

On Profit-Maximizing Pricing for the Highway and Tollbooth Problems

Khaled Elbassioni¹, Rajiv Raman¹, Saurabh Ray², and René Sitters³

¹ Max-Planck-Institut für Informatik, Saarbrücken, Germany;
{elbassio,rraman}@mpi-inf.mpg.de

² Universität des Saarlandes, Saarbrücken, Germany; saurabh@cs.uni-sb.de

³ Department of Mathematics and Computer Science, VU, Amsterdam, the Netherlands;
rsitters@feweb.vu.nl

Abstract. In the *tollbooth problem*, 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 tree. 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 *coupon 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 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, 5–10, 13, 11, 14, 15, 8].

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. [14], under the name of *tollbooth problem*. The authors of [14] 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 [14], 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 [14], and not necessarily paths of a network, and even in the non single-minded case [4]. Very recently, and more generally, Cheung and Swamy [8] 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 [14], 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 [10] 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 [6]. 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. Most proofs have been omitted due to lack of space from this extended abstract, and can be found in the appendix.

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 $\mathcal{I} = \{I_1, \dots, I_m\}$, 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

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

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 $\text{OPT} \subseteq \mathcal{I}$ 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 is 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 of the edges $\{v, v'\}$ for each child v' of v .

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

Proposition 1 ([10]). *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) $\tilde{p}(e) \in \{0, 1, \dots, P\}$, for every $e \in E$, where $P = nm/\epsilon$,
- (ii) $\tilde{p}(I) \leq \frac{B(I)}{1+\epsilon}$, for every $I \in \text{OPT}$, 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, \dots, s_r be the sizes of the components obtained by deleting v from \mathbf{T} , then there is a subset $S \subseteq [r] \stackrel{\text{def}}{=} \{1, \dots, r\}$ such that*

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

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 T_1, \dots, T_r be the components of $T - v_0$. Recursively, find separator vertices v_1, \dots, v_r in $\mathbf{T}_1, \dots, \mathbf{T}_r$. We say that node v_0 has $\text{level}(v_0) = 1$, nodes v_1, \dots, 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 $\text{level}(v) = \text{level}(v') + 1$. 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 ([14]). *The tollbooth problem on rooted paths can be solved in polynomial time using dynamic programming.*

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

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

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 $\mathcal{I}(i)$, $i = 1, \dots, k$, 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, \dots, 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 $X \subseteq \{v_1, \dots, v_r\}$ be a subset obtained by picking each v_i randomly and independently with probability $1/2$.
2. Let $\mathcal{I}' = \{I_j \in \mathcal{I} : j \in \{1, 2\}, I_j \text{ contains exactly one vertex of } X\}$.
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 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 T' , and return (\mathcal{J}, p) .

Let (OPT, p^*) be an optimal solution. We now argue that the solution returned by this algorithm has expected revenue of $p^*(\text{OPT})/8$. 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 $\sum_{I \in \text{OPT}} p^*(I_*) \geq p^*(\text{OPT})/2$. Let $\text{OPT}' = \{I \in \text{OPT} : I_* \text{ contains exactly one vertex of } X\}$. Note that with probability at least $1/4$ each $I \in \text{OPT}$ has I_* intersecting the random set X in exactly one vertex. In particular,

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

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

The randomized algorithm above can be derandomized using the method of *pairwise independence* [16, 17, 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 $u_1, \dots, u_k \in V$, k indices $- \infty \leq i_1 < \dots < i_k \leq K$, and k' values $p_1, \dots, p_{k'} \in \{0, 1, \dots, nP\}$, such that $w, u_1, u_2, \dots, u_k, \mathbf{r}$ lie on the path $[w, \mathbf{r}]$ in that order, an ϵ -relative pricing w.r.t. w , and denote it by $(w, k, k', u_1, \dots, u_k, i_1, \dots, i_k, p_1, \dots, p_{k'})$.*

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

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

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

Definition 2. (Consistent pricings) *Let $R = (w, k, k', u_1, \dots, u_k, i_1, \dots, i_k, p_1, \dots, p_{k'})$ be an ϵ -relative pricing w.r.t. node $w \in V$, $\mathcal{L} = \{s_1, \dots, s_{k'}\}$ 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 $j = 1, \dots, k-1$, $(1 + \epsilon)^{i_j} \leq p([w, u]) \leq (1 + \epsilon)^{i_{j+1}}$ if u lies in the interval $[u_j, u_{j+1})$ (excluding u_{j+1}),
- (C2) for $j = 1, \dots, k'$, $p([w, s_j]) = p_j$.

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 $\mathcal{L} = \{s_1, \dots, s_{k'}\}$ 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 $R = (w, k, k', u_1, \dots, u_k, i_1, \dots, i_k, i'_1, \dots, i'_{k'})$ 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

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

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 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 same⁴. Note that from the choice of w , we have $\max\{|\mathcal{I}_L|, |\mathcal{I}_R|\} \leq \frac{2n}{3} + 1$, 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.

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 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.

⁴ 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 $\{I \in \mathcal{I}_0 : I = [s, t] \text{ and } [s, w] \in \mathcal{I}_L\}$

Algorithm 1 $\text{TB}(\mathbf{T}, \mathcal{I}, \mathbf{r}, B, h, \mathcal{S})$

Require: An uncrossing 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: $\mathcal{S}' \leftarrow \text{REDUCE}(\mathcal{S}, E)$
- 3: return (p, \emptyset) , where p is any feasible solution of \mathcal{S}'
- 4: **end if**
- 5: **if** $d(\mathbf{T}) = 1$ **then**
- 6: **for** edge e of \mathbf{T} **do**
- 7: $\mathcal{S}' \leftarrow \text{REDUCE}(\mathcal{S}, \{e\})$
- 8: $p(e) \leftarrow \text{argmax}\{\sum_{I \in \mathcal{I}: p' \leq B(I)} (h(I) + p') : p' \text{ satisfies } \mathcal{S}'\}$
- 9: $\mathcal{J}(e) \leftarrow \{I \in \mathcal{I} : B(I) \geq p(e)\}$
- 10: **end for**
- 11: 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: **for** $I \in \mathcal{I}_0$ **do**
- 16: $B(I) \leftarrow B(I) - (1 + \epsilon)v(I, R)$
- 17: $h(I) \leftarrow h(I) + v(I, R)$
- 18: **end for**
- 19: $(p_1, \mathcal{J}_1) \leftarrow \text{TB}(\mathbf{T}_L, \mathcal{I}_L, w, B, h, \mathcal{S})$
- 20: $(p_2, \mathcal{J}_2) \leftarrow \text{TB}(\mathbf{T}_R, \mathcal{I}_R, \mathbf{r}, B, h, \mathcal{S} \wedge S(R))$
- 21: Let p be the pricing defined by $p(e) = p_1(e)$ if $e \in E_L$ and $p(e) = p_2(e)$ if $e \in E_R$
- 22: $\mathcal{J} \leftarrow \mathcal{J}_1 \cup \mathcal{J}_2$
- 23: record (p, \mathcal{J})
- 24: **end for**
- 25: Return the recorded solution with largest $p(\mathcal{J}) + h(\mathcal{J})$ value

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 $\{x_1, \dots, x_n\}$ and m clauses $\{C_1, \dots, C_m\}$. Let the variables be numbered $1, \dots, n$.

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, \dots, 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, \dots, a_4 , and b_1, \dots, b_4 respectively. The intervals $a_i = b_i = [e_i]$, $i = 1, \dots, 4$. The intervals a_1, \dots, a_4 have budgets of 1, 2, 2, 1 respectively, and the intervals b_1, \dots, 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 now show that there are exactly two price assignments for $\{e_1, \dots, e_4\}$ that gives us optimum profit.

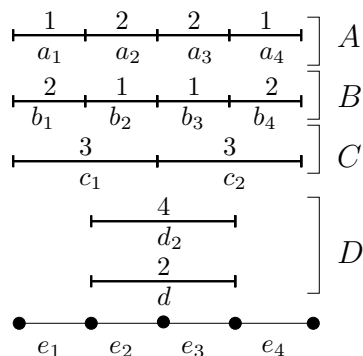


Fig. 1. A basic gadget. The gadget consists of 4 edges, and 4 types of intervals A, B, C and D . The interval labels are shown below each interval, and the budgets are shown above each interval.

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, \dots, e_4) that achieve this profit.*

The price assignment $(1, 2, 2, 1)$ and $(2, 1, 1, 2)$ to the edges e_1, \dots, e_4 respectively are called TRUE and FALSE assignments respectively. The variable gadget is constructed on $8n + 1$ edges $(e_{4n}, e_{4n-1}, \dots, e_1, h, f_1, \dots, f_{4n})$, 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, \dots, 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}, \dots, e_{4i} and B_i^2 consists of intervals interested in the edges f_{4i-3}, \dots, f_{4i} . Finally, the intervals ensuring consistency of the gadget for variable x_i spans from $e_{4i-1}, \dots, f_{4i-3}$. 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 now 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.

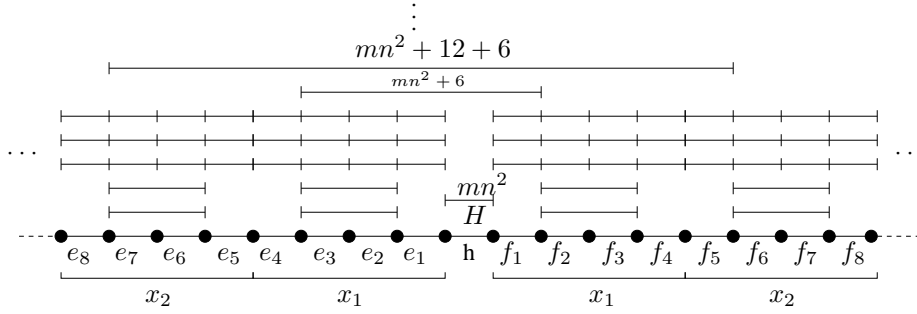


Fig. 2. The variable gadget.

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. The clause gadgets for the four types of clauses are shown in Table 3 and in Figures 5, 6, 7, and 8 in the Appendix.

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

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

Proof. We will show that the problem is equivalent to a pricing problem on bipartite graphs and prove that the latter problem is APX-hard. Assume we are given an instance of the highway problem in which edge e is contained in each of the intervals. We split edge e by adding a node v_0 on e . This has no effect on the problem. We construct a bipartite graph H with one set consisting of the points left of v_0 and the other set consisting of the points right of v_0 . Now, an interval containing e becomes an edge in the bipartite graph. The items are the vertices and a customer is interested in the two items on the vertices of its edge. Given a pricing p for the highway instance, we define the pricing q_p of H by letting the price of a vertex v in H be the cumulative price $p([v_0, v])$. Conversely, for any pricing q of the vertices of H there is a corresponding pricing p of the highway problem such that $q = q_p$.

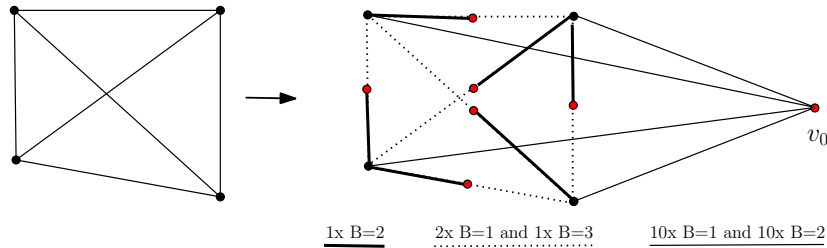


Fig. 4. Reduction: Each edges e is partitioned in edges e_1 (fat) and e_2 (dotted). Which is which is arbitrary. The notation 1x B=2 means that there is one customer with budget 2 on this type of edge.

We prove that the pricing problem on bipartite graphs is APX-hard by a reduction from maxcut on 3-regular graphs. Given a 3-regular graph $G = (V, E)$, we make it bipartite by placing an extra vertex v_e on every edge e , dividing it in two new edges e_1 and e_2 . For e_1 there is one customer with budget 2 and for e_2 there are two customers with budget 1 and one customer with budget 3. We define one extra vertex v_0 and define for each $v \in V$ an edge $a_v = (v, v_0)$. For each such edge a_v there are ten customers with budget 1 and ten customers with budget 2. The bipartite graph partitions into V and the new vertices $V' = \{v_e \mid e \in E\} \cup \{v_0\}$. (To enhance reading we write prices and budgets in decimal and amounts of customers in words.)

Consider any pricing p of the bipartite graph. We may assume that $p(v_0) = 0$ since subtracting $p(v_0)$ from all vertices in V' and adding $p(v_0)$ to all vertices in V does not change the profit. We will have to take into account though that prices may be negative.

Next, we argue that in any optimal solution the price $p(v)$ of any vertex $v \in V$ is either 1 or 2. Denote by $p(e)$ the profit we get from the customer on e_1 plus the three customers on e_2 . It is easy to see that $3 \leq p(e) \leq 5$ for any edge e . Suppose $p(v) = \alpha \notin \{1, 2\}$. If $\alpha > 2$ or $\alpha \leq 0$ then the profit on a_v is 0. By changing the price to $p(v) = 2$ the profit becomes ten times 2 is 20. The maximum profit on any of the three adjacent edges e is 5. Hence, we gain 20 and loose at most 15. Now assume $0 < \alpha < 1$. We raise the price of v to price $p(v) = 1$ and reduce the price on the vertices v_e by $1 - \alpha$ for each of the three adjacent edges e of v in G . The 20 customers on a_v add an extra $20(1 - \alpha)$ to the profit. No other customer sees an increase of its bundle price and at most 9 customers will see a reduction of the bundle price. Hence, we loose at most $9(1 - \alpha)$ on them. Now assume $1 < \alpha < 2$. We raise the price to $p(v) = 2$ and reduce the price on the vertices v_e of adjacent edges e by $2 - \alpha$. The argument is the same: We gain $10(2 - \alpha)$ and loose at most $9(2 - \alpha)$ since at most 9 customers will see their bundle price drop.

We showed that there is an optimal pricing in which $p(v_0) = 0$ and $p(v) \in \{1, 2\}$ for all $v \in V$. Next we prove that there is a cut of size k in G if and only if the maximum profit is $20|V| + 4|E| + k$. Given a cut of size k we price the vertices on one side 1 and on the other side 2. Now consider an edge $e = (v, w) \in E$ with $p(v) = p(w) = 1$. We can get a profit $p(e) = 4$ by setting $p(v_e) = 0$ or $p(v_e) = 1$. This is also the maximum profit possible. Similarly, if $p(v) = p(w) = 2$ then we can get the maximum profit $p(e) = 4$ by setting $p(v_e) = 0$. Finally, if $p(v) = 1$ and $p(w) = 2$ then we can get the maximum possible profit $p(e) = 5$ by setting $p(v_e) = 0$ or $p(v_e) = 1$, depending on how we chose e_1 and e_2 . From the customers on edges adjacent to v_0 we get the maximum profit of 20 per edge. The total profit is exactly $20|V| + 5k + 4(|E| - k) = 20n + 4|E| + k$ and this is maximum possible if the maximum cut is k . The reduction is gap-preserving since $3|V| = 2|E|$ and $|E| \leq 2k$. \square

In the reduction we showed that there always is an optimal pricing of the bipartite graph with only non-negative prices. Hence, the bipartite graph pricing problem remains APX-hard if we restrict to non-negative prices.

Corollary 1. *The graph pricing problem is APX-hard on bipartite graphs and all budgets in $\{1, 2, 3\}$. This holds for the non-negative version as well as for the version with negative prices allowed.*

Guruswami et al. [14] show that the graph-pricing problem is APX-hard even if all budgets are 1. Note that the bipartite case is trivially solved in that case by setting a price of 1 to all items on one side.

6 Conclusion

In this paper, we presented an $O(\log n)$ -approximation algorithm for the tollbooth problem on trees, which is better than the upper bound currently known 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 [12]. 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 becomes 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. G. Aggarwal and J. D. Hartline, *Knapsack auctions*, SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm (New York, NY, USA), ACM Press, 2006, pp. 1083–1092.
2. M. F. Balcan and A. Blum, *Approximation algorithms and online mechanisms for item pricing*, EC '06: Proceedings of the 7th ACM conference on Electronic commerce (New York, NY, USA), ACM Press, 2006, pp. 29–35.
3. M.-F. Balcan, A. Blum, H. Chan, and M. Hajiaghayi, *A theory of loss-leaders: Making money by pricing below cost*, WINE, 2007, pp. 293–299.
4. M. F. Balcan, A. Blum, and Y. Mansour, *Item pricing for revenue maximization*, EC '08: Proceedings of the 9th ACM conference on Electronic commerce, to appear (New York, NY, USA), ACM Press, 2008.
5. M.F. Balcan and A. Blum, *Approximation algorithms and online mechanisms for item pricing*, Theory of Computing **3** (2007), 179–195.
6. P. Briest and P. Krysta, *Single-minded unlimited supply pricing on sparse instances*, SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm (New York, NY, USA), ACM Press, 2006, pp. 1093–1102.
7. ———, *Buying cheap is expensive: Hardness of non-parametric multi-product pricing*, Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM-SIAM, 2007.
8. M. Cheung and C. Swamy, *Approximation algorithms for single-minded envy-free profit-maximization problems with limited supply*, FOCS, 2008, pp. 35–44.
9. E. D. Demaine, M. T. Hajiaghayi, U. Feige, and M. R. Salavatipour, *Combination can be hard: approximability of the unique coverage problem*, SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm (New York, NY, USA), ACM Press, 2006, pp. 162–171.
10. K.M. Elbassioni, R.A. Sitters, and Y. Zhang, *A quasi-PTAS for profit-maximizing pricing on line graphs*, ESA (L. Arge, M. Hoffmann, and E. Welzl, eds.), Lecture Notes in Computer Science, vol. 4698, Springer, 2007, pp. 451–462.
11. P. W. Glynn, B. Van Roy, and P. Rusmevichientong, *A nonparametric approach to multi-product pricing*, Operations Research **54** (2006), no. 1, 82–98.
12. D. Golovin, V. Nagarajan, and M. Singh, *Approximating the k-multicut problem*, SODA, 2006, pp. 621–630.
13. A. Grigoriev, J. van Loon, R. Sitters, and M. Uetz, *How to sell a graph: Guidelines for graph retailers.*, WG, 2006, pp. 125–136.
14. V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry, *On profit-maximizing envy-free pricing*, SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (Philadelphia, PA, USA), Society for Industrial and Applied Mathematics, 2005, pp. 1164–1173.
15. 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.

16. M.Luby and A.Wigderson, *Pairwise independence and derandomization*, Foundations and Trends in Theoretical Computer Science **1** (2005), no. 4, 237–301.
17. R. Motwani and P. Raghavan, *Randomized algorithms*, Cambridge University Press, 1995.

Appendix A: Proofs

Proof of Lemma 1. Let $R = (w, k, k', u_1, \dots, u_k, i_1, \dots, i_k, p_1, \dots, p_{k'})$ be defined as follows: write $u_0 = w$ and $i_0 = -\infty$, and let i_j and u_j , for $j = 1, 2, \dots$, be respectively the smallest non-negative index and the closest node to u_{j-1} on the path $[u_{j-1}, \mathbf{r}]$ with $(1 + \epsilon)^{i_{j-1}} < \tilde{p}([w, u_j]) \leq (1 + \epsilon)^{i_j}$; k will be the largest such index j . Finally, for $j = 1, \dots, k'$, let $p_j = p([w, s_j])$. Note that $k \leq K$ since $(1 + \epsilon)^k \leq nP$, and $k' \leq 2 \log_{3/2} n$ since the number of separators on the path from any node to the root is at most $2 \log_{3/2} n$. \square

Proof of Lemma 2. The number of possible ϵ -relative pricing is at most L , given in (3). This gives the recurrence

$$T(n) \leq \text{poly}(n, m) + 2L \cdot T\left(\frac{2n}{3}\right).$$

for the running time. Thus $T(n) \leq L^{O(\log n)} \text{poly}(m)$ and the lemma follows. \square

Proof of Lemma 3. Let $(p_{\mathbf{T}}, \mathcal{J}_{\mathbf{T}})$ be the solution returned by the algorithm when the input is $(\mathbf{T}, \mathcal{I}_{\mathbf{T}}, \mathbf{r}_{\mathbf{T}}, B_{\mathbf{T}}, h_{\mathbf{T}}, \mathcal{S}_{\mathbf{T}})$. We show by induction on the depth of the recursion tree that, if there exists a pricing $p'_{\mathbf{T}}$ satisfying $\mathcal{S}_{\mathbf{T}}$, and a set $\mathcal{J}'_{\mathbf{T}}$ such that $p'_{\mathbf{T}}(I) \leq B_{\mathbf{T}}(I)/(1 + \epsilon)$ for all $I \in \mathcal{J}'_{\mathbf{T}}$, then

- (i) $p_{\mathbf{T}}(I) \leq B_{\mathbf{T}}(I)$ for all $I \in \mathcal{J}_{\mathbf{T}}$, and $p_{\mathbf{T}}$ is feasible for $\mathcal{S}_{\mathbf{T}}$; and
- (ii) $p_{\mathbf{T}}(\mathcal{J}_{\mathbf{T}}) + h_{\mathbf{T}}(\mathcal{J}_{\mathbf{T}}) \geq \frac{p'_{\mathbf{T}}(\mathcal{J}'_{\mathbf{T}})}{1 + \epsilon} + h_{\mathbf{T}}(\mathcal{J}'_{\mathbf{T}})$.

The statement of the theorem follows from (i) and (ii) by taking, at the highest level where $h_{\mathbf{T}}(I) = 0$ for all I , $p'_{\mathbf{T}} = \tilde{p}$ (an ϵ -optimal pricing) and $\mathcal{J}'_{\mathbf{T}} = \text{OPT}$.

Base case. At a leaf of the recursion tree, we either have $|\mathcal{I}| = 0$ in which case (i) and (ii) are trivially satisfied, or $d(\mathbf{T}) = 1$ in which case (i) and the stronger version of (ii), $p_{\mathbf{T}}(\mathcal{J}_{\mathbf{T}}) + h(\mathcal{J}_{\mathbf{T}}) \geq p'_{\mathbf{T}}(\mathcal{J}'_{\mathbf{T}}) + h_{\mathbf{T}}(\mathcal{J}'_{\mathbf{T}})$, are insured by the computation in line 7.

General recursion level. Let $w, \mathbf{T}_L, \mathbf{T}_R, \mathcal{I}_0, \mathcal{I}_L, \mathcal{I}_R$ be as defined in line 10 at the current level, and p_1, p_2, \mathcal{J}_1 and \mathcal{J}_2 the returned pricings and sets at lines 15 and 16. Let $R_{\mathbf{T}}$ be an ϵ -relative pricing consistent with $p'_{\mathbf{T}}$. Then the restrictions $p'_{\mathbf{T}_L}$ and $p'_{\mathbf{T}_R}$ of $p'_{\mathbf{T}}$ on \mathbf{T}_L and \mathbf{T}_R satisfy, respectively, $\mathcal{S}_{\mathbf{T}_L}$ and $\mathcal{S}_{\mathbf{T}_R}$. Moreover, for any $I \in \mathcal{J}'_{\mathbf{T}} \cap (\mathcal{I}_L \setminus \mathcal{I}_0)$, we have $p'_{\mathbf{T}_L}(I) \leq \frac{B_{\mathbf{T}_L}(I)}{1 + \epsilon}$; for any $I \in \mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_R$, we have $p'_{\mathbf{T}_R}(I) \leq \frac{B_{\mathbf{T}_R}(I)}{1 + \epsilon}$; and for any $I = [s, t] \in \mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_0$, we have $p'_{\mathbf{T}_L}(I) = p'_{\mathbf{T}}([s, w]) = p'_{\mathbf{T}}([s, t]) - p'_{\mathbf{T}}([w, t]) \leq p'_{\mathbf{T}}([s, t]) - v(I, R_{\mathbf{T}}) \leq \frac{B_{\mathbf{T}}(I)}{1 + \epsilon} - v(I, R_{\mathbf{T}}) = \frac{B_{\mathbf{T}_L}(I)}{1 + \epsilon}$, where the first inequality follow from (4), and the last equation follows from line 13 of the procedure. Thus we can apply the induction hypothesis to the two subproblems, and hence get that

$$p_1(I) \leq B_{\mathbf{T}_L}(I) \text{ for all } I \in \mathcal{J}_1, \quad (5)$$

$$p_2(I) \leq B_{\mathbf{T}_R}(I) \text{ for all } I \in \mathcal{J}_2, \quad (6)$$

$$p_1(\mathcal{J}_1) + h_{\mathbf{T}_L}(\mathcal{J}_1) \geq \frac{p'_{\mathbf{T}_L}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_L)}{1 + \epsilon} + h_{\mathbf{T}_L}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_L), \quad (7)$$

$$p_2(\mathcal{J}_2) + h_{\mathbf{T}_R}(\mathcal{J}_2) \geq \frac{p'_{\mathbf{T}_R}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_R)}{1 + \epsilon} + h_{\mathbf{T}_R}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_R), \quad (8)$$

and both p_1 and p_2 , and hence $p_{\mathbf{T}}$, satisfy $\mathcal{S}_{\mathbf{T}}$. By (5) and (6), we have $p_{\mathbf{T}}(I) \leq B_{\mathbf{T}}(I)$ for all $I \in \mathcal{J}_1 \cup \mathcal{J}_2 \setminus \mathcal{I}_0$. By (5) and line 13 of the procedure, we also have $p_1([s, w]) \leq B_{\mathbf{T}}(I) - (1 + \epsilon)v(I, R_{\mathbf{T}})$ for all $I = [s, t] \in \mathcal{I}_0 \cap \mathcal{J}_1$. Since p_2 satisfies $\mathcal{S}(R_{\mathbf{T}})$ (c.f. line 16), and hence

is ϵ -consistent with $R_{\mathbf{T}}$, we get by (4) that $p_2([w, t]) \leq (1 + \epsilon)v(I, R_{\mathbf{T}})$ for all $I = [s, t] \in \mathcal{I}_0$. Combining this with the above inequality gives $p_{\mathbf{T}}(I) = p_1[s, w] + p_2[w, t] \leq B_{\mathbf{T}}(I)$ for all $I = [s, t] \in \mathcal{I}_0 \cap \mathcal{J}_1$, and hence proves (i).

Now we prove (ii). We have the following: $p'_{\mathbf{T}_L}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_L \setminus \mathcal{I}_0) = p'_{\mathbf{T}}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_L \setminus \mathcal{I}_0)$, $h_{\mathbf{T}_L}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_L \setminus \mathcal{I}_0) = h_{\mathbf{T}}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_L \setminus \mathcal{I}_0)$, $p'_{\mathbf{T}_R}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_R) = p'_{\mathbf{T}}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_R)$, $h_{\mathbf{T}_R}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_R) = h_{\mathbf{T}}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_R)$, and

$$\begin{aligned} p'_{\mathbf{T}_L}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_0) &= \sum_{I=[s,t] \in \mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_0} p'_{\mathbf{T}}([s, w]), \\ h_{\mathbf{T}_L}(\mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_0) &= \sum_{I=[s,t] \in \mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_0} h_{\mathbf{T}}([s, t]) + \sum_{I=[s,t] \in \mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_0} v(I, R_{\mathbf{T}}), \\ &\geq \sum_{I \in \mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_0} h_{\mathbf{T}}(I) + \sum_{I=[s,t] \in \mathcal{J}'_{\mathbf{T}} \cap \mathcal{I}_0} \frac{p'_{\mathbf{T}}([w, t])}{1 + \epsilon}, \end{aligned}$$

where the last inequality follows by (4). Summing all these together gives (ii) and concludes the proof of the lemma. \square

Proof of Lemma 4. Consider the pair of intervals $\{a_i, b_i\}$ for each $i = 1, \dots, 4$. The maximum profit that can be obtained from such a pair is 2, which is obtained by setting either $p(e_i) = 1$, or $p(e_i) = 2$. Any other price clearly yields a smaller profit. Similarly if we consider only the intervals of type D , the maximum profit is obtained by setting $p(e_2) + p(e_3) = 2$, or $p(e_2) + p(e_3) = 4$. This gives us 4 price vectors that give us maximum profit from all except the type C intervals, viz. $(1, 1, 1, 1)$, $(2, 2, 2, 2)$, $(1, 2, 2, 1)$, $(2, 1, 1, 2)$. In the first case, we only obtain a profit of 4 from the type C intervals for a total profit of 16, while in the second case, we exceed the budget of both the type C intervals giving us a profit of only 12. Thus there are only two profit maximizing price assignments. \square

Proof of Lemma 5. Consider the gadget for variable x_i . If the gadget is consistent, we see that both the consistency gadget, and the type H interval spend their entire budget, and we obtain a profit of $2mn^2 + 6(2i - 2) + 6 + 36$. Suppose B_i^1 is TRUE and B_i^2 is FALSE. Then, we are forced to set the price of edge h to $mn^2 - 1$, otherwise the consistency gadget is unable to purchase its edges and we lose at least $mn^2 + 6(2i - 2) + 6$ from the total profit. However, by setting $p(h) = mn^2 - 1$, the maximum profit we obtain is at most $(mn^2 - 1) + mn^2 + 6(2i - 2) + 36$, which is smaller than the maximum profit by 1 unit. On the other hand, if B_i^1 is FALSE, and B_i^2 is TRUE, we lose 1 unit from the maximum profit since we cannot raise the price of edge h to more than mn^2 , and the consistency gadget is unable to spend its entire budget. Hence, the maximum profit is obtained only when the variable gadget is consistent. \square

Proof of Lemma 6. Consider a consistent price assignment, with the edge h having a price of n^2 and a clause $(x_i \vee x_j)$. If the clause $(x_i \vee x_j)$ is not satisfied, then the gadgets for variables x_i and x_j have a FALSE price assignment, and the prices for the edges in the gadgets for x_i and x_j are $2, 1, 1, 2$, and $2, 1, 1, 2$ respectively. Then, it is easy to see that the price of the bundle of the clause interval in this case is $mn^2 + 6(i + j - 2) + 4$, exceeding the budget of the clause interval. In the other three cases, the price of the bundle is at most $mn^2 + 6(i + j - 2) + 3$, and the profit from the clause interval is at least $mn^2 + 6(i + j - 2) + 2$ (In the case when both x_i and x_j are TRUE, the profit is $mn^2 + 6(i + j - 2) + 2$, in the

two other satisfying assignments the profit is $mn^2 + 6(i + j - 2) + 3$. The proofs for the other types of clauses $(\overline{x_i} \vee x_j)$, $(x_i \vee \overline{x_j})$, and $(\overline{x_i} \vee \overline{x_j})$ are similar. \square

Proof of Theorem 5. Suppose the instance of MAX-2-SAT has k satisfied clauses. We set the prices for the edges corresponding to the two basic gadgets corresponding to the variable x_i to TRUE if $x_i = 1$ and FALSE otherwise. We set the price of edge h to n^2 . This gives a total profit of

$$S = T \cdot 36n + T \cdot \sum_{i=1}^n (mn^2 + 6(2i - 2) + 6) + T \cdot mn^2 + kmn^2 + O(kn)$$

The first term of the sum comes from the basic gadgets of each variable set to TRUE or FALSE, the second term comes from the consistency gadgets, the third term comes from the H intervals, and the last two terms, from the satisfied clause gadgets.

To show the reverse direction, consider a price assignment that achieves a profit of at least S . We claim that in an optimal price assignment, the gadgets corresponding to the variables are all consistent, and the edge h has a price of mn^2 . Note first that the maximum profit we can gain from all the clauses is $O(m^2n^2)$. Now, if we have larger than, say $T = m^3n^2$ copies of each variable gadget, it follows from Lemma 5 that we only lose by making either the variable gadgets inconsistent, or if the H intervals and the consistency gadgets do not spend their entire budget. Hence, in the optimal solution, the variables are consistent, and h has a price of mn^2 . This then leaves only the clause intervals. Note that our profit maximizing pricing will try to maximize the number of clause intervals satisfied, since the clause intervals differ by at most $O(n)$ in their budgets, but their individual budgets themselves are at least mn^2 . By the obvious assignment of truth values to the variables from the variable price assignment, we get an assignment that satisfies k clauses. \square

Appendix B: The gadgets used in the NP-hard construction

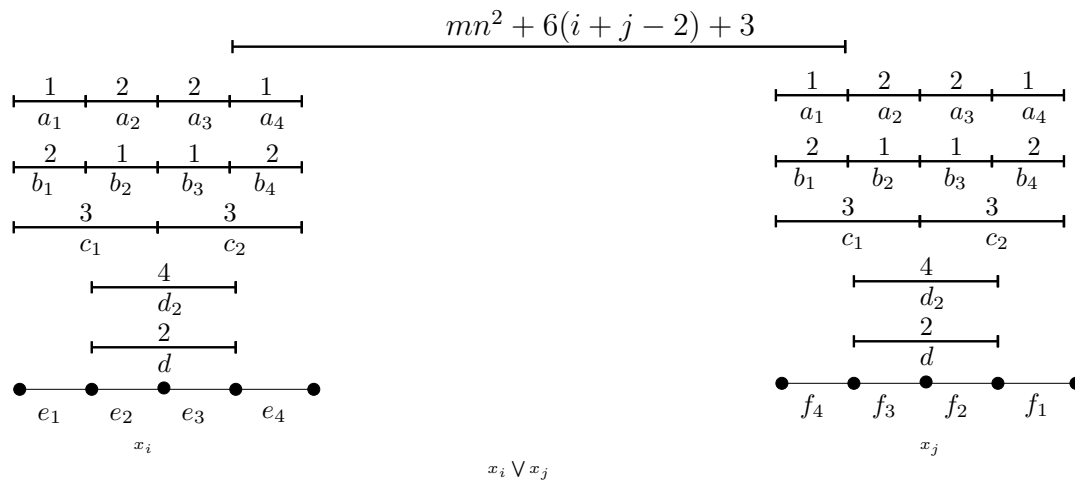


Fig. 5. The clause gadget for clause $(x_i \vee x_j)$

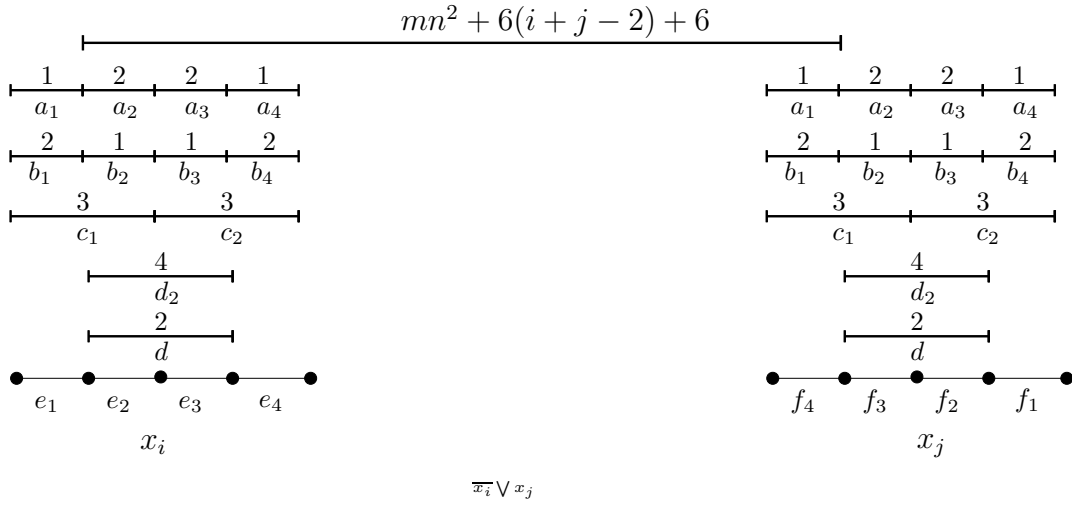


Fig. 6. The clause gadget for clause $(\overline{x_i} \vee x_j)$

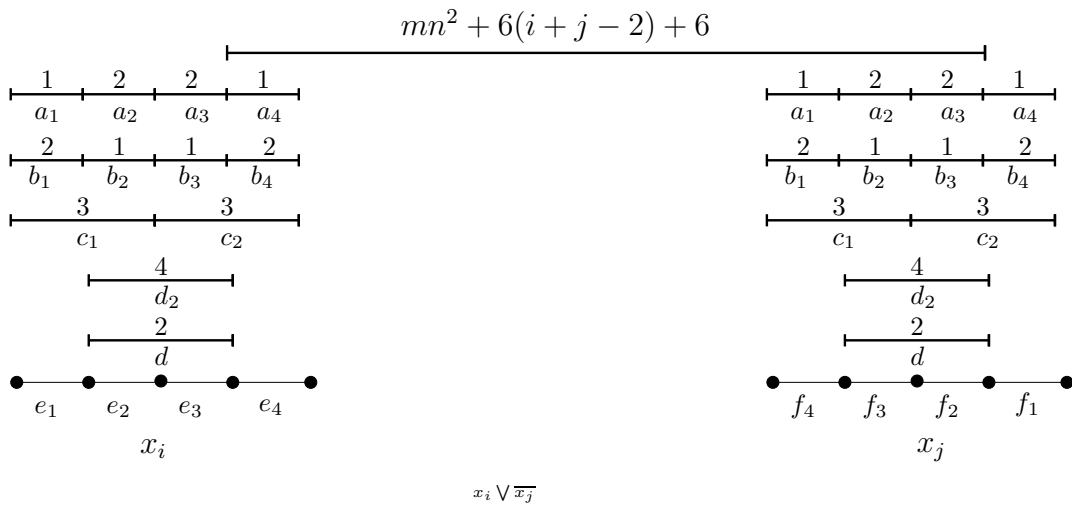


Fig. 7. The clause gadget for clause $(x_i \vee \overline{x_j})$

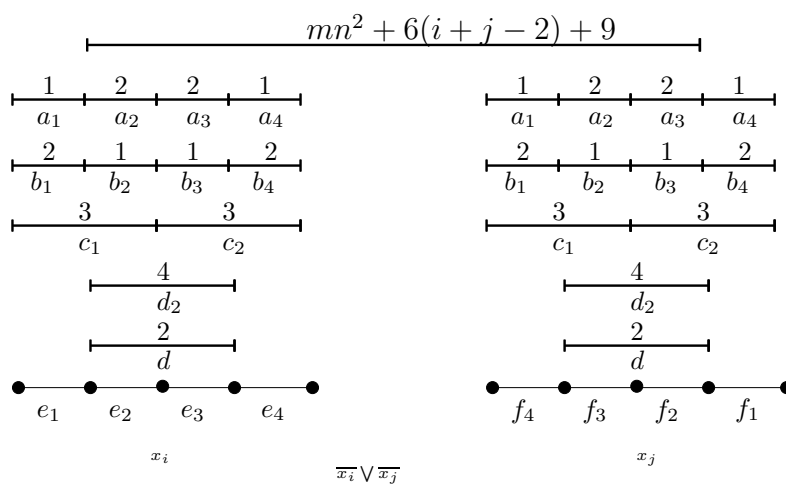


Fig. 8. The clause gadget for clause $(\overline{x_i} \vee \overline{x_j})$