On Profit-Maximizing Pricing for the Highway and Tollbooth Problems

Khaled Elbassioni[1], Rajiv Raman[1], Saurabh Ray[1], and René Sitters[2]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
{elbassio,rraman,saurabh}@mpi-inf.mpg.de
[2] Department of Econometrics and Operations Research, VU University, Amsterdam, the Netherlands
rsitters@feweb.vu.nl

Abstract. In the *tollbooth problem on trees*, we are given a tree $\mathbf{T} = (V, E)$ with n edges, and a set of m customers, each of whom is interested in purchasing a path on the graph. Each customer has a fixed budget, and the objective is to price the edges of $\mathbf{T}$ such that the total revenue made by selling the paths to the customers that can afford them is maximized. An important special case of this problem, known as the *highway problem*, is when $\mathbf{T}$ is restricted to be a line. For the tollbooth problem, we present an $O(\log n)$-approximation, improving on the current best $O(\log m)$-approximation. We also study a special case of the tollbooth problem, when all the paths that customers are interested in purchasing go towards a fixed root of $\mathbf{T}$. In this case, we present an algorithm that returns a $(1 - \epsilon)$-approximation, for any $\epsilon > 0$, and runs in quasi-polynomial time. On the other hand, we rule out the existence of an FPTAS by showing that even for the line case, the problem is strongly NP-hard. Finally, we show that in the *discount model*, when we allow some items to be priced below zero to improve the overall profit, the problem becomes even APX-hard.

1 Introduction

Consider the problem of pricing the bandwidth along the links of a network such that the revenue obtained from customers interested in buying bandwidth along certain paths in the network is maximized. Suppose that each customer declares a set of paths she is interested in buying, and a maximum amount she is is willing to pay for each path. The network service provider's objective is to assign single prices to the links such that the total revenue from customers who can afford to purchase their paths is maximized. Recently, numerous papers have appeared on the computational complexity of such pricing problems [1,6,5,7,8,9,10,11,12,15,13,16,17,18].

A special case of this problem, where each customer is interested in purchasing only a single path (*single-minded*), and where there is no upper bound on the number of customers purchasing each link (*unlimited supply*) was studied by Guruswami et al. [16], under the name of *tollbooth problem.* The authors of [16]

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 275–286, 2009.

showed that the problem is already APX-hard when the network is restricted to be a tree, and also presented a polynomial time algorithm for the case when all paths start at a certain root of the tree. In [16], the authors also studied the *highway problem*, a further restriction where the tree is a path, and gave polynomial time algorithms when either the budgets are bounded and integral, or all paths have a bounded length.

In this paper, we continue the study of these problems. For the tollbooth problem, a uniform pricing gives an approximation factor of $O(\log n + \log m)$, where n and m are respectively the number of edges of the tree and the number of customers. This result applies in fact for general sets [16], and not necessarily paths of a network, and even in the non single-minded case [4]. Very recently, and more generally, Cheung and Swamy [10] gave an algorithm that, given any LP-based α-approximation algorithm for maximizing the social welfare under limited supply, returns a solution with profit within a factor of $\alpha \log u_{max}$ of the maximum, where u_{max} is the maximum supply of an item. In particular, this gives an $O(\log m)$-approximation for the tollbooth problem on trees. In this paper, we give an $O(\log n)$-approximation which is an improvement over the $O(\log m)$ since $n \leq 3m$ can be always assumed (see Section 2). While the problem is APX-hard even in the very simple case of a star [16], we show that if all the paths are going towards (but not necessarily starting at) a certain root, then a $(1 - \epsilon)$-approximation can be obtained in quasi-polynomial time. This result extends a recently developed quasi-PTAS [12] for the highway problem, and uses essentially the same technique. However, there is a number of technical issues that have to be resolved for this technique to work on trees; most notably is the use of the Separator Theorem for trees, and the modification of the price-guessing strategy to allow only for *one-sided* guesses.

The existence of a quasi-PTAS for the highway problem indicates that a PTAS or even an FPTAS is still a possibility, since the problem was only known to be weakly NP-hard [8]. In the last section of this paper, we show that the highway problem is indeed strongly NP-hard and hence admits no FPTAS unless P=NP.

Balcan et al. [3] considered a model in which some items can be priced below zero (in the form of a discount) so that the overall profit is maximized. They gave a 4-approximation for the uniform budgets case, and a quasi-PTAS for a special case in which there is an optimal pricing that has only a bounded number of negatively priced items. Here we show that the existence of a quasi-PTAS in the general case is highly unlikely, by showing that the problem is APX-hard.

In the next section, we give a formal definition of the problem. In Section 3, we give an $O(\log n)$ approximation for trees and in Section 4 we give a quasi-PTAS for the case of uncrossing paths. We describe our hardness results in Section 5. We conclude in Section 6. Due to lack of space, most proofs have been omitted from this extended abstract.

2 The Tollbooth Problem on Trees

2.1 Notation

Let $\mathbf{T} = (V, E)$ be a tree. We assume that we are given a (multi)set of paths $\mathcal{I} = \{I_1, \ldots, I_m\}$, defined on the set of edges E, where $I_j = [s_j, t_j] \subseteq E$ is the path connecting s_j and t_j in $\mathbf{T}$. For $I_j \in \mathcal{I}$, we denote by $B(I_j) \in \mathbb{R}_+$ the *budget* of path I_j, i.e., the maximum amount of money customer j is willing to pay for purchasing path I_j. In the *tollbooth problem*, denoted henceforth by TB, the objective is to assign a price $p(e) \in \mathbb{R}_+$ for each edge $e \in E$, and to find a subset $\mathcal{J} \subseteq \mathcal{I}$, so as to maximize $\sum_{I \in \mathcal{J}} p(I)$ subject to the budget constraints

$$p(I) \leq B(I), \quad \text{for all } I \in \mathcal{J}, \tag{1}$$

where, for $I \in \mathcal{I}$, $p(I) = \sum_{e \in I} p(e)$.

For a node $w \in V$, let $\mathcal{I}[w] \subseteq \mathcal{I}$ be the set of paths that pass through w. In section 4, we will assume that the tree is rooted at some node $\mathbf{r} \in V$. The depth of $\mathbf{T}$, denoted $d(\mathbf{T})$, is the length of the longest path from the root $\mathbf{r}$ to a leaf. For a node $w \in V$, we denote by $\mathbf{T}(w)$, the subtree of $\mathbf{T}$ rooted at w (excluding the path from the parent of w to $\mathbf{r}$), and for a subtree $\mathbf{T}'$ of $\mathbf{T}$ we denote by $V(\mathbf{T}'), E(\mathbf{T}')$ and $\mathcal{I}(\mathbf{T}')$ the vertex set, edge set, and set of intervals contained completely in $\mathbf{T}'$, respectively.

2.2 Preliminaries

In the following sections, we denote by $p^* : E \mapsto \mathbb{R}_+$ an optimal set of prices, and by OPT $\subseteq \mathcal{I}$ the set of intervals purchased in this optimum solution. For a subset of intervals $\mathcal{I}' \subseteq \mathcal{I}$, and a price function $p : E \mapsto \mathbb{R}_+$, we denote by $p(\mathcal{I}') = \sum_{I \in \mathcal{I}'} p(I)$ the total price of intervals in $\mathcal{I}'$.

It easy to see that $n \leq 3m$ may be assumed without loss of generality. Indeed, if we root the tree at some vertex $\mathbf{r}$, then for every vertex $v \in V$, we may assume that there is either an interval $I \in \mathcal{I}$ beginning at v or an interval $I \in \mathcal{I}$ that passes through two different children of v; otherwise, every interval through v must contain its parent u (unless $v = \mathbf{r}$ in which case all edges incident to $\mathbf{r}$ can be contracted), and hence we can contract the edge $e = \{u, v\}$ and increase by $p^*(e)$ the prices of each the edges $\{v, v'\}$ for each child v' of v.

Let $\epsilon > 0$ be a given constant.

Proposition 1 ([12]). *Let p^* be an optimal solution for a given instance of* TB, *and $\epsilon > 0$ be a given constant. Then there exists a price function $\tilde{p} : E \mapsto \mathbb{R}_+$ for which*

(i) $\tilde{p}(e) \in \{0, 1, \ldots, P\}$, for every $e \in E$, where $P = nm/\epsilon$,
(ii) $\tilde{p}(I) \leq \frac{B(I)}{1+\epsilon}$, for every $I \in$ OPT, and
(iii) $\tilde{p}(\text{OPT}) \geq (1 - 2\epsilon)p^*(\text{OPT})$.

We shall call the set of prices $\tilde{p}$ satisfying the conditions of Proposition 1, ϵ-*optimal prices.*

We will make use of the following well-known separator result for trees.

Proposition 2. *Let $T = (V, E)$ be a tree. Then there exists a node v (called separator node) with the following property: Let $s_1, \ldots, s_r$ be the sizes of the components obtained by deleting v from $\mathbf{T}$, then there is a subset $S \subseteq [r] \stackrel{\text{def}}{=} \{1, \ldots, r\}$ such that*

$$\lfloor \frac{n}{3} \rfloor \leq \sum_{i \in S} s_i \leq \lceil \frac{2n}{3} \rceil. \qquad (2)$$

Such a separator can be found in linear time.

This gives a recursive partitioning of $\mathbf{T}$ in the following standard way: Let v_0 be a separator vertex in $\mathbf{T}$ and $\mathbf{T}_1, \ldots, \mathbf{T}_r$ be the components of $\mathbf{T} - v_0$. Recursively, find separator vertices $v_1, \ldots v_r$ in $\mathbf{T}_1, \ldots, \mathbf{T}_r$. We say that node v_0 has level$(v_0) = 1$, nodes $v_1, \ldots, v_r$ have level 2, and in general if node v is a separator vertex in the subtree $\mathbf{T}'$ obtained by deleting one-higher level separator vertex v' then level$(v) =$ level$(v') + 1$. By (2), the maximum number of levels k in this decomposition is at most $\log_{3/2} n$. We shall denote by $\mathcal{N}(\mathbf{T})$ the set of separator nodes used in the full decomposition of $\mathbf{T}$.

3 An $O(\log n)$ Approximation for the Tollbooth Problem on Trees

In this section, we prove the following theorem.

Theorem 1. *There is a deterministic $O(\log n)$-approximation algorithm for* Tb.

The proof goes along the same lines used in [2] to obtain an $O(\log n)$-approximation for the highway problem. The algorithm consists of 3 main steps: Partitioning, "randomized cut", and then dynamic programming. We can then derandomize it to obtain a deterministic algorithm. We give the details below.

We say that the given set of paths $\mathcal{I}$ is *rooted*, if all the paths in $\mathcal{I}$ start at some node $\mathbf{r}$, called the root of $\mathbf{T}$. We will also make use of the following theorem.

Theorem 2 ([16]). *The tollbooth problem on rooted paths can be solved in polynomial time using dynamic programming.*

For $i = 1, \ldots, k$, let

$$\mathcal{I}(i) = \{I \in \mathcal{I} : \ i \text{ is the smallest level of a separator } v \in \mathcal{N}(\mathbf{T}) \text{ contained in } I\}.$$

Then $\mathcal{I} = \cup_{i \in [k]} \mathcal{I}(i)$ and $I \cap J = \emptyset$ for all $I, J \in \mathcal{I}(i)$ that contain distinct separators at level i. Let (Opt, p^*) be an optimal solution. Then, $p^*(\text{Opt}) = \sum_{i=1}^{k} p^*(\text{Opt} \cap \mathcal{I}(i))$. Thus if we solve k independent problems on each of the sets $\mathcal{I}(i)$, $i = 1, \ldots, k$, and take the solution with maximum revenue, we get a solution of value at least $p^*(\text{Opt})/k$. Thus it remains to show the following result.

Theorem 3. *Let v be a node of $\mathbf{T}$, and suppose that all the paths in $\mathcal{I}$ go through v. Then a solution $(\mathcal{J}, p)$ of expected value $p(\mathcal{J}) \geq p^*(\text{Opt})/8$ can be found in polynomial time.*

Proof. Let $v_1, \ldots, v_r$ be the nodes adjacent to v. Note that each path $I \in \mathcal{I}$ can be divided into two sub-paths starting at v; we denote them by I_1 and I_2. We use the following procedure.

1. Let $X \subseteq \{v_1, \ldots, v_r\}$ be a subset obtained by picking each v_i randomly and independently with probability 1/2.
2. Let $\mathcal{I}' = \{I_j : I \in \mathcal{I}, j \in \{1,2\}, I_j$ contains exactly one vertex of $X\}$.
3. Use dynamic programming (cf. Theorem 2) to get an optimal solution $(\mathcal{J}, p)$ on the instance defined by $\mathcal{I}'$ and the tree $\mathbf{T}'$ with root v and sub-trees rooted at the children in X.
4. Extend p with zeros on all the other arcs not in $\mathbf{T}'$, and return $(\mathcal{J}, p)$.

Let $(\textsc{Opt}, p^*)$ be an optimal solution. We now argue that the solution returned by this algorithm has expected revenue of $p^*(\textsc{Opt})/8$. Clearly, for every $I \in \mathcal{I}$, either $p^*(I_1) \geq p^*(I)/2$ or $p^*(I_2) \geq p^*(I)/2$; let us call this more profitable part by I_*. Then $\sum_{I \in \textsc{Opt}} p^*(I_*) \geq p^*(\textsc{Opt})/2$. Let $\textsc{Opt}' = \{I_* : I \in \textsc{Opt}, I$ contains exactly one vertex of X and this vertex lies on $I_*\}$. Note that with probability exactly 1/4 each $I \in \textsc{Opt}$ has I_* belonging to $\textsc{Opt}'$. In particular,

$$\mathbb{E}[p^*(\textsc{Opt}')] = \sum_{I \in \textsc{Opt}} \mathbb{E}[p^*(I_*)] = \frac{1}{4} \sum_{I \in \textsc{Opt}} p^*(I_*) \geq \frac{1}{8} p^*(\textsc{Opt}).$$

Since what our procedure returns is at least as profitable as this quantity, the theorem follows. □

The randomized algorithm above can be derandomized using the method of *pairwise independence* [19,20,2].

4 Uncrossing Paths

Here we assume that the tree is rooted at some node $\mathbf{r} \in V$, and that paths in $\mathcal{I}$ have the following *uncrossing* property: If $I = [s,t] \in \mathcal{I}$ then t lies on the path $[s, \mathbf{r}]$. This property implies that once paths in $\mathcal{I}$ meet they cannot diverge.

In the course of the solution, we shall consider the following generalized version of the problem: Given intervals as above, and also a function $h : \mathcal{I} \times \mathbb{R}_+^n \mapsto \mathbb{R}_+$, find $\mathcal{J} \subseteq \mathcal{I}$ and a pricing $p : E \mapsto \mathbb{R}_+$, satisfying (1) and maximizing $\sum_{I \in \mathcal{J}} h(I, p)$.

Given a price function $p : E \mapsto \mathbb{R}_+$ and a node $w \in V$, the *accumulative price* at any node u *on the path* $[w, \mathbf{r}]$ with respect to w is defined as $p([w,u])$. Obviously, this monotonically increases as u moves towards the root. In this section we prove the following theorem.

Theorem 4. *There is a quasi-polynomial time approximation scheme for the tollbooth problem with uncrossing paths.*

In the following, we fix $K = \lceil \log(nP)/\log(1+\epsilon) \rceil$, where $P = \frac{nm}{\epsilon}$ (c.f. Proposition 1).

Definition 1. (ϵ-Relative pricings) *Let $w \in V$ be a given node of* $\mathbf{T}$*, and* $0 \leq k \leq K$ *and* $0 \leq k' \leq 2\log_{3/2} n$ *be given integers. We call any selection of k nodes $u_1, \ldots, u_k \in V$, k indices $-\infty \leq i_1 < \cdots < i_k \leq K$, and k' values $p_1, \ldots, p_{k'} \in \{0, 1, \ldots, nP\}$, such that $w, u_1, u_2, \ldots, u_k, \mathbf{r}$ lie on the path $[w, \mathbf{r}]$ in that order, an* ϵ-relative pricing *w.r.t. w, and denote it by $(w, k, k', u_1, \ldots, u_k, i_1, \ldots, i_k, p_1, \ldots, p_{k'})$.*

The total number of possible ϵ-relative pricings with respect to a given $w \in V$ is at most

$$L = (d(T)K)^K (nP + 1)^{2\log_{3/2} n}, \tag{3}$$

which is $m^{\mathrm{polylog}(m)}$ for every fixed $\epsilon > 0$.

Definition 2. (Consistent pricings) *Let $R = (w, k, k', u_1, \ldots, u_k, i_1, \ldots, i_k, p_1, \ldots, p_{k'})$ be an ϵ-relative pricing w.r.t. node $w \in V$, $\mathcal{L} = \{s_1, \ldots, s_{k'}\}$ be the set of separators from $\mathcal{N}(\mathbf{T})$ on the path from $(w, \mathbf{r}]$, and $p : E \mapsto \mathbb{R}_+$ be a pricing of E. We say that R is* ϵ-consistent *with p and $\mathcal{L}$ if*

(C1) for $j = 1, \ldots, k-1$, $(1+\epsilon)^{i_j} \leq p([w, u]) \leq (1+\epsilon)^{i_j+1}$ if u lies in the interval $[u_j, u_{j+1})$ (excluding u_{j+1}),
(C2) for $j = 1, \ldots, k'$, $p([w, s_j]) = p_j$.

Lemma 1. *Let $\tilde{p} : E \mapsto \mathbb{R}_+$ be an ϵ-optimal pricing for a given instance of* Tb*, $w \in V$ be an arbitrary node, and $\mathcal{L} = \{s_1, \ldots, s_{k'}\}$ be the set of separators in $\mathcal{N}(\mathbf{T})$ on the path from $[w, \mathbf{r}]$. Then there exists an ϵ-relative pricing R w.r.t. w, that is ϵ-consistent with $\tilde{p}$ and $\mathcal{L}$.*

With every ϵ-relative pricing R, we can associate a system of linear inequalities, denoted by $S(R)$, on a set of E variables $\{p(e) : e \in E\}$, consisting of the constraints $(C1)$ and $(C2)$, together with the non-negativity constraints $p(e) \geq 0$. The feasible set for this system gives the set of all possible pricings with which R is ϵ-consistent. For two systems of inequalities S_1, S_2, we denote by $S_1 \wedge S_2$ the system obtained by combining their inequalities.

Let $R = (w, k, k', u_1, \ldots, u_k, i_1, \ldots, i_k, p_1, \ldots, p_{k'})$ be an ϵ-relative pricing w.r.t. a node $w \in V$. Given an interval $I \in \mathcal{I}[w]$, we associate a value $v(I, R)$ to I, defined with respect to R as follows: Let $j(I)$ be the largest index such that $u_{i_{j(I)}}$ is contained in I. Then, define $v(I, R) = (1+\epsilon)^{j(I)}$. For a subset of intervals $\mathcal{I}' \subseteq \mathcal{I}$, we define, as usual, $v(\mathcal{I}', R) = \sum_{I \in \mathcal{I}'} v(I, R)$. It follows that for any ϵ-relative pricing R w.r.t. a node $w \in V$, any $p : E \mapsto \mathbb{R}_+$ with which R is consistent, and any $I = [s, t] \in \mathcal{I}[w]$, we have

$$v(I, R) \leq p([w, t]) \leq (1+\epsilon) v(I, R). \tag{4}$$

Decomposition into Two Subproblems. Let $w \in \mathcal{N}(\mathbf{T})$ be a separator node. Then $\mathbf{T}$ can be decomposed into two subtrees $\mathbf{T}_L = (V_L, E_L)$ and $\mathbf{T}_R = (V_R, E_R)$, such that the root $\mathbf{r} \in V_R$ and $w \in V_L \cap V_R$ is the root of $\mathbf{T}_L$. We define two Tb instances $(\mathbf{T}_L, \mathcal{I}_L)$ and $(\mathbf{T}_R, \mathcal{I}_R)$ where:

$$\begin{aligned}
\mathcal{I}_0 &= \{[s,t] \in \mathcal{I}[w] \ : \ s \in V_L \text{ and } t \in V_R\}, \\
\mathcal{I}_L &= \{[s,t] \in \mathcal{I} \ : \ s, t \in V_L\} \cup \{[s,w] \ : \ [s,t] \in \mathcal{I}_0\}, \\
\mathcal{I}_R &= \{[s,t] \in \mathcal{I} \ : \ s, t \in V_R\}.
\end{aligned}$$

In other words, the intervals passing through w, crossing from $\mathbf{T}_L$ to $\mathbf{T}_R$ are truncated in $\mathbf{T}_L$ while all other intervals remain the same[1]. Note that from the choice of w, we have $\max\{|V(\mathbf{T}_L)|, |V(\mathbf{T}_R)|\} \leq \frac{2n}{3} + 1$, and both instances $(\mathbf{T}_L, \mathcal{I}_L)$ and $(\mathbf{T}_R, \mathcal{I}_R)$ are of the uncrossing type, with roots w and $\mathbf{r}$, respectively.

The algorithm is given as Algorithm 1 below. It is initially called with an empty $\mathcal{S}$, and with $h(I) = 0$ for all $I \in \mathcal{I}$. The procedure iterates over all ϵ-relative pricings R, consistent with $\mathcal{S}$, w.r.t. the middle edge e^*, then recurses on the subsets of intervals to the left and right of e^*. Intervals crossing from $\mathbf{T}_L$ to $\mathbf{T}_R$ will be truncated and their values will be charged to $\mathbf{T}_L$; hence the corresponding budgets are reduced, and the corresponding h-values are increased.

Algorithm 1. $\textsc{Tb}(\mathbf{T}, \mathcal{I}, \mathbf{r}, B, h, \mathcal{S})$

Require: An uncrossing $\textsc{Tb}$ instance $(\mathbf{T} = (V, E), \mathcal{I})$ with root $\mathbf{r}$, budgets and values $B : \mathcal{I} \rightarrow \mathbb{R}_+$ and $h : \mathcal{I} \times \mathbb{R}^n_+ \rightarrow \mathbb{R}_+$, and a feasible system of inequalities $\mathcal{S}$
Ensure: A pricing $p : E \rightarrow \mathbb{R}_+$ and a subset $\mathcal{J} \subseteq \mathcal{I}$
1: **if** $|\mathcal{I}| = 0$ **then**
2: $\quad\mathcal{S}' \leftarrow$REDUCE$(\mathcal{S}, E)$
3: $\quad$return $(p, \emptyset)$, where p is any feasible solution of $\mathcal{S}'$
4: **end if**
5: **if** $d(\mathbf{T}) = 1$ **then**
6: $\quad$**for** edge e of $\mathbf{T}$ **do**
7: $\quad\quad\mathcal{S}' \leftarrow$REDUCE$(\mathcal{S}, \{e\})$
8: $\quad\quad p(e) \leftarrow \text{argmax}\{\sum_{I \in \mathcal{I}:\ p' \leq B(I)} (h(I) + p') \ :\ p' \text{ satisfies } \mathcal{S}'\}$
9: $\quad\quad\mathcal{J}(e) \leftarrow \{I \in \mathcal{I} : B(I) \geq p(e)\}$
10: $\quad$**end for**
11: $\quad$Return $((p(e) : e \in E), \bigcup_{e \in E} \mathcal{J}(e))$
12: **end if**
13: Let w be a separator node of $\mathbf{T}$ and $\mathbf{T}_L, \mathbf{T}_R, \mathcal{I}_0, \mathcal{I}_L, \mathcal{I}_R$ be as defined above
14: **for** every ϵ-relative pricing R w.r.t. w for which $S \wedge S(R)$ is feasible **do**
15: $\quad$**for** $I \in \mathcal{I}_0$ **do**
16: $\quad\quad B(I) \leftarrow B(I) - (1 + \epsilon)v(I, R)$
17: $\quad\quad h(I) \leftarrow h(I) + v(I, R)$
18: $\quad$**end for**
19: $\quad(p_1, \mathcal{J}_1) \leftarrow \textsc{Tb}(\mathbf{T}_L, \mathcal{I}_L, w, B, h, \mathcal{S})$
20: $\quad(p_2, \mathcal{J}_2) \leftarrow \textsc{Tb}(\mathbf{T}_R, \mathcal{I}_R, \mathbf{r}, B, h, \mathcal{S} \wedge S(R))$
21: $\quad$Let p be the pricing defined by $p(e) = p_1(e)$ if $e \in E_L$ and $p(e) = p_1(e)$ if $e \in E_R$
22: $\quad\mathcal{J} \leftarrow \mathcal{J}_1 \cup \mathcal{J}_2$
23: $\quad$record $(p, \mathcal{J})$
24: **end for**
25: Return the recorded solution with largest $p(\mathcal{J}) + h(\mathcal{J})$ value

Solving the Base Case. At the lowest level of recursion (either line 3 or 8), we have to solve a linear program defined by the system $\mathcal{S}$. Note that the system may

[1] Throughout, we will make the implicit assumption that each interval has an "identity"; so, for instance, $\mathcal{I}_L \cap \mathcal{I}_0$ will be used to denote the set $\{I \in \mathcal{I}_0 :\ I = [s, t]$ and $[s, w] \in \mathcal{I}_L\}$.

contain constraints on variables outside the current set of edges E of the current tree $\mathbf{T}$ (resulting from previous nodes of the recursion tree). However, we can reduce this LP to one that involves only variables in E. Indeed, any constraint that involves a variable not in E, has the form $L \leq p([w,u]) \leq U$, where $u \in V(\mathbf{T})$, and $w \notin V(\mathbf{T})$ is a separator node such that there is another separator node $w' \in V(\mathbf{T})$ on the path from w to u. Then when w' was considered in the recursion, a constraint of the form $p([w,w'])] = q$, for some value q, was appended to $\mathcal{S}$ (recall $(C2)$ in the definition of consistent pricings). Now, we can replace the first constraint by the equivalent constraint $L - q \leq p([w',u]) \leq U - q$, which only involves variables from E. This is exactly what procedure REDUCE($\mathcal{S}, \cdot$) does in lines 2 and 7.

When the procedure returns, we get a pricing $p : E \mapsto \mathbb{R}_+$ and a set of intervals $\mathcal{J} \subseteq \mathcal{I}$ which can be purchased under this pricing.

Theorem 4 follows from the following two lemmas.

Lemma 2. *Algorithm* TB *runs in quasi-polynomial time in* m*, for any fixed* $\epsilon > 0$.

Lemma 3. *For any* $\epsilon > 0$*, Algorithm* TB *returns a pricing* p *and a set of intervals* $\mathcal{J}$ *such that* $p(I) \leq B(I)$ *for all* $I \in \mathcal{J}$ *and* $p(\mathcal{J}) \geq (1 - 3\epsilon)p^*(\text{OPT})$.

5 Hardness of the Highway Problem

5.1 Strong NP-Hardness in the Standard Model

Recall that the highway problem is the special case of the tollbooth problem when the underlying graph is a path. In this section, we show that the problem is strongly NP-hard, thus ruling out the existence of an FPTAS for the problem. Consider a MAX-2-SAT instance with n variables $\{x_1, \ldots, x_n\}$ and m clauses $\{C_1, \ldots, C_m\}$. Let the variables be numbered $1, \ldots, n$.

Theorem 5. *The highway problem is strongly NP-hard.*

Proof. The proof follows by the construction of gadgets for the variables and clauses in a given MAX-2-SAT instance. We next describe their construction.
Variable Gadget: The gadget for each variable consists of two copies of the following *basic gadget*, and a *consistency gadget*.
Basic Gadget: The basic gadget consists of 4 edges $e_1, \ldots, e_4$, and 4 types of intervals A, B, C and D. There are 4 intervals each of type A and B, labeled $a_1, \ldots, a_4$, and $b_1, \ldots, b_4$ respectively. The intervals $a_i = b_i = [e_i]$, for $i = 1, \ldots, 4$. The intervals $a_1, \ldots, a_4$ have budgets of $1, 2, 2, 1$ respectively, and the intervals $b_1, \ldots, b_4$ have budgets $2, 1, 1, 2$ respectively. There are 2 type C intervals, c_1 and c_2, with $c_1 = [e_1, e_2]$, and $c_2 = [e_3, e_4]$. These intervals have a budget of 3. There are two intervals of type D, $d_1 = d_2 = [e_2, e_3]$ with d_1 having a budget of 4, and d_2, a budget of 2. The basic gadget is shown in Figure 1. We can show that there are exactly two price assignments for $\{e_1, \ldots, e_4\}$ that gives us optimum profit.

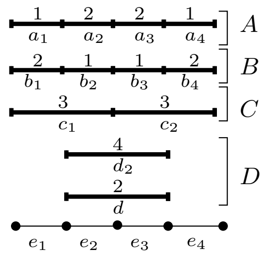

Fig. 1. A basic gadget. The gadget consists of 4 edges, and 4 types of intervals A, B, C and D. The interval labels are shown below each interval, and the budgets are shown above each interval.

Lemma 4. *The maximum profit that can be obtained from a basic gadget is* 18, *and there are exactly two sets of prices namely* $(1, 2, 2, 1)$ *and* $(2, 1, 1, 2)$ *for the edges* $(e_1, \ldots, e_4)$ *that achieve this profit.*

The price assignment $(1, 2, 2, 1)$ and $(2, 1, 1, 2)$ to the edges $e_1, \ldots, e_4$ respectively are called TRUE and FALSE assignments respectively. The variable gadget is constructed on $8n + 1$ edges $(e_{4n}, e_{4n-1}, \ldots, e_1, h, f_1, \ldots, f_{4n})$, where n is the number of variables in the MAX-2-SAT instance. Each variable gadget consists of two copies of the basic gadget, along with a consistency gadget. The consistency gadget ensures that the two basic gadgets have the same price assignment. More formally, let $(x_1, \ldots, x_n)$ be an order on the variables of the MAX-2-SAT instance. Then, the gadget for variable x_i consists of two basic gadgets, B_i^1 and B_i^2. B_i^1 consists of intervals (customers) interested in the edges $e_{4i-3}, \ldots, e_{4i}$ and B_i^2 consists of intervals interested in the edges $f_{4i-3}, \ldots, f_{4i}$. Finally, the intervals ensuring consistency of the gadget for variable x_i spans from $e_{4i-1}, \ldots, f_{4i-3}$. The consistency gadget consists of a single interval that has a budget of $mn^2 + 6(2i-2) + 6$. Finally, we add a new type of interval, called a type H interval that is interested only in the edge h, and has a budget of mn^2.

Figure 2 shows the arrangement of the variable gadgets. We can show that the consistency intervals do their job, i.e., if for a variable gadget, B_i^1 and B_i^2 have different price assignments, we obtain a smaller profit than when they are the same.

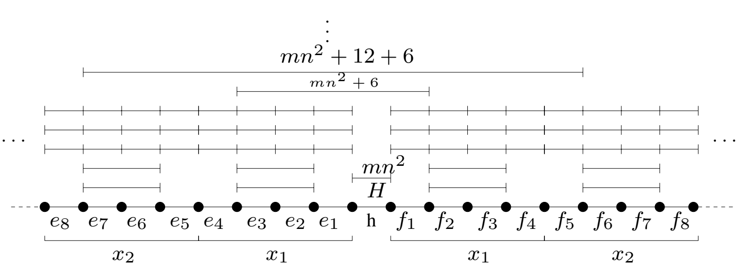

Fig. 2. The variable gadget

Lemma 5. *The maximum profit of* $2mn^2 + 6(2i-2) + 6 + 36$ *from a variable gadget and the interval* h *is achieved only when both the basic gadgets corresponding to a variable are consistent, and the type* H *interval purchases edge* h *at a price of* mn^2*.*

We will create several copies of the basic gadgets, the consistency gadgets for each variable as well as several copies of the H interval to ensure that in an optimum price assignment, the basic gadgets are consistent, and the reduction goes through. But before we do this, we describe the clause gadgets.

Clause Gadgets: The clause gadget for a clause of variables x_i and x_j runs between the basic gadget B_i^1 and B_j^2. There are four types of clause gadgets corresponding to the four types of clauses. Each clause gadget consists of one interval. These intervals have the property that we obtain a certain revenue from the clause interval if and only if the clause is satisfied; otherwise we obtain nothing. (See the table in Figure 3).

Clause	Interval	Budget
$(x_i \vee x_j)$	$[e_{4i-3}, f_{4j-3}]$	$mn^2 + 6(i+j-2) + 3$
$(\overline{x_i} \vee x_j)$	$[e_{4i-1}, f_{4j-3}]$	$mn^2 + 6(i+j-2) + 6$
$(x_i \vee \overline{x_j})$	$[e_{4i-3}, f_{4j-1}]$	$mn^2 + 6(i+j-2) + 6$
$(\overline{x_i} \vee \overline{x_j})$	$[e_{4i-1}, f_{4j-1}]$	$mn^2 + 6(i+j-2) + 9$

Fig. 3. This table shows the lengths and budgets of the intervals making up a clause gadget for the four different kinds of clauses

We say that a pricing is *consistent* if for every variable, the price assignment to the two basic gadgets of the variable gadget are both TRUE or both FALSE, and the consistency intervals spend their entire budgets.

Lemma 6. *Consider a clause* C *consisting of variables* x_i *and* x_j *and a consistent price assignment to the edges. Then, the intervals corresponding to* C *will be able to purchase their desired edges if and only if the corresponding truth assignment to the variables satisfies the clause* C*.*

We now describe the final reduction. As mentioned earlier, we have to create copies of the variable gadget, consistency gadget and the H interval for the proof to go through. We make T copies of each basic gadget, of each consistency gadget, and of the H interval, where any value of T, larger than m^2n^3 will suffice for the proof. Observe that for a variable gadget again, the profit maximizing prices achieve consistency of the variable gadget, and making T copies of the H intervals ensures that the price of the edge h is set to mn^2.

5.2 APX-Hardness in the Discount Model

When negative prices are allowed, the highway problem becomes APX-hard.

Theorem 6. *The highway problem with negative prices is APX-hard even when restricted to instances in which one edge is shared by all the customers.*

We prove the above theorem by first showing that it is equivalent to a pricing problem on bipartite graphs and then prove that the latter is APX-hard via a reduction from maxcut on 3-regular graphs. The details of the proof can be found in the extended version of this paper. This result has been independently obtained in [18].

6 Conclusion

We presented an $O(\log n)$-approximation algorithm for the tollbooth problem on trees, which is better than the current upper bound for the general problem. Improving this bound is an interesting open problem. One plausible direction towards this is to use as a subroutine, the quasi-polynomial time algorithm for the case of uncrossing paths. Such techniques have been used before, for example for the multicut problem on trees [14]. However, it is unclear how a general instance of the Tb problem can be decomposed into a set of problems of the uncrossing type. For the highway problem, the strong NP-hardness presented in this paper shows that the problem is almost closed, modulo improving the running time from quasi-polynomial to polynomial. When negative prices are allowed, somewhat surprisingly, the problem becomes harder, and even a quasi-PTAS is unlikely to exist.

Acknowledgements. We would like to thank Naveen Garg for suggesting to use the separator theorem in the proof of Theorem 1.

References

1. Aggarwal, G., Hartline, J.D.: Knapsack auctions. In: SODA 2006: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp. 1083–1092. ACM Press, New York (2006)
2. Balcan, M.F., Blum, A.: Approximation algorithms and online mechanisms for item pricing. In: EC 2006: Proceedings of the 7th ACM conference on Electronic commerce, pp. 29–35. ACM Press, New York (2006)
3. Balcan, M.-F., Blum, A., Chan, H., Hajiaghayi, M.: A theory of loss-leaders: Making money by pricing below cost. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 293–299. Springer, Heidelberg (2007)
4. Balcan, M.F., Blum, A., Mansour, Y.: Item pricing for revenue maximization. In: EC 2008: Proceedings of the 9th ACM conference on Electronic commerce. ACM Press, New York (to appear) (2008)
5. Balcan, M.F., Blum, A.: Approximation algorithms and online mechanisms for item pricing. Theory of Computing 3, 179–195 (2007)
6. Briest, P.: Uniform budgets and the envy-free pricing problem. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 808–819. Springer, Heidelberg (2008)

7. Briest, P., Hoefer, M., Krysta, P.: Stackelberg network pricing games. In: Proc. 25th International Symposium on Theoretical Aspects of Computer Science (STACS), pp. 133–142 (2008)
8. Briest, P., Krysta, P.: Single-minded unlimited supply pricing on sparse instances. In: SODA 2006: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp. 1093–1102. ACM Press, New York (2006)
9. Briest, P., Krysta, P.: Buying cheap is expensive: Hardness of non-parametric multi-product pricing. In: Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM-SIAM (2007)
10. Cheung, M., Swamy, C.: Approximation algorithms for single-minded envy-free profit-maximization problems with limited supply. In: FOCS 2008, pp. 35–44 (2008)
11. Demaine, E.D., Hajiaghayi, M.T., Feige, U., Salavatipour, M.R.: Combination can be hard: approximability of the unique coverage problem. In: SODA 2006: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp. 162–171. ACM Press, New York (2006)
12. Elbassioni, K.M., Sitters, R.A., Zhang, Y.: A quasi-PTAS for profit-maximizing pricing on line graphs. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 451–462. Springer, Heidelberg (2007)
13. Glynn, P.W., Van Roy, B., Rusmevichientong, P.: A nonparametric approach to multi-product pricing. Operations Research 54(1), 82–98 (2006)
14. Golovin, D., Nagarajan, V., Singh, M.: Approximating the k-multicut problem. In: SODA 2006, pp. 621–630 (2006)
15. Grigoriev, A., van Loon, J., Sitters, R., Uetz, M.: How to sell a graph: Guidelines for graph retailers. In: Fomin, F.V. (ed.) WG 2006. LNCS, vol. 4271, pp. 125–136. Springer, Heidelberg (2006)
16. Guruswami, V., Hartline, J.D., Karlin, A.R., Kempe, D., Kenyon, C., McSherry, F.: On profit-maximizing envy-free pricing. In: SODA 2005: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA, pp. 1164–1173. Society for Industrial and Applied Mathematics (2005)
17. Hartline, J.D., Koltun, V.: Near-optimal pricing in near-linear time. In: Dehne, F., López-Ortiz, A., Sack, J.-R. (eds.) WADS 2005. LNCS, vol. 3608, pp. 422–431. Springer, Heidelberg (2005)
18. Khandekar, R., Kimbrel, T., Makarychev, K., Sviridenko, M.: On hardness of pricing items for single-minded bidders. In: Proceedings, 12th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX (2009)
19. Luby, M., Wigderson, A.: Pairwise independence and derandomization. Foundations and Trends in Theoretical Computer Science 1(4), 237–301 (2005)
20. Motwani, R., Raghavan, P.: Randomized algorithms. Cambridge University Press, Cambridge (1995)