Optimal Pricing of Capacitated Networks

Alexander Grigoriev[⋆1], Joyce van Loon[⋆⋆1], René Sitters[2], and Marc Uetz[3]

[1] Maastricht University, Quantitative Economics,
P.O.Box 616, NL–6200 MD Maastricht, The Netherlands
{a.grigoriev,j.vanloon}@ke.unimaas.nl
[2] Eindhoven University of Technology, Mathematics and Computer Science,
P.O. Box 513, NL–5600 MB Eindhoven, The Netherlands
r.sitters@tue.nl
[3] University of Twente, Applied Mathematics,
P.O. Box 217, NL–7500 AE Enschede, The Netherlands
m.uetz@utwente.nl

Abstract. We address the algorithmic complexity of a profit maximization problem in capacitated, undirected networks. We are asked to price a set of m capacitated network links to serve a set of n potential customers. Each customer is interested in purchasing a network connection that is specified by a simple path in the network and has a maximum budget that we assume to be known to the seller. The goal is to decide which customers to serve, and to determine prices for all network links in order to maximize the total profit. We address this pricing problem in different network topologies. More specifically, we derive several results on the algorithmic complexity of this profit maximization problem, given that the network is either a path, a cycle, a tree, or a grid. Our results include approximation algorithms as well as inapproximability results.

Keywords: Network pricing, highway problem, computational complexity, dynamic programming, approximation algorithms.

1 Introduction

We consider a profit maximization problem that is defined on a capacitated, undirected network. Given is a simple undirected graph $G = (V, E)$ with $|E| = m$ edges, and given are integral edge capacities c_e, $e \in E$. Each edge can be thought of as a network link, and the edge capacity determines the maximum number of customers the link can accommodate. We mainly, but not exclusively, discuss problems where edge capacities c_e are finite. Given is a set of n potential *customers* $J = \{1, \ldots, n\}$ each of which is interested in purchasing a network connection between two vertices of the graph. In contrast to many classical network problems, we assume that each customer wants one specific simple path, denoted $E_j \subseteq E$, rather than *any* path that connects the two vertices. In literature on auctions, this is also known as *single-minded* customers [15]. Each customer $j \in J$ has an integral

⋆ Partially supported by the Netherlands Organization for Scientific Research (NWO), within the project Treewidth and Combinatorial Optimization (TACO).
⋆⋆ Supported by METEOR, the Maastricht Research School of Economics of Technology and Organizations.

budget (or valuation) b_j, which is the largest amount that a customer is willing to pay for her path E_j.

A feasible solution to the problem consists of a subset $W \subseteq J$ of customers that do get their requested paths E_j (the *winners*), and a vector of prices $p = (p_1, \ldots, p_m)$, one for each edge $e \in E$, such that

1. each edge $e \in E$ accommodates no more than c_e customers, and
2. $\sum_{e \in E_j} p_e \leq b_j$ for all customers $j \in W$ (customers can afford their paths).

The optimization problem consists of finding a feasible solution such that the total profit $\Pi(W, p) = \sum_{j \in W} \sum_{e \in E_j} p_e$ 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 j is *not* a winner, then the total price of path E_j 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 edges should be priced under known budgets may be useful also for the more difficult problem with unknown budgets. This latter problem is addressed in [4] and [9]. Both papers study the design of incentive-compatible mechanisms to induce customers to truthfully report their budgets. Particularly, Balcan et al. [4] suggest a general approach for reducing incentive-compatible mechanism design problems to the underlying algorithmic pricing problems. Hence, there is interest in purely algorithmic pricing problems also from the perspective of mechanism design. Furthermore, data on customer valuations is nowadays collected at large scale, for example via specifically designed web sites; see e.g. [19]. 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, 13, 14, 19].

Recall that we address problems where customers request a specific simple path in an underlying graph G. We consider four different graph topologies, namely paths, cycles, trees, and grids. Compared to previous work in which the edge capacities were unlimited, we consider the setting with *limited* capacities in this paper.

1.1 Related work

The profit maximization problem with unlimited edge capacities in which the customers purchase a specific path in a graph is the 'tollbooth problem' addressed by Guruswami et al. [13]. This problem is motivated by the question to find optimal tolls for the usage of highway segments. They show that the problem is APX-hard even if the underlying graph is a star, all budgets are equal to one, and the customers' paths contain at most two edges. Briest and Krysta [5] prove that this problem remains APX-hard under other various strong restrictions.

The tollbooth problem with the restriction that the underlying graph is a path is the 'highway problem' introduced by Guruswami et al. [13]. NP-hardness of this problem was recently shown by Briest and Krysta [5]. Guruswami et al. [13] 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 number of edges in the customers' paths is bounded by a constant. For the highway problem with additional constraint that a longer path should be more expensive than a shorter path, there exists a $\log B$-approximation algorithm [12], where B is the largest budget. For the highway problem in which customers have a non-unit demand and edge capacities are limited, Elbassioni et al. [8] present a quasi-polynomial time approximation scheme.

When the potential customers do not purchase a path in a network but arbitrary subsets of the given set of edges, the whole network connotation is meaningless, and we arrive at the problem to price a set of items, and customers request arbitrary bundles of those items. In this setting, Demaine et al. [6] show that the problem with unlimited availability of items is hard to approximate within a (semi-)logarithmic factor. For the same problem, there exists an approximation scheme with almost linear computation time by Hartline and Koltun [14], given that the number of distinct items is constant. Moreover, Balcan and Blum [3] derive an $O(k)$-approximation algorithm, given that each customer is interested in a subset of at most k items. Under the additional restriction that a larger bundle is more expensive than a smaller one, Grigoriev et al. [12] present a PTAS.

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 subsets of items E_a and E_b it holds that $E_a \subseteq E_b$, $E_b \subseteq E_a$ or $E_a \cap E_b = \emptyset$.

1.2 Our results

In Section 2, we address the problem where the underlying graph G is a path. If the edge capacities are unlimited, we derive an exact dynamic programming algorithm for the case of uniform budgets, leading to a logarithmic approximation algorithm for the problem with non-uniform budgets. Then, we regard the problem on a path with an upper bound C on the capacity of any edge, that is, $c_e \leq C$ for all $e \in E$. We propose a dynamic programming algorithm that computes an optimal solution in time $\mathrm{O}(n^{2C}B^{2C}m)$. Here, $B = \max_j b_j$ is an upper bound on the budgets. Based on our dynamic programming algorithm, we moreover derive an FPTAS for that problem, given that the maximum capacity of any edge, C, is a constant. In contrast to previous results in this direction [3, 5, 14], our FPTAS does neither require a constant number of edges in the subpaths, nor nested paths or bounded budgets. The NP-hardness proof for the profit maximization problem on a path can easily be adapted to prove NP-hardness for the same problem on a cycle, as shown in Section 3. Moreover, in this section, we not only regard the problem in which each customer $j \in J$ requests a path $E_j \subseteq E$, but also the problem where customers only specify two vertices to be connected. For both problems, we show how to adapt the dynamic programming approach of Section 2 in order to solve them to optimality in time $\mathrm{O}(n^{3C}B^{3C}m)$ and $\mathrm{O}(n^{3C}B^{3C}4^{C}m)$, respectively.

In Section 4 we address the problem where the capacity of any edge 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 [16]. When we generalize from a path to a cycle, the problem reduces to finding a maximum weight independent set in a circular arc graph, which is also known to be solvable in polynomial time [10]. Furthermore, if the underlying graph G is 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 within a factor $n^{1-\varepsilon}$, for any $\varepsilon > 0$. Recall that n is the number of customers.

2 Selling a Path

In this section, we restrict the underlying graph $G = (V, E)$ to be a path. We first discuss some preliminaries. Thereafter, we present a dynamic programming algorithm for the case with unlimited edge capacities and uniform budgets, which leads to a logarithmic

approximation algorithm for the case with non-uniform budgets. Finally, we present a dynamic programming algorithm and a fully polynomial time approximation scheme for the case where edge capacities c_e are bounded by some constant C.

2.1 Preliminaries

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

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

Proof. Given the vector of prices $p = (p_1, \ldots, p_m)$, we only need to find a feasible set of winners that maximizes the total revenue. Whenever the edge capacities are unlimited, this is trivial and the set of winners is just $W := \{j \in J : \sum_{e \in E_j} p_e \leq b_j\}$. For the case of limited capacity, let W' be the set of customers for which the requested path is affordable, given the price vector p. For any edge e, we can not accommodate more than c_e customers. Let a_{ej} be equal to 1 if edge $e \in E_j$ for customer j, and 0 otherwise. We find a profit-maximizing feasible subset of winners by solving linear program (1), where $x_j = 1$ if and only if customer j is a winner. Note that $x_j = 0$ for all customers $j \in J \setminus W'$.

$$\begin{aligned} \max \sum_{j \in W'} \left(\sum_{e \in E_j} p_e \right) x_j & \\ \text{s.t.} \quad \sum_{j \in W'} a_{ej} x_j \leq c_e & \quad \forall e \in E \\ 0 \leq x_j \leq 1 & \quad \forall j \in W' \end{aligned} \tag{1}$$

The constraint matrix $\{a_{ej}\}_{e \in E, j \in J}$ of this linear program has the consecutive ones property, that is, all entries that are 1 appear consecutively in any column. This is because the requested path E_j of any customer j consists only of consecutive edges. A consecutive ones matrix is totally unimodular [17]. Hence, the corresponding polyhedron only has integral vertices, and linear program (1) yields an integral optimal solution. □

If a feasible set of winners $W \subseteq J$ is given, we find an optimal price vector $p = (p_1, \ldots, p_m)$ by solving linear program (2). In the case with limited edge capacities, the maximum

capacity of the edges is taken into account.

$$\begin{aligned} \max \sum_{j \in W} \sum_{e \in E_j} p_e & \\ \text{s.t.} \sum_{e \in E_j} p_e \leq b_j \; \forall j \in W & \\ p_e \geq 0 \quad \forall e \in E & \end{aligned} \tag{2}$$

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

Lemma 2 ([13, Lemma 5.1]). *The profit maximization problem on a 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

Theorem 1 ([5, Theorem 2.1]). *The profit maximization problem on a path is NP-hard.*

For the reduction by Briest and Krysta [5] from PARTITION it suffices, but it is also necessary, that the capacity of any edge is 3. If the capacity of any edge is at most 2, the complexity status of the problem remains open.

2.3 Unlimited Edge Capacities

Consider the problem with unlimited edge capacities. First let us assume that every customer has the same positive budget b, that is, all customers have *uniform* budgets.

Theorem 2. *The profit maximization problem on a path can be solved in $O(m^3 n)$ time if it has unlimited edge capacities and uniform positive budgets, i.e., $b_j = b$ for all $j \in J$.*

Proof. Consider the problem with $b = 1$. In Lemma 2 we already argued that, given integral budgets and a feasible set of winners $W \subseteq J$, there exists an optimal integral price vector. Since $b_j = 1$ for all $j \in J$, all RHS coefficients in linear program (2) are equal to 1 and therefore, there even exists an optimal 0-1 price vector. The following dynamic program[4] obtains the optimal set of winners and the optimal 0-1 price vector in $\mathrm{O}(m^3 n)$ time.

[4] Note that Guruswami et al. [13] introduce yet another $\mathrm{O}(B^{B+2} n^{B+3})$ time dynamic program to solve the profit maximization problem on a path with unlimited edge capacities and a constant upper bound B on the valuations, which is $\mathrm{O}(n^4)$ in case $b_j = 1$ for all $j \in J$.

Without loss of generality, we assume that the edges on the path are indexed consecutively. In the dynamic programming algorithm for the problem with $b = 1$, for every customer j, let s_j denote the edge in path E_j with the smallest index and t_j the edge in E_j with the largest index. For all edges $k, \ell \in E$ with $k < \ell$, let $R(k, \ell)$ be the total optimal revenue obtained by the customers for whom $t_j \leq \ell$, given that $p_k = p_\ell = 1$ and $p_{k+1} = \ldots = p_{\ell-1} = 0$. We define $R(0, \ell) = |\{j \in J : t_j = \ell\}|$ for all $\ell = 1, \ldots, m$. Then, given $R(e, k)$ for all $e = 0, \ldots, k-1$ and $k < \ell$, we compute the revenue $R(k, \ell)$ by finding the maximum sum of $R(e, k)$ and the contribution of all customers $j \in J$ whose path E_j contains either edge k or ℓ (not both), but not edge e. Formally,

$$R(k, \ell) = \max_{0 \leq e < k} \left\{ R(e, k) + \sum_{j \in J} r_j(e, k, \ell) \right\}, \quad \text{where}$$

$$r_j(e, k, \ell) = \begin{cases} 1 & \text{if } e < s_j \leq k \text{ and } k < t_j < \ell \\ 1 & \text{if } k < s_j \text{ and } t_j = \ell \\ 0 & \text{otherwise.} \end{cases}$$

To obtain the optimal solution for arbitrary uniform budget b, it is sufficient to multiply p_e obtained in the case of unit budgets by b, for every edge $e \in E$.

The computation time of this dynamic program is $\mathrm{O}(m^3 n)$. The optimal revenue is equal to $\max\{R(k, \ell) : 0 \leq k < \ell \leq m\}$. □

Corollary 1. *Consider the problem on a path with unlimited edge capacities. If for every customer $j \in J$, the budget b_j is such that $b \leq b_j \leq \delta b$ for $b \geq 0$ and $\delta \geq 1$, then there exists a polynomial time δ-approximation.*

Proof. By rounding down all budgets to b, the loss in the optimal revenue is at most a factor of δ. Therefore, the corollary follows straightforwardly from Theorem 2. □

For the same problem with arbitrary budgets, we can derive a polynomial time $\mathrm{O}(\log_2 B)$-approximation algorithm, where $B = \max_{j \in j} b_j$ is the maximum budget. To achieve this, let us define subproblems such that the ℓ^{th} subproblem contains customers $\{j \in J : 2^{\ell-1} \leq b_j < 2^\ell\}$. Using Corollary 1, we find a 2-approximate solution for each of these $\log_2 B$ subproblems and we derive the following corollary.

Corollary 2. *There exists a $(2 \log_2 B)$-approximation algorithm for the profit maximization problem on a path with unlimited edge capacities.*

2.4 Limited Edge Capacities

In the problem on a path with limited edge capacities, each edge $e \in E$ can accommodate no more than c_e customers. Let $C \geq \max_{e \in E} c_e$ be an upper bound on the capacity of any edge. We show that we can solve this problem in time $\mathrm{O}(n^{2C} B^{2C} m)$ by finding a longest path in an acyclic digraph.

We create an m-layered digraph D with an additional source s and sink t, layers 0 and $m+1$, respectively. There are arcs only from layer e to $e+1$, for $e = 0, \ldots, m$. Hence, in any $s-t$ path, there are exactly $m+2$ nodes. In every node in layer e, corresponding to edge e, we store all winners j that are accommodated by edge e. Moreover, we store the respective total amounts all these winners spend on all edges (network links) in their respective path E_j up to and including edge e. Any node x (more precisely, the path $s-x$) in the digraph represents a feasible partial solution. Arcs from node x of layer e to node y of layer $e+1$ 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 e to a node of layer $e+1$ is equal to the profit earned on edge $e+1$, that is, the total amount that the corresponding winners pay for edge $e+1$. Therefore, the weight of the longest $s-t$ path in digraph D is equal to the maximum total profit. Moreover, the set of winners can be reconstructed from the longest $s-t$ path. Algorithm 1 shows a more formal description.

Theorem 3. *There is a $O(n^{2C} B^{2C} m)$ time algorithm for the profit maximization problem on a path. Here, C is an upper bound on the edge capacities.*

Proof. Consider an arbitrary $s-t$ path $\mathcal{P}$ in digraph D. Abusing notation, let (W^e, h^e) be the nodes on $\mathcal{P}$. Set W^e is thus the set of customers that are accommodated by edge e, and h^e is the vector of the total amounts these customers pay up to and including edge e. By definition of the nodes and arcs of the digraph, no customer j will be accommodated to an edge outside her requested path E_j. For any customer j, consider an edge $e \in E_j$ such that $j \in W^e$. That is, edge e accommodates customer j. By condition (1) of the arc definition, all other edges of path E_j must accommodate customer j as well. Next, by definition we have for any node (W^e, h^e) that $|W^e| \leq c_e$. Hence, the edge capacities are satisfied. Finally, let us consider the budget constraint of customer j. We know that

Algorithm 1: Dynamic Programming Algorithm

Input: The profit maximization problem on a path with maximum capacity of any edge at most C
Output: Accommodation of customers to edges and edge prices p_e
begin (construction of digraph D)

nodes: For each edge $e \in E$, we introduce a layer of nodes: Denote by J^e the set of customers with $e \in E_j$. By $K^e = (j_1, j_2, \ldots, j_k)$ we denote any (sorted) subset of J^e of cardinality k, where $k \leq \min\{c_e, |J^e|\} \leq C$. Define $H_j := \{0, 1, \ldots, b_j\}$ as the possible total amount customer $j \in K^e$ can spend for edges $\{1, \ldots, e\} \cap E_j$. Let $h^e \in H_{j_1} \times H_{j_2} \times \cdots \times H_{j_k}$ be a vector denoting how much each customer j spends for items $\{1, \ldots, e\} \cap E_j$, for each $j \in K^e$. If $K^e = \emptyset$, we let $h^e = 0$. Let all such pairs (K^e, h^e) be the nodes in layer e of D, for $e = 1, \ldots, m$. Let s and t denote source and sink. To unify notation, assume $s = (\emptyset, 0)$ and $t = (\emptyset, 0)$;

arcs: Insert an arc a^e from node (K^{e-1}, h^{e-1}) to node (K^e, h^e), for $e = 2, \ldots, m$, if:
(1) For all $j \in K^{e-1}$ with $e \in E_j$, $j \in K^e$, and for all $j \in J^{e-1} \setminus K^{e-1}$ with $e \in E_j$, $j \notin K^e$.
(2) There exists a unique integral value $d \geq 0$ such that $d = h^e_j - h^{e-1}_j$ for all $j \in K^{e-1} \cap K^e$, and $d = h^e_j$ for all $j \in K^e \setminus K^{e-1}$.
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^e that connects (K^{e-1}, h^{e-1}) and (K^e, h^e): If $K^e = \emptyset$, we let the length of arc a^e be $\ell(a^e) = 0$, and if $K^e \neq \emptyset$, we let the length of arc a^e be $\ell(a^e) = d|K^e|$, 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 $j \in K^e$ with $(K^e, h^e) \in \mathcal{P}$, edge e accommodates customer j. The price p_e for edge e equals $\ell(a^e)/|K^e|$, where a^e is the arc from path $\mathcal{P}$ that connects nodes (K^{e-1}, h^{e-1}) and (K^e, h^e);

$E_j = \{k, \ldots, l\}$ for some edges $k \leq l$. We have that

$$\sum_{e \in E_j} p_e = \sum_{e=k}^{l} p_e = \sum_{e=k}^{l} \frac{\ell(a^e)}{|W^e|} = h^k_j + \sum_{e=k+1}^{l} (h^e_j - h^{e-1}_j) = h^l_j \leq b_j \,.$$

The third equality holds due to condition (2) of the arc definition, and the last inequality holds because $h^l_j \in H_j = \{0, 1, \ldots, b_j\}$.

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

$$\sum_{a^e \in \mathcal{P}} \ell(a^e) = \sum_{e \in E} p_e |W^e| = \sum_{j \in W} \sum_{e \in E_j} p_e \,.$$

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 $\mathrm{O}(n^{2C} B^{2C} m)$, we estimate the size of digraph D. For every edge $e \in E$, there are at most $\mathrm{O}(n^C)$ different sets K^e and at most $\mathrm{O}(B^C)$ different vectors h^e. Thus, per edge $e \in E$, we have at most $\mathrm{O}(n^C B^C)$ nodes (K^e, h^e). For any $e \in E$, every node (K^e, h^e) is connected to at most $\mathrm{O}(n^C B^C)$ nodes (K^{e+1}, h^{e+1}). So, per layer e, there are at most $\mathrm{O}(n^{2C} B^{2C})$ arcs to layer $e + 1$, which means that there are

at most $O(n^{2C}B^{2C}m)$ 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.

2.5 FPTAS

We next show how to turn the dynamic programming algorithm into a fully polynomial time approximation scheme (FPTAS); that is, for any $\varepsilon > 0$, we have 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 the customers' budgets are $b'_j := \lfloor b_j/\beta \rfloor$ where $\beta := (\varepsilon B/(2n^2))$ for $\varepsilon > 0$. Note that without loss of generality, we may assume that β is integer.

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

Proof. Let (W, p) be a feasible solution to the original instance with profit $\Pi(W, p)$. Let $p''_e = \lfloor p_e/\beta \rfloor$ for all edges $e \in E$. Note that $(p_e/\beta) - 1 < p''_e \leq (p_e/\beta)$. For the original instance we have for every winner $j \in W$, $\sum_{e \in E_j} p_e \leq b_j$, and it follows that

$$\sum_{e \in E_j} p''_e = \sum_{e \in E_j} \left\lfloor \frac{p_e}{\beta} \right\rfloor \leq \left\lfloor \frac{\sum_{e \in E_j} p_e}{\beta} \right\rfloor \leq \left\lfloor \frac{b_j}{\beta} \right\rfloor = b'_j \,.$$

Hence, the solution (W, p'') is feasible to the rounded instance. Finally, we have

$$\Pi(W, p'') = \sum_{j \in W} \sum_{e \in E_j} p''_e > \sum_{j \in W} \sum_{e \in E_j} \left(\frac{p_e}{\beta} - 1 \right) \geq \frac{1}{\beta}\Pi(W, p) - mn.$$

□

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

Proof. Let (W', p') be a solution to the rounded instance with revenue $\Pi(W', p')$. Let $\tilde{p}_e = p'_e\beta$ be prices in the original instance for all edges $e \in E$. This is integer because

p'_e and β are integer. Then the budget constraint for every customer $j \in W'$ is satisfied, because

$$\sum_{e \in E_j} \tilde{p}_e = \beta \sum_{e \in E_j} p'_e \leq \beta b'_j = \beta \left\lfloor \frac{b_j}{\beta} \right\rfloor \leq b_j \,.$$

Hence, the same set of customers W can get their requested path, and solution $(W', \tilde{p})$ is feasible to the original instance. The revenue can be written as

$$\Pi(W', \tilde{p}) = \sum_{j \in W'} \sum_{e \in E_j} \tilde{p}_e = \sum_{j \in W'} \sum_{e \in E_j} (p'_e \beta) = \beta \sum_{j \in W'} \sum_{e \in E_j} p'_e = \beta \Pi(W', p').$$

□

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

Theorem 4. *There exists an FPTAS for the profit maximization on a path with edge capacities bounded by a constant.*

Proof. Let (W, p) and (W', p') be the optimal solutions to the original and rounded instances, respectively. Consider solution $(W', \tilde{p})$ to the original instance, where $\tilde{p}_e = \beta p'_e$ for all edges $e \in E$, and solution (W, p'') to the rounded instance, where $p''_e = \lfloor p_e / \beta \rfloor$ for all edges $e \in E$. An application of the previous two lemmas now yields

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

where the first inequality holds due to optimality of (W', p') for the rounded instance. Note that for n customers there are at most $2n$ distinct endpoints of their requested paths. Without loss of generality, we identify a set of items between two consecutive endpoints with a single item. Therefore, the number of distinct items m is at most $2n - 1$. Since the maximum profit $\Pi(W, p)$ is at least the maximum budget B, we have that $\Pi(W', \tilde{p}) > (1 - \varepsilon)\Pi(W, p)$.

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

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

3 Selling a Cycle

The complexity result and the dynamic program we obtained for the profit maximization problem on a path can be extended to the problem on a cycle.

Corollary 3. *The profit maximization problem on a cycle is NP-hard.*

In fact, the proof of Theorem 1 immediately results in NP-hardness for the problem on a cycle, as the two ends of the path can be connected. Customers still request the same subpaths after this transformation.

Theorem 5. *The profit maximization problem on a cycle can be solved in time* $O(n^{3C}B^{3C}m)$.

Proof. We adapt the dynamic program of Algorithm 1. Choose any edge $e \in E$ to start the procedure. Other than in Algorithm 1, we now start in layer 1, and include all possible nodes (K^e, h^e) in this layer; these nodes are created similar as in Algorithm 1, where K^e is a subset of the customers for which $e \in E_j$, and h^e represents the total amount spent on edges up to and including edge e. For example, for a customer j with $E_j = \{e-2, e-1, e, e+1\}$, h^e represents the amount this customer spends on edges $e-2$, $e-1$ and e, regardless of the fact that we do not yet know how much exactly she spends on edges $e-2$ and $e-1$.

Arcs between layers e and $e+1$ exist if the criteria (1) and (2) of Algorithm 1 are satisfied. In the final layer, $m+1$, there are exactly the same nodes as in layer 1 to complete the cycle. For every node (K^e, h^e) in layer 1, we find the longest path to the *same* node (K^e, h^e) in layer $m+1$. Among all longest paths, we select the one with the highest total value. This value is equal to the maximum total profit, and the set of winners can be reconstructed from this longest path as well.

The size of the digraph is similar as before, that is, there are $\mathrm{O}(n^C B^C)$ nodes per layer, $\mathrm{O}(n^{2C}B^{2C})$ arcs between any two consecutive layers, and therefore a total of $\mathrm{O}(n^{2C}B^{2C}m)$ arcs. Finding a longest path in the digraph is linear in the number of arcs since the digraph is acyclic [2]. As we have to find the longest path for each of the $\mathrm{O}(n^C B^C)$ nodes in the first layer, all longest paths can be found in $\mathrm{O}(n^{3C}B^{3C}m)$, which is the total time needed to find the optimal solution to the profit maximization problem on a cycle with limited edge capacities. □

An interesting extension is the problem in which each customer j requests a *connection* between two vertices instead of a specific path. In this case, for each customer we decide

whether to provide her with a connection or not. If we do establish a connection then we have the freedom to decide on the direction of that connection. Notice that we do not guarantee envy-freeness in this extension of the problem, hence customers are not necessarily assigned to the cheaper of the two alternatives.

Theorem 6. *The profit maximization problem on a cycle is NP-hard, even if customers are indifferent to the direction used for the connection.*

Proof. We reduce from the problem on a path. First, we include an additional edge that connects the two end vertices of the original path to create a cycle. Then, we introduce one extra customer that requests a connection between the two original end vertices and has a budget larger than all other customers' budgets. This forces the price on the new edge to be expensive enough in the optimal solution such that no other customer can afford using this edge. Therefore, all customers will still request the same paths as in the instance on a path. □

To solve this problem, we further adapt the previous dynamic programming approach to include the direction of the connection between the requested vertices for every customer. Therefore, given set $K^e = (j_1, \ldots, j_k)$ of customers with $e \in E_j$, we introduce vector $dir^e \in \{0,1\}^k$ to denote for each customer $j \in K^e$ the direction of the connection between the requested vertices; 0 for clockwise, 1 for counter-clockwise. Thus, in layer 1, corresponding to chosen edge $e \in E$, we have all possible nodes (K^e, h^e, dir^e).

Arcs are again created if conditions (1) and (2) of Algorithm 1 are met. However, yet another condition should be satisfied, which assures that the connection for each customer has only one direction. Thus, for all $j \in K^e \cap K^{e+1}$, there is an arc if also $dir^e_j = dir^{e+1}_j$. In the final layer, $m+1$, there are again the same nodes as in layer 1. Finally, after the complete digraph is created, we find for every node (K^e, h^e, dir^e) in layer 1 the longest path to the corresponding node (K^e, h^e, dir^e) in layer $m+1$. Among all longest paths, we select the one with the highest total value. This value is equal to the maximum total profit, and the set of winners can be reconstructed from this longest path as well.

Theorem 7. *The profit maximization problem on a cycle can be solved in time $O(n^{3C}B^{3C}4^C m)$, even if customers are indifferent to the direction used for the connection.*

Proof. By a similar reasoning as in the proof of Theorem 3, and the extra condition that each customer that gets a connection between the requested vertices has a connection

oriented in only one direction, the longest of all longest paths between similar nodes defines the profit of the optimal solution.

For every edge $e \in E$, there are at most $O(n^C)$ different sets K^e, at most $O(B^C)$ different vectors h^e, and at most $O(2^C)$ different vectors dir^e. Thus, per edge $e \in E$, we have at most $O(n^C B^C 2^C)$ nodes (K^e, h^e, dir^e). For any $e \in E$, every node (K^e, h^e, dir^e) is connected to at most $O(n^C B^C)$ nodes $(K^{e+1}, h^{e+1}, dir^{e+1})$, as there only exists an arc when the values dir^e_j and dir^{e+1}_j are the same for all customers $j \in K^e \cap K^{e+1}$. So, there are at most $O(n^{2C} B^{2C} 2^C)$ arcs between layers e and $e+1$, which means that there are at most $O(n^{2C} B^{2C} 2^C m)$ arcs in the digraph. The computation time to find the longest path in the digraph is linear in the number of arcs since the digraph is acyclic [2]. We have to find the longest path for each node in layer 1, which takes $O(n^{3C} B^{3C} 4^C m)$ time. □

4 Selling Graphs with Unit Edge Capacities

In this section we assume that the capacity of any edge is one, that is, $c_e = 1$ for all $e \in E$. We consider four types of graphs, namely paths, cycles, trees and grids.

Theorem 8 ([18, Theorem 10]). *The profit maximization problem on a path with unit edge capacities can be solved in $O(n^2)$ time.*

In fact, the result of Theorem 8 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 quadratic time [16].

Theorem 9 ([18, Corollary 11]). *The profit maximization problem on a cycle with unit edge capacities can be solved in $O(n^3)$ time.*

The problem on a cycle with edge capacities one reduces to finding a maximum weight independent set in a circular arc graph, which is known to be solvable in cubic time [10]. The problem in which the customers are indifferent to the direction of their requested connection can be solved by straightforward Algorithm 2, also in $O(n^3)$ time.

4.1 Trees

Guruswami et al. [13] show that the problem with unlimited edge capacities is APX-hard even on star graphs. Contrasting this complexity result, we prove that if the capacity of each edge is exactly one, the problem on a tree can be solved in polynomial time. (Again, recall that we do not require the solution to be envy-free.)

Algorithm 2:

Let $\overrightarrow{E_j}$ be the edges contained in the clockwise connection for customer j, and $\overleftarrow{E_j}$ the edges in the counterclockwise connection.
For $j = 1, \ldots, n$
Assign $\overrightarrow{E_j}$ to j, and solve the problem on path $E \setminus \overrightarrow{E_j}$ with customers $J \setminus j$. Let $\overrightarrow{\pi_j}$ be the profit.
Assign $\overleftarrow{E_j}$ to j, and solve the problem on path $E \setminus \overleftarrow{E_j}$ with customers $J \setminus j$. Let $\overleftarrow{\pi_j}$ be the profit.
Let $\pi_j = \max\{\overrightarrow{\pi_j}, \overleftarrow{\pi_j}\}$.
$\pi = \max_{j \in J}\{\pi_j\}$.

Theorem 10. *The profit maximization problem on a tree with unit edge capacities can be solved in $O(n^4)$ time.*

Proof. Consider the graph $H = (J, E')$ where $(j, k) \in E'$ if and only if $E_j \cap E_k \neq \emptyset$, for two customers $j, k \in J$. Since $\{E_j : j \in J\}$ is a collection of simple paths in a tree, graph H is an *EPT graph* [11]. Since the capacity of each edge is 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 obtained straightforwardly by setting the price of one arbitrary edge of E_j to b_j, and setting the prices of all other edges in E_j to 0, $j \in W$. 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 [20]. The algorithm is a recursive procedure that decomposes the problem on the basis of clique separators. The polynomial computation time is a consequence of the fact that the atoms, that is, 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 $\mathrm{O}(n^3)$ time by Edmonds' algorithm [7]. As the total number of atoms is $\mathrm{O}(n)$, the total time complexity is bounded by $\mathrm{O}(n^4)$. □

4.2 Grids

Demaine et al. [6] show that the profit maximization problem with unlimited capacities where the customers request arbitrary subsets of network links is hard to approximate within a (semi-)logarithmic factor. If we restrict the requested subsets of links to be a path of edges in a general graph, then the problem is APX-hard even under strong restrictions [5, 13]

Here we show that if the capacities of the edges are bounded, then we can derive an even stronger inapproximability result for a very restricted class of graphs and customers' requests.

Theorem 11. *For all $\varepsilon > 0$, approximating the profit maximization problem on a grid with unit edge capacities within a factor $n^{1-\varepsilon}$ is NP-hard, even with unit budgets, and when each edge is an element of at most two requested paths. 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 $H = (V', E')$, 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 [21].

Let $V' = \{v_1, \ldots, v_n\}$ and $E' = \{a_1, \ldots, a_m\}$. We construct the instance of the profit maximization problem as follows. We create an $(n+1) \times (2m+2)$ grid, that is, a simple graph with $2nm + 2n + 2m + 2$ vertices $\{(j, e) : 1 \leq j \leq n+1, 1 \leq e \leq 2m+2\}$, where (j, e) and (j', e') are adjacent if $|j' - j| + |e' - e| = 1$. Let horizontal layer $j \in \{1, \ldots, n\}$ correspond to vertex $v_j \in V'$, and let the edge $((n+1, 2e), (n+1, 2e+1))$ in the grid correspond to edge $a_e \in E'$. Next, for each vertex $v_j \in V'$, we introduce a customer in the profit maximization problem with a requested path defined by the following simple path in the grid graph. The path starts at point $(j, 1)$ and ends at point $(j, 2m+2)$ following the layer j everywhere except for the edges $((j, 2e), (j, 2e+1))$ such that $v_j \in a_e$. These edges are substituted by vertical detours, passing through edges $((n+1, 2e), (n+1, 2e+1))$; see Figure 1 for an example. We complete the construction setting the budget of each customer to 1.

We claim that there exists an independent set of cardinality K in H if and only if there exists a solution to the profit maximization problem with total profit K. By construction, two paths in the grid corresponding to adjacent vertices in H must share some edge $a \in E'$ in layer $(n+1)$. Since the capacity of edge a 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 H. Now, consider an independent set S in H 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 to the profit maximization problem where S defines the set of winners and each edge

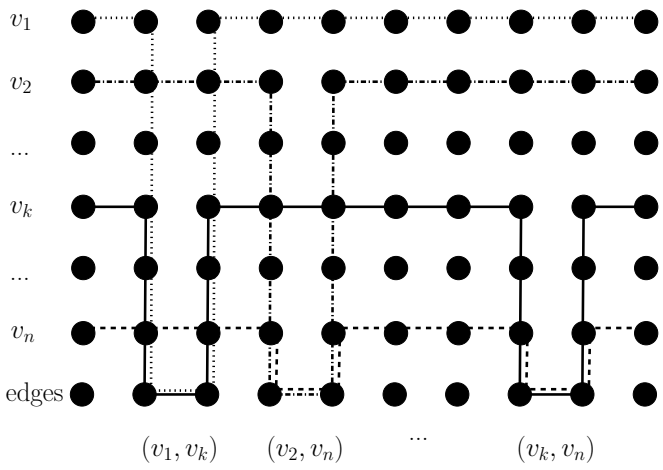

Fig. 1. Grid graph from the reduction.

$e \in E$ accommodates at most one customer. For each $v_j \in S$ we set the price of the grid edge $((j,1),(j,2))$ to 1, for each $v_j \notin S$ we set the price of $((j,1),(j,2))$ to 2, and for all other edges of the grid we set the prices to 0. In the constructed solution to 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 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 her budget. Therefore, the theorem holds also if we require the solution to be envy-free. □

5 Conclusion

The currently best known negative result on the tractability of the profit maximization problem on a path is NP-hardness [5]. Even though several FPTAS's (including the one of this paper) exist whenever certain parameters are constantly bounded, the best known positive results for the general case are a logarithmic approximation [3, 13] and a quasi-polynomial time approximation scheme [8]. It thus remains an intriguing open problem whether it is possible 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. 7th ACM Conference on Electronic Commerce, ACM, 2006, pp. 29–35.
4. M. F. Balcan, A. Blum, J. D. Hartline, and Y. Mansour, Mechanism design via machine learning, Proc. 46th IEEE Symposium on Foundations of Computer Science, FOCS, 2005, pp. 605–614.
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. K. Elbassioni, R. Sitters, and Y. Zhang, A quasi-ptas for profit-maximizing pricing on line graphs, Proc. 15th Annual European Symposium on Algorithms (L. Arge and E. Welzl, eds.), Lecture Notes in Computer Science, vol. 4698, Springer, 2007, pp. 451–462.
9. A. V. Goldberg, J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright, Competitive auctions, Games and Economic Behavior 55 (2006), 242–269.
10. M. C. Golumbic and P. L. Hammer, Stability in circular arc graphs, Journal of Algorithms 9 (1988), 56–63.
11. 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.
12. A. Grigoriev, J. van Loon, M. Sviridenko, M. Uetz, and T. Vredeveld, Bundle pricing with comparable items, Proc. 15th Annual European Symposium on Algorithms (L. Arge and E. Welzl, eds.), Lecture Notes in Computer Science, vol. 4698, Springer, 2007, pp. 475–486.
13. 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.
14. 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.
15. D. Lehman, L. I. O'Callaghan, and Y. Shoham, Truth revelation in approximately efficient combinatorial auctions, Journal of the ACM 49(5) (2002), 1–26.
16. R. H. Möhring, Algorithmic aspects of comparability graphs and interval graphs, Graphs and Order (I. Rival, ed.), Reidel, Dordrecht, 1985, pp. 41–101.
17. G. L. Nemhauser and L. A. Wolsey, Integer and combinatorial optimization, John Wiley & Sons, New York, 1988.
18. M. H. Rothkopf, A. Pekec, and R. M. Harstad, Computationally manageable combinatorial auctions, Management Science 44(8) (1998), 1131–1147.
19. P. Rusmevichientong, B. Van Roy, and P. W. Glynn, A nonparametric approach to multi-product pricing, Operations Research 54(1) (2006), 82–98.
20. R. E. Tarjan, Decomposition by clique separators, Discrete Mathematics 55 (1985), 221–232.
21. D. Zuckerman, Linear degree extractors and the inapproximability of MAX CLIQUE and CHROMATIC NUMBER, Proc. 38th annual ACM Symposium on Theory of Computing, ACM-SIAM, 2006, pp. 681–690.