this type of problems. An understanding of how items should be priced under known budgets, for example, may be useful also for the more difficult problem with unknown budgets. This connection was recently made concrete by Goldberg and Hartline [9]; they reduced a mechanism design problem with unknown budgets to the underlying profit maximization problem with known budgets. Furthermore, data on customer valuations is nowadays collected at large scale, for example via specifically designed web sites; see e.g. [8]. Hence, the assumption of known budgets is reasonable in many settings. Finally, the underlying combinatorial pricing problems have their own appeal, and results with respect to their computational tractability have been obtained only recently [1,3,5,6,8,11,12].

Recall that we address problems where the bundle I_j of any customer$j \in J$ corresponds to a path in an underlying simple graph G. We consider three different types of underlying graphs, namely paths, trees, and grids. The problem on a path (with unlimited availability of items) was recently discussed by Guruswami et al. [11]. They call it the 'highway problem', motivated by the question to find optimal tolls for the usage of a single highway. In their setting, the availability of items is unlimited. To motivate the problem on a path with *limited* availability of items, consider the rental of a set of identical objects over discrete time periods; e.g. houses in holiday parks on the basis of weeks. Customers are interested in renting an object in consecutive periods. Since all objects are identical, in any time period they need to have the same price. But prices may vary from period to period. We can thus interpret a time period as an edge in a path, with edge multiplicity equal to the number of objects available in that period. Notice that envy-freeness is not necessarily an issue, since the manager can freely decide to whom to rent the objects.

1.1 Related Work

The profit maximization problem in which the customer's bundles are paths of an arbitrary graph G, and the availability of items is unlimited, is the 'tollbooth problem' addressed by Guruswami et al. [11]. They show that the problem is APX-hard even if the graph is a star, all budgets are equal to one, and the bundles contain at most 2 items. Another APX-hardness proof for the same problem was given by Briest and Krysta [5]; in contrast to the result of [11] it is also valid if several problem parameters are constantly bounded.

The tollbooth problem with the restriction that the underlying graph is a path is the 'highway problem' introduced by Guruswami et al. [11]. NP-hardness of this problem was recently shown by Bodlaender and Penninkx [4] and Briest and Krysta [5]. Guruswami et al. [11] furthermore propose a polynomial time dynamic programming algorithm when the budgets are bounded by a constant, and a pseudo-polynomial time dynamic programming algorithm when the bundle sizes are bounded by a constant.

When bundles are not paths in a graph, but arbitrary subsets of the given set of items, Demaine et al. [6] show that the problem (again with unlimited availability of items) is hard to approximate to within a (semi-)logarithmic factor. For the same problem, there exists an approximation scheme with almost linear running time, given that the number of distinct items is constant, by Hartline and Koltun [12]. Moreover, Balcan and Blum [3] derive a $\mathcal{O}(k)$-approximation algorithm, given that each customer is interested in bundles of at most k items.

Finally, independently in [3] and [5], two FPTAS's are presented for the problem where the customers' bundles are *nested*. That is, for any two bundles I_a and I_b it holds that $I_a \subseteq I_b$, $I_b \subseteq I_a$ or $I_a \cap I_b = \emptyset$. Notice that such an instance can be interpreted as a problem where the bundles are subpaths of a single path, by ordering the items appropriately. (The converse is not true, however.)

1.2 Our Results

In Section 2, we address the problem where the underlying graph G is a path, and there is an upper bound C on the availability of any item. We propose a dynamic programming algorithm that computes an optimal solution in time $\mathcal{O}(n^{2C}B^{2C}m)$. Here, $B = \max_j b_j$ is an upper bound on the budgets. The same problem with unlimited supply of items allows a dynamic programming algorithm with a computation time of $\mathcal{O}(B^{B+2}n^{B+3})$, see [11]. Based on our dynamic programming algorithm, we moreover derive an FPTAS for that problem, given that the maximum availability of any item C is constant. In contrast to previous results in that direction [3,5,12], this FPTAS does neither require a constant number of items, nor nested bundles or bounded budgets.

In Section 3 we address the problem where the availability of any item is exactly one. For the case that graph G is a path, the problem reduces to finding a maximum weight independent set in an interval graph; thus it is polynomially solvable [14]. When we generalize from a path to a tree, we can show that the problem remains polynomially solvable. When the underlying graph G is a grid, however, we show that it is NP-complete to approximate the maximum profit to within $n^{1-\varepsilon}$, for any $\varepsilon > 0$. (Recall that n is the number of customers.)

2 Selling a Multi Path

In this section, we restrict the underlying graph $G = (V, I)$ to be a path. We first discuss some preliminaries. Thereafter, we present a dynamic programming approach and a fully polynomial time approximation scheme for the case where the edge multiplicities c_i are bounded by some constant C.

2.1 Preliminaries

It is not hard to see that the profit maximization problem on a multi path is polynomially solvable if either the set of winners W is given, or if the vector of prices p is given.

Lemma 1. *The profit maximization problem on a multi path is polynomially solvable if the vector of prices $p = (p_1, \ldots, p_m)$ is given.*

Proof. If the vector of prices $p = (p_1, \ldots, p_m)$ is given, we only need to find a feasible set of winners that maximizes the total revenue. Whenever the items are available in unlimited supply, this is trivial and the set of winners is just $W := \{j \mid \sum_{i \in I_j} p_i \le b_j\}$. For the case of limited supply, let W' be the set of customers for which the bundle is affordable, given the price vector p. For any item i, we can not sell more than c_i copies. Let a_{ij} be equal to 1 if item $i \in I_j$ for customer j, and 0 otherwise. We find a profit-maximizing feasible subset of winners by solving the following linear program, where $x_j = 1$ iff customer j is a winner.

$$\begin{array}{lll} \max & \sum\limits_{j \in W'} \left(\sum\limits_{i \in I_j} p_i \right) x_j & \\ \text{s.t.} & \sum\limits_{j \in W'} a_{ij} x_j \le c_i & \forall i \in I \\ & 0 \le x_j \le 1 & \forall j \in W' \end{array}$$

The constraint matrix of this linear program has the consecutive ones property, that is, all entries that are 1 appear consecutively in any column. This because the bundles I_j of any customer j consist only of consecutive items. A consecutive ones matrix is totally unimodular [15]. Hence, the corresponding polyhedron only has integral vertices, and the LP yields an integral optimal solution. □

If on the other hand a feasible set of winners $W \subseteq J$ is given, we find an optimal price vector $p = (p_1, \ldots, p_m)$ by solving the following linear program.

$$\begin{aligned} \max \ & \sum_{j\in W}\sum_{i\in I_j} p_i \\ \text{s.t.} \ & \sum_{i\in I_j} p_i \le b_j \ \forall j \in W \\ & p_i \ge 0 \qquad \forall i \in I \end{aligned}$$

Since this constraint matrix has the consecutive ones property, too, we obtain the following.

Lemma 2 (See also [11, Lemma 5.1]). *The profit maximization problem on a multi path is polynomially solvable if a feasible set of winners $W \subseteq J$ is given. Moreover, since the budgets b_j are integral, there exists an optimal, integral price vector.*

2.2 Complexity

Bodlaender and Penninkx [4] as well as Briest and Krysta [5] recently showed the following.

Theorem 1 ([4,5]). *The profit maximization problem on a multi path is (weakly) NP-hard, even if the edge multiplicity is at most three.*

For their reduction it suffices, but it is also necessary, that the availability of any item is 3. If we restrict the availability of any item to be at most 2, the complexity remains open.

2.3 Dynamic Programming Algorithm

Recall that for each item $i \in I$ there are c_i copies available, and for convenience of notation let

$$C \ge \max_{i\in I} c_i$$

be an upper bound on the availability of any item. We show that we can solve this problem in time $\mathcal{O}(n^{2C}B^{2C}m)$ by finding a longest path in an acyclic digraph.

We create an m-layered digraph with an additional source s and sink t (layers 0 and $m+1$, respectively). There are arcs only from layer i to $i+1$, for $i = 0, \ldots, m$. Hence, in any $s-t$ path, there are exactly $m+2$ nodes. In every node in layer i (corresponding to item i), we store all winners j that purchase item i. Moreover, we store the respective total amounts all these winners spend on all items in their respective bundles up to and including item i. Any node x (more precisely, the path $s-x$) represents a feasible partial solution. Arcs from node x of layer i to node y of layer $i+1$ are only introduced if the path $s-y$ represents a feasible extension of the partial solution represented by the path $s-x$. The weight on an arc that connects a node of layer i to a node of layer $i+1$ is equal to the profit earned on item $i+1$, that is, the total amount that the corresponding winners pay for item $i+1$. Therefore, the weight of the longest $s-t$ path in the digraph will equal the maximum total profit, and the set of winners can be reconstructed from the longest $s-t$ path, too. The algorithm below shows a more formal description.

Algorithm 1. Dynamic Programming Algorithm

```
Input: Profit maximization problem on a multi path with maximal availability
        of any item $\leq C$
Output: Assignment of items to customers and item prices $p_i$
begin (construction of digraph $D$)
    nodes: For each item $i \in I$, we introduce a layer of nodes: Denote by $J^i$ the
    set of customers with $i \in I_j$. By $K^i = (j_1, j_2, \ldots, j_k)$ we denote any (sorted)
    subset of $J^i$ of cardinality $k$, where $k \leq \min\{c_i, |J^i|\} \leq C$. Define
    $H_j := \{0, 1, \ldots, b_j\}$ as the possible total amount customer $j \in K^i$ can spend
    for items $\{1, \ldots, i\} \cap I_j$. Let $h^i \in H_{j_1} \times H_{j_2} \times \cdots \times H_{j_k}$ be a vector denoting
    how much each customer $j$ spends for items $\{1, \ldots, i\} \cap I_j$, for each $j \in K^i$. If
    $K^i = \emptyset$, we let $h^i = 0$. Let all such pairs $(K^i, h^i)$ be the nodes in layer $i$ of $D$,
    for $i = 1, \ldots, m$. Let $s$ and $t$ denote source and sink. To unify notation,
    assume $s = (\emptyset, 0)$ and $t = (\emptyset, 0)$;
    arcs: Insert an arc from node $(K^i, h^i)$ to node $(K^{i+1}, h^{i+1})$ if:
    (1) For all $j \in K^i$ with $i + 1 \in I_j$, $j \in K^{i+1}$, and for all $j \in J^i \setminus K^i$ with
    $i + 1 \in I_j$, $j \notin K^{i+1}$.
    (2) There exists a unique integral value $d \geq 0$ such that $d = h_j^{i+1} - h_j^i$ for all
    $j \in K^i \cap K^{i+1}$, and $d = h_j^{i+1}$ for all $j \in K^{i+1} \setminus K^i$.
    We furthermore connect source node $s$ to all nodes $(K^1, h^1)$, and we connect
    all nodes $(K^m, h^m)$ to sink node $t$.
    arc lengths: For an arc $a^i$ that connects $(K^{i-1}, h^{i-1})$ and $(K^i, h^i)$: If
    $K^i = \emptyset$, we let the length of arc $a^i$ be $\ell(a^i) = 0$, and if $K^i \neq \emptyset$, we let the
    length of arc $a^i$ be $\ell(a^i) = d|K^i|$, where $d$ is the (unique) value from
    condition (2) above.
end
solution: Compute the longest $s - t$ path $\mathcal{P}$ in digraph $D$. Whenever for
customer $j$ we have that $j \in K^i$ with $(K^i, h^i) \in \mathcal{P}$, customer $j$ gets assigned
item $i$. The price $p_i$ for item $i$ equals $\ell(a^i)/|K^i|$, where $a^i$ is the arc from path $\mathcal{P}$
that connects nodes $(K^{i-1}, h^{i-1})$ and $(K^i, h^i)$;
```

Theorem 2. *The dynamic programming algorithm outputs an optimal solution for an instance of the profit maximization problem on a multi path in* $\mathcal{O}(n^{2C} B^{2C} m)$ *time.*

Proof. The algorithm assigns items to customers and computes prices for all items according to an $s - t$ path $\mathcal{P}$. This solution is feasible if each customer j either gets the complete bundle I_j or nothing, no item $i \in I$ is assigned to more than c_i customers, and if for every customer j who gets her bundle I_j, we have that $\sum_{i \in I_j} p_i \leq b_j$.

Let an $s - t$ path $\mathcal{P}$ be fixed, let a^i be the arcs on $\mathcal{P}$, $\ell(a^i)$ the length of these arcs, and abusing notation let (W^i, h^i) be the nodes on $\mathcal{P}$. Set W^i is thus the set of customers that get assigned item i.

By definition of the nodes of the digraph, no customer j will be assigned an item not from her bundle I_j. For any customer j, consider an item $i \in I_j$ such that $j \in W^i$. That is, customer j gets assigned item i by the algorithm. By condition (1) of the digraph construction, all other items of bundle I_j must be assigned to customer j as well. Next, for any node (W^i, h^i), we have by definition that $|W^i| \leq c_i$, hence no item is oversold. Finally, let us consider the budget constraint of customer j. We know that $I_j = \{k, \ldots, k'\}$ for some $k \leq k'$. We have that

$$\sum_{i \in I_j} p_i = \sum_{i=k}^{k'} p_i = \sum_{i=k}^{k'} \frac{\ell(a^i)}{|W^i|} = h_j^k + \sum_{i=k+1}^{k'} (h_j^i - h_j^{i-1}) = h_j^{k'} \leq b_j \,.$$

The third equality holds due to condition (2) of the digraph construction, and the last inequality holds because $h_j^{k'} \in H_j = \{0, 1, \ldots, b_j\}$.

Now we know that any $s-t$ path $\mathcal{P}$ in D defines a feasible solution, and $W := \bigcup_{i \in I} W^i$ denotes the set of winners. The length of a path is

$$\sum_{a^i \in \mathcal{P}} \ell(a^i) = \sum_{i \in I} p_i |W^i| = \sum_{j \in W} \sum_{i \in I_j} p_i \,.$$

In other words, the path length defines the profit of the corresponding solution, thus the longest path yields an optimal solution.

To arrive at the computation time of $\mathcal{O}(n^{2C} B^{2C} m)$, we only need to estimate the size of the digraph D. For every item $i \in I$, there are at most $\mathcal{O}(n^C)$ different sets K^i and at most $\mathcal{O}(B^C)$ different vectors h^i. Thus, per item $i \in I$, we have at most $\mathcal{O}(n^C B^C)$ nodes (K^i, h^i). For any $i \in I$, every node (K^i, h^i) is connected to at most $\mathcal{O}(n^C B^C)$ nodes (K^{i+1}, h^{i+1}). So, per item, there are at most $\mathcal{O}(n^{2C} B^{2C})$ arcs, which means that there are at most $\mathcal{O}(n^{2C} B^{2C} m)$ arcs in D. The computation time to find the longest path in D is linear in the number of arcs, since D is acyclic [2]. □

Notice that the solution constructed by the dynamic programming algorithm need not be envy-free.

2.4 FPTAS

We next show how to turn the dynamic programming algorithm into a fully polynomial time approximation scheme (FPTAS); that is, an algorithm that computes a solution with profit at least $(1 - \varepsilon)$ times the optimum profit, in time polynomial in the input and $1/\varepsilon$. To that end, we just apply the dynamic programming algorithm on a rounded instance in which $K := \frac{\varepsilon B}{2n^2}$ for any $\varepsilon > 0$ and the customers' budgets are $b'_j := \lfloor b_j / K \rfloor$.

Lemma 3. *For every solution (W, p) of the original instance, there exists a solution (W, p'') of the rounded instance with $\Pi(W, p) > \frac{1}{K} \Pi(W, p'') - mn$.*

Proof. Let (W, p) be a feasible solution of the original instance with profit $\Pi(W, p)$. Let $p''_i = \lfloor p_i / K \rfloor$, $i = 1, \ldots, m$. Note that $p_i/K - 1 < p''_i \leq p_i/K$. For the original instance we have for every winner $j \in W$, $\sum_{i \in I_j} p_i \leq b_j$, and it follows that

$$\sum_{i \in I_j} p''_i = \sum_{i \in I_j} \left\lfloor \frac{p_i}{K} \right\rfloor \leq \left\lfloor \frac{\sum_{i \in I_j} p_i}{K} \right\rfloor \leq \left\lfloor \frac{b_j}{K} \right\rfloor = b'_j \,.$$

Hence, the same set of customers W can be made winners in the rounded instance. Then the capacity constraint is satisfied as well, and the solution (W, p'') is feasible. Finally, we have

$$\Pi(W, p'') = \sum_{j \in W} \sum_{i \in I_j} p''_i > \sum_{j \in W} \sum_{i \in I_j} \left(\frac{p_i}{K} - 1 \right) \geq \frac{1}{K} \Pi(W, p) - mn. \qquad \square$$

Lemma 4. *For every solution (W', p') of the rounded instance, there exists a solution $(W', \tilde{p})$ of the original instance with $\Pi(W', \tilde{p}) = K\Pi(W', p')$.*

Proof. Let (W', p') be a solution in the rounded instance with revenue $\Pi(W', p')$. Let $\tilde{p}_i = p'_i K$ be prices in the original instance, $i = 1, \ldots, m$. (This is integer because p'_i and K are integer.) Then the budget constraint for every customer $j \in W'$ is satisfied, because

$$\sum_{i \in I_j} \tilde{p}_i = K \sum_{i \in I_j} p'_i \leq K b'_j = K \left\lfloor \frac{b_j}{K} \right\rfloor \leq b_j \, .$$

Hence, we can make the same set W' of customers winners, and solution $(W', \tilde{p})$ is feasible for the original instance. The revenue can be written as

$$\Pi(W', \tilde{p}) = \sum_{j \in W'} \sum_{i \in I_j} \tilde{p}_i = \sum_{j \in W'} \sum_{i \in I_j} (p'_i K) = K \sum_{j \in W'} \sum_{i \in I_j} p'_i = K\Pi(W', p'). \qquad \square$$

We can now combine Lemmas 3 and 4 to obtain an FPTAS.

Theorem 3. *There exists an FPTAS for the profit maximization problem on a multi path.*

Proof. Let (W, p) and (W', p') be the optimal solutions in the original and rounded instances, respectively. Consider solution $(W', \tilde{p})$ for the original instance, where $\tilde{p}_i = Kp'_i$, $i = 1, \ldots, m$, and solution (W, p'') for the rounded instance, where $p''_i = \lfloor p_i/K \rfloor$. An application of the previous two lemmas now yields

$$\Pi(W', \tilde{p}) = K\Pi(W', p') \geq K\Pi(W, p'') > K \left(\frac{1}{K} \Pi(W, p) - mn \right) = \Pi(W, p) - \varepsilon B \frac{mn}{2n^2} \, ,$$

where the first inequality holds due to optimality of (W', p') for the rounded instance. Note that $m \leq 2n - 1 < 2n$, and the optimal profit is at least equal to the maximum budget B, so $B \leq \Pi(W, p)$. Thus, $\Pi(W', \tilde{p}) > (1 - \varepsilon)\Pi(W, p)$.

Concerning the computation time to compute the optimal solution (W', p'), observe that the size of the digraph is $\mathcal{O}(n^{6C+1}/\varepsilon^{2C})$. Hence, the computation time to find the longest path is polynomial in terms of n and $1/\varepsilon$. $\square$

Again, notice that the solution constructed by the FPTAS need not be envy-free.

3 Selling Simple Graphs

In this section we assume that the availability of any item i is one, or in other words, the edge multiplicities c_i are one. We consider three types of graphs, namely paths, trees and grids.

Theorem 4 ([16]). *The profit maximization problem on a simple path can be solved in $\mathcal{O}(n^2)$ time.*

In fact, the result of Theorem 4 is not surprising, since the problem reduces to finding a maximum weight independent set in an interval graph, a problem known to be solvable in polynomial time [14].

3.1 Trees

Guruswami et al. [11] show that the problem with unlimited availability of items is APX-hard even on star graphs. Contrasting this complexity result, we prove that if the availability of each item is exactly one, the profit maximization problem on a tree can be solved in polynomial time. (Again, recall that we do not require the solution to be envy-free.)

Theorem 5. *The profit maximization problem on a simple tree can be solved in $\mathcal{O}(n^5)$ time.*

Proof. Consider the graph $H = (J, E)$ where $(j,k) \in E$ if and only if $I_j \cap I_k \neq \emptyset$, for two customers $j, k \in J$. Since $\{I_j | j \in J\}$ is a collection of simple paths in a tree, graph H is called an *EPT graph* [10]. Since G is a tree and availability of each item is exactly one, the maximum weight independent set in H with vertex weights b_j, $j \in J$, is the optimal set of winners W, and the weight of this independent set is equal to the maximum profit. The vector of optimal prices can be straightforwardly obtained by setting the price of one arbitrary edge from I_j, $j \in W$, to b_j, and setting the prices of all other edges in I_j to 0. The remaining edges in the tree can be priced arbitrarily.

A polynomial time algorithm to compute a maximum weight independent set in an EPT graph was described by Tarjan [17]. The algorithm is a recursive procedure that decomposes the problem on the basis of clique separators. The polynomial running time is a consequence of the fact that the atoms, i.e., the non-decomposable subgraphs of EPT graphs are line graphs. For line graphs, the maximum weight independent set problem is just the maximum weight matching problem, which can be solved in $\mathcal{O}(n^3)$ time by Edmonds algorithm [7]. The total time complexity is bounded by $\mathcal{O}(n^5)$. □

3.2 Grids

Demaine et al. [6] show that the profit maximization problem where the bundles are arbitrary subsets of items (and with unlimited availability of items) is hard to approximate to within a (semi-)logarithmic factor. If we restrict bundles to be paths in a general graph, Briest and Krysta [5] show that the problem is APX-hard even if several parameters of the problem are constantly bounded. For the even more restricted problem where the bundles are paths in a star, Guruswami et al. [11] also show APX-hardness.

Here we show that if the availability of items is bounded, we can derive an even stronger inapproximability result, even for a very restricted class of graphs and customers' bundles.

Theorem 6. *For all $\varepsilon > 0$, approximating the profit maximization problem on a simple grid to within $n^{1-\varepsilon}$ is NP-hard, even with unit budgets, and when each item is an element of at most two bundles. The same result holds if the solution is required to be envy-free.*

Proof. For the proof we construct an approximation preserving reduction from Independent Set. In the latter problem, given a graph $G = (V, E)$, the problem is to find a maximum cardinality subset $S \subseteq V$ such that no two vertices from S are adjacent. It is NP-hard to approximate Independent Set within a factor $|V|^{1-\varepsilon}$; see [18].

Let $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$. We construct the instance of the profit maximization problem as follows. We create a grid graph with $(n + 1)$ horizontal layers and $(2m + 2)$ vertical layers. We index the vertices of the grid graph by pairs (i, j) where i is the index of the vertical layer and j is the index of the horizontal layer. Let horizontal layer $j \in \{1, \dots, n\}$ correspond to vertex $v_j \in V$, and let the edge $((2i, n + 1), (2i + 1, n + 1))$ in the grid correspond to edge $e_i \in E$. Next, for each vertex $v_j \in V$, we introduce a customer in the profit maximization problem with a bundle defined by the following simple path in the grid graph. The path starts at point $(1, j)$ and ends at point $(2m + 2, j)$ following the layer j everywhere except for the edges $((2i, j), (2i + 1, j))$ such

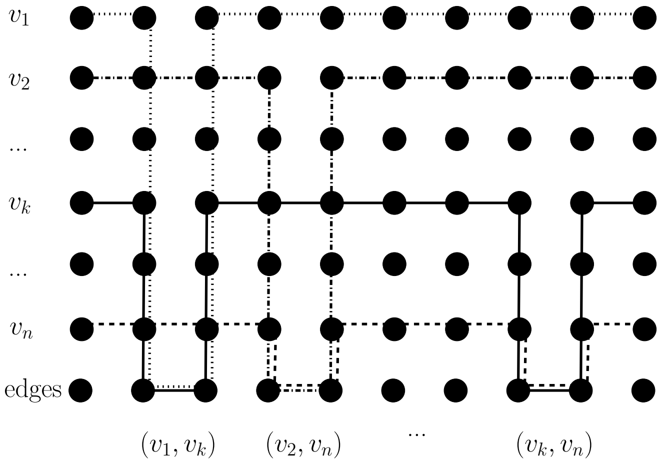

Fig. 1. Grid graph from the reduction

that $v_j \in e_i$. These edges are substituted by vertical detours, passing through edges $((2i, n+1), (2i+1, n+1))$; see Figure 1 for an example. We complete the construction setting the budget of each customer to 1.

We claim that in G there exists an independent set of cardinality K if and only if there exists a solution of the corresponding profit maximization problem with total profit K. By construction, two paths corresponding to adjacent vertices in G must share some edge e in layer $(n+1)$. Since the multiplicity of edge e is 1, only one of these paths can be present in a feasible solution. Hence, the total profit in the profit maximization problem is at most the maximum cardinality independent set in G. Now, consider an independent set S in G and any two vertices in this independent set. By construction, the two corresponding paths in the profit maximization problem are edge disjoint. Therefore, there is a solution of the profit maximization problem where S defines the set of winners and this allocation is feasible with respect to availability of items. For each $v_j \in S$ we set the price of the grid edge $((1, j), (2, j))$ to 1, for each $v_j \notin S$ we set the price of $((1, j), (2, j))$ to 2, and for all other edges of the grid we set the prices to 0. In the constructed solution of the profit maximization problem the total profit equals $|S|$. Thus, the reduction preserves the objective value.

Since the number of customers n in the profit maximization problem exactly equals $|V|$, we derive that the profit maximization problem is hard to approximate to within a factor $n^{1-\varepsilon}$. It remains to notice that the constructed solution is envy-free since the price of the bundle of each non-winning customer equals 2, which is greater than the budget. Therefore, the theorem holds also if we require the solution to be envy-free. □

4 Conclusion

Notice that the currently best known negative result on the tractability of the profit maximization problem on a path is NP-hardness [4,5]. Even though several FPTAS's (including the one of this paper) exist whenever certain parameters are constantly bounded, the best known positive result for the general case is a logarithmic approximation [3,11]. It thus remains an intriguing open problem to obtain a (deterministic) constant approximation algorithm.

Acknowledgements. Thanks to Jason Hartline for several interesting remarks and pointers to several references.

References

1. G. Aggarwal, T. Feder, R. Motwani, and A. Zhu, *Algorithms for multi-product pricing*, Automata, Languages and Programming - ICALP 2004 (J. Díaz, J. Karhumäki, A. Lepistö, and D. Sannella, eds.), Lecture Notes in Computer Science, vol. 3142, Springer, 2004, pp. 72–83.
2. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows*, Prentice Hall, New Jersey, 1993.

3. M.F. Balcan and A. Blum, *Approximation algorithms and online mechanisms for item pricing*, Proc. of the 7th ACM Conference on Electronic Commerce, ACM, 2006, pp. 29–35.
4. H. Bodlaender and E. Penninkx, *An elegant NP-completeness proof for profit maximization on paths*, personal communication, May 2005.
5. P. Briest and P. Krysta, *Single-minded unlimited supply pricing on sparse instances*, Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM-SIAM, 2006, pp. 1093–1102.
6. E. D. Demaine, U. Feige, M.T. Hajiaghayi, and M. R. Salavatipour, *Combination can be hard: Approximability of the unique coverage problem*, Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM-SIAM, 2006, pp. 162–171.
7. J. Edmonds, *Matching and a polyhedron with 0-1 vertices*, Journal of Research of the National Bureau of Standards, B **69** (1965), 125–130.
8. P. W. Glynn, B. Van Roy, and P. Rusmevichientong, *A nonparametric approach to multi-product pricing*, Operations Research **54** (2006), no. 1, 82–98.
9. A. V. Goldberg and J. D. Hartline, *Competitive auctions for multiple digital goods*, Algorithms - ESA 2001 (F. Meyer auf der Heide, ed.), Lecture Notes in Computer Science, vol. 2161, Springer, 2001, pp. 416–427.
10. M. C. Golumbic and R. E. Jamison, *The edge intersection graphs of paths in a tree*, Journal of Combinatorial Theory, Series B **38** (1985), 8–22.
11. V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry, *On profit-maximizing envy-free pricing*, Proc. 16th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM-SIAM, 2005, pp. 1164–1173.
12. J. D. Hartline and V. Koltun, *Near-optimal pricing in near-linear time*, Algorithms and Data Structures - WADS 2005 (F. K. H. A. Dehne, A. López-Ortiz, and J.-R. Sack, eds.), Lecture Notes in Computer Sciences, vol. 3608, Springer, 2005, pp. 422–431.
13. D. Lehman, L. I. O'Callaghan, and Y. Shoham, *Truth revelation in approximately efficient combinatorial auctions*, Journal of the ACM **49** (2002), no. 5, 1–26.
14. R. H. Möhring, *Algorithmic aspects of comparability graphs and interval graphs*, Graphs and Order (I. Rival, ed.), Reidel, Dordrecht, 1985, pp. 41–101.
15. G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, John Wiley & Sons, New York, 1988.
16. M. H. Rothkopf, A. Pekec, and R. M. Harstad, *Computationally manageable combinatorial auctions*, Management Science **44** (1998), no. 8, 1131–1147.
17. R. E. Tarjan, *Decomposition by clique separators*, Discrete Mathematics **55** (1985), 221–232.
18. D. Zuckerman, *Linear degree extractors and the inapproximability of* MAX CLIQUE *and* CHROMATIC NUMBER, Electronic Colloquium on Computational Complexity, Report TR05-100 (2005), `http://www.eccc.uni-trier.de/eccc/`.