Improved Hardness of Approximation for Stackelberg Shortest-Path Pricing

Patrick Briest *, Parinya Chalermsook **, Sanjeev Khanna ***, Bundit Laekhanukit †, and Danupon Nanongkai ‡

Abstract. We consider the *Stackelberg shortest-path pricing problem*, which is defined as follows. Given a graph G with fixed-cost and pricable edges and two distinct vertices s and t, we may assign prices to the pricable edges. Based on the predefined fixed costs and our prices, a customer purchases a cheapest s-t-path in G and we receive payment equal to the sum of prices of pricable edges belonging to the path. Our goal is to find prices maximizing the payment received from the customer. While Stackelberg shortest-path pricing was known to be APX-hard before, we provide the first explicit approximation threshold and prove hardness of approximation within $2 - o(1)$. We also argue that the nicely structured type of instance resulting from our reduction captures most of the challenges we face in dealing with the problem in general and, in particular, we show that the gap between the revenue of an optimal pricing and the only known general upper bound can still be logarithmically large.

1 Introduction

The notion of *algorithmic pricing* encompasses a wide range of optimization problems aiming to assign revenue-maximizing prices to some fixed set of items given information about the valuation functions of potential customers [1, 13]. In a line of recent work the approximation complexity of this kind of problem has received considerable attention.

Without supply constraints, the very simple *single-price algorithm*, which reduces the search to the one-dimensional subspace of pricings assigning identical prices to all the items, achieves an approximation guarantee of $\mathcal{O}(\log n + \log m)$, where n and m denote the number of item types and customers, respectively [4, 7]. Corresponding hardness of approximation results of $\Omega(\log^{\varepsilon} m)$ for some

* Department of Computer Science, University of Paderborn, Germany. patrick.briest@upb.de. Work done while the author was staying at Cornell University, supported by a scholarship of the German Academic Exchange Service (DAAD).

** Department of Computer Science, University of Chicago, IL, USA. parinya@uchicago.edu

*** Department of Computer and Information Science, University of Pennsylvania, PA, USA. sanjeev@cis.upenn.edu.

† Department of Computer Science, McGill University, QC, Canada. blaekh@cs.mcgill.ca

‡ College of Computing, Georgia Tech, Atlanta, GA, USA. danupon@cc.gatech.edu

$\varepsilon > 0$ are known to hold (under different complexity theoretic assumptions) even in the special cases that valuation functions are *single-minded* (items are *strict complements*) [12] or *unit-demand* (items are *strict substitutes*) [5, 8, 11]. In these cases, it is the potentially conflicting nature of different customers' valuations that constitutes the combinatorial difficulty of multi-dimensional pricing.

Another line of research has been considering so-called *Stackelberg pricing* problems [17], in which valuation functions are expressed implicitly in terms of some optimization problem. More formally, we are given a set of items, each of which has some fixed cost associated with it. In addition to these fixed costs, we may assign prices to a subset of the items. Given both fixed costs and prices, a single customer will purchase a min-cost subset of items subject to some feasibility constraints and we receive payment equal to the prices assigned to items purchased by the customer. As an example, we may think of items as being the edges of a graph and a customer aiming to buy a min-cost spanning tree, cheapest path, etc.

Clearly, as there is only a single customer in this type of problem, conflicting valuation functions can no longer pose a barrier for the design of efficient pricing algorithms and, indeed, there are several examples of algorithmic results breaking the logarithmic approximation barrier of the general case in situations where the optimization problem solved by the customer is of a certain type [7], the underlying graph is particularly well-structured [10] or the customer is restricted to applying a specific approximation algorithm solving her cost-minimization problem sub-optimally [6].

Yet, many central Stackelberg pricing problems - and in particular the aforementioned spanning tree and shortest path versions in their unrestricted form - have so far resisted all attempts at improving over the single-price algorithm's logarithmic approximation guarantee. At the same time, the best known hardness results to date only prove APX-hardness of both the spanning tree [9] and shortest path [14] cases without even deriving explicit constants.

1.1 Preliminaries

In the *Stackelberg shortest-path pricing problem* (StackSP), we are given a directed graph $G = (V, A)$, a cost function $c : A \to \mathbb{R}_0^+$, a distinguished set of *pricable edges* $\mathcal{P} \subset A$, $|\mathcal{P}| = m$, and two distinguished nodes $s, t \in V$. We may assign prices $p : \mathcal{P} \to \mathbb{R}_0^+$ to the pricable edges. Given these prices, a customer will purchase a shortest directed s-t-path P^* in G, i.e.,

$$P^* \in \operatorname{argmin}\Big\{\sum_{e \in P}(c(e) + p(e)) \,\Big|\, P \text{ is } s\text{-}t\text{-path}\Big\},$$

and we receive revenue $rev(p) = \sum_{e \in P^*} p(e)$. We assume w.l.o.g.[1] that in case of a tie, the customer selects from the above set a path maximizing our revenue.

[1] We can make this assumption since decreasing all prices by a factor arbitrarily close to 1 will break ties in favor of higher revenue paths.

We want to find a price assignment p maximizing $rev(p)$. Throughout the rest of this paper, we will w.l.o.g. only consider StackSP instances for which $c(e) = 0$ for all $e \in \mathcal{P}$, i.e., every edge is either pricable or fixed-cost, but never both.

1.2 Contributions

In this paper, we present the first explicit hardness of approximation result for the shortest path version of Stackelberg pricing, which we show to be hard to approximate within a factor of $2 - o(1)$. The result is based on a reduction that is somewhat similar to the ones previously described in [16] and [14] to derive NP-hardness and APX-hardness, respectively. Our contribution consists of identifying the right starting point for the reduction in order to utilize the full potential of the construction and applying a completely new analysis which yields the desired gap. This novel analysis parts completely from previous approaches, as it argues explicitly about the structure of solutions to the resulting path pricing instances. These results are found in Section 2. Despite their apparent simplicity, these instances, which we refer to as a *shortcut instances*, seem to capture most of the challenges we face in dealing with the problem in general. It is then a natural question to ask whether we can obtain improved approximation results by exploiting the special structure of these instances or the insights gained from our analysis of the structural properties of their solutions. Unfortunately, it turns out that this might not be an easy task, since we can prove that the gap between the optimal revenue and the upper bound used in all known algorithmic results can still be of essentially logarithmic size. These results are presented in Section 3.

2 Hardness of Approximation

Theorem 1. StackSP *cannot be approximated in polynomial time within a factor of* $2 - 2^{-\Omega(\log^{1-\varepsilon} m)}$ *for any* $\varepsilon > 0$*, unless NP* $\subseteq$ *DTIME(*$n^{\mathcal{O}(\log n)}$*).*

2.1 Proof of Theorem 1

The proof of the Theorem is based on a reduction from the *label cover problem* (LabelCover), which is defined as follows. Given a bipartite graph $G = (V, W, E)$, a set $L = \{1, \dots, k\}$ of *labels* and a set $R_{(v,w)} \subseteq L \times L$ of satisfying label combinations for every edge $(v, w) \in E$, we want to find a label assignment $\ell : V \cup W \to L$ to the vertices of G satisfying the maximum possible number of edges, i.e., edges (v, w) with $(\ell(v), \ell(w)) \in R_{(v,w)}$. The following hardness result for LabelCover, which is an easy consequence of the PCP theorem [3] combined with Raz' parallel repetition theorem [15], is found, e.g., in the survey by Arora and Lund [2].

Theorem 2. *For* LabelCover *on graphs with* n *vertices,* m *edges and label set of size* $k = \mathcal{O}(n)$ *there exists no polynomial time algorithm to decide whether* *the maximum number of satisfiable edges is m or at most $m/2^{\log^{1-\varepsilon} m}$ for any $\varepsilon > 0$, unless NP $\subseteq$ DTIME($n^{\mathcal{O}(\log n)}$).*

Reduction: Let an instance $G = (V, W, E)$ with label set $L = \{1, \ldots, k\}$ as in Theorem 2 be given. Denote $E = \{(v_1, w_1) \ldots, (v_m, w_m)\}$, where the ordering of the edges is chosen arbitrarily. Note that in our notation, v_i, v_j with $i \neq j$ may well refer to the same vertex (and the same is true for w_i, w_j). For ease of notation we denote by R_i the satisfying label combinations for edge (v_i, w_i).

We create a StackSP instance as follows. For every edge (v_i, w_i) we construct a gadget as depicted in Fig. 1. Essentially, the gadget consist of a set of parallel pricable edges, one for each satisfying label assignment $(\kappa, \lambda) \in R_i$ and an additional parallel fixed-cost edge of price 2.

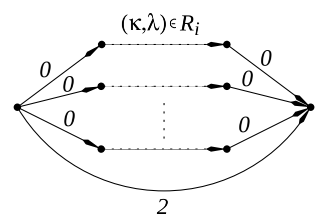

Fig. 1. Gadget for an edge (v_i, w_i) in the label cover instance. Each pricable edge corresponds to one satisfying label assignment (κ, λ) to vertices v_i, w_i.

These gadgets are joined together sequentially (see Fig. 2). Let $i < j$ and consider two pricable edges corresponding to label assignments $(\kappa, \lambda) \in R_i$ and $(\mu, \nu) \in R_j$. We connect the endpoint of the first edge with the start point of the second edge with a *shortcut edge* of cost $j - i - 1$, if the two label assignments are conflicting, i.e., if either $v_i = v_j$ and $\kappa \neq \mu$ or $w_i = w_j$ and $\lambda \neq \nu$. This construction is depicted in Fig. 2. Finally, we define the first node in the gadget corresponding to edge (v_1, w_1) and the last node in the gadget corresponding to (v_m, w_m) as nodes s and t the customer seeks to connect via a directed shortest path. We will refer to the gadgets by their indices $1, \ldots, m$ and denote the pricable edge corresponding to label assignment (κ, λ) in gadget i as $e_{i,\kappa,\lambda}$.

Completeness: Let ℓ be a label assignment satisfying all edges in G. We define a corresponding pricing p by setting for every pricable edge $p(e_{i,\kappa,\lambda}) = 2$ if $\ell(v_i) = \kappa$, $\ell(w_i) = \lambda$ and $p(e_{i,\kappa,\lambda}) = +\infty$ else.

The resulting shortest path from s to t cannot use any of the shortcut edges, because, as ℓ is a feasible label assignment, out of any two pricable edges corresponding to conflicting assignments, one must be priced at $+\infty$. Consequently, no path using a shortcut edge can have finite cost. On the other hand, since ℓ satisfies every edge, there is a pricable edge of cost 2 in each of the gadgets. It is then w.l.o.g. to assume that the customer purchases the shortest path using the maximum possible number of pricable edges and, hence, total revenue is $2m$.

Soundness: Let p be a given pricing resulting in overall revenue $m+c$ and let P denote the shortest path purchased by the customer given these prices. We will argue that there exists a label assignment ℓ satisfying $c/4$ of the edges in G.

First note that w.l.o.g. any pricable edge that is not part of path P has price $+\infty$ under price assignment p. In particular, this means that in every gadget i there is at most a single pricable edge with a finite price. We call this edge the *P-edge* of gadget i. We proceed by grouping gadgets into so-called *islands* as detailed below.

Islands: Let σ_1 be the first gadget with a P-edge and call σ_1 the *start point* of an *island*. Now for each σ_i find the maximum value of $j > \sigma_i$, such that gadget j has a P-edge and there exists a shortcut edge between the P-edges of gadgets σ_i and j. If such a j exists, define $\sigma_{i+1} = j$, else call σ_i an *end point* of an island, let $k > \sigma_i$ be the minimum value such that gadget k has a P-edge, define $\sigma_{i+1} = k$ and call σ_{i+1} a start point. If no such k exists, call σ_i an end point and stop. Let σ_r be the end point of the final island.

We call $\sigma_1, \ldots, \sigma_r$ the *significant gadgets*. Note that by construction every gadget with a P-edge is covered by some *island*, i.e., the interval defined by some consecutive start and end points.

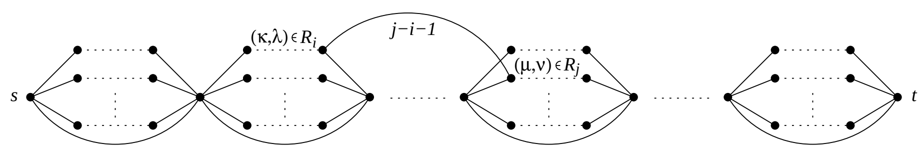

Fig. 2. Assembling the edge gadgets into a StackSP instance. Conflicting label assignments on two edges (v_i, w_i), (v_j, w_j) are connected by a shortcut of length $j-i-1$. All edges are directed from left to right.

Fact 1 *Consider an island $\sigma_\alpha, \ldots, \sigma_\omega$. Path P does not enter gadget σ_α or exit gadget σ_ω via a shortcut edge.*

Proof. If P exits σ_ω via a shortcut edge, then σ_ω could not have been declared an end point. If σ_α is entered via a shortcut edge, this shortcut must originate from a gadget $i < \sigma_\alpha$ which lies within the preceding island. As P cannot bypass the endpoint of the preceding island via a shortcut, i must in fact be the end point $\sigma_{\alpha-1}$ and so σ_α could not have become a start point. $\square$

Consider now a single island $\sigma_\alpha, \ldots, \sigma_\omega$. By ℓ_i we denote the length of the shortcut edge between gadgets σ_i and σ_{i+1} for $\alpha \leq i \leq \omega - 1$. Furthermore, by in_i and out_i we refer to the lengths of the shortcut edges used by path P to enter and exit gadget σ_i, respectively, and set them to 0 if no shortcuts are used. From Fact 1 above it follows that $in_\alpha = out_\omega = in_{\alpha+1} = 0$. See Fig. 3 for an illustration.

For $\alpha \leq i \leq \omega$ let the cost of path P between shortcut edges out_i and in_{i+1} be $r_i + c_i$, where r_i denotes the cost due to pricable edges and c_i the cost due to fixed-cost edges, respectively. We are going to bound the expression $p_{\sigma_i} + r_i$, which is the sum of prices paid for the section of path P running from gadget σ_i to $\sigma_{i+1} - 1$.

We note that $\ell_\omega = 0$, since by the fact that gadget σ_ω is an endpoint, no shortcut edge connects its P-edge to the P-edge of another gadget. Similarly, we have $r_\omega = 0$, since path P does not use pricable edges between islands, as we have argued before.

Path P crosses the end node of the P-edge in gadget σ_i (node v_2 in Fig. 3) and the start node of the P-edge of gadget σ_{i+1} (node v_4 in Fig. 3) for $\alpha \leq i \leq \omega - 1$. The total cost of path P between these two vertices is $out_i + r_i + c_i + in_{i+1}$. An alternative path P_1 is obtained by replacing this part of P with the shortcut edge of length ℓ_i between σ_i and σ_{i+1}. By the fact that P is the shortest path we have $out_i + r_i + c_i + in_{i+1} \leq \ell_i$ and, thus,

$$r_i \leq \ell_i - out_i - in_{i+1} \qquad \text{for } \alpha \leq i \leq \omega, \tag{1}$$

where the bound on r_ω follows from the fact that for $i = \omega$ all summands in the above expression are 0. Similarly, the cost of path P between the start node of the shortcut edge into gadget σ_i (node v_1 in Fig. 3) and the end node of the shortcut edge exiting σ_i (node v_3 in Fig. 3) is $in_i + p_{\sigma_i} + out_i$ for $\alpha \leq i \leq \omega$. We obtain an alternative path P_2 by taking only fixed cost edges of cost 2 to bypass both shortcuts and gadget σ_i at total cost $2(in_i + out_i + 1)$. Again, since P is the shortest path, we get $in_i + p_{\sigma_i} + out_i \leq 2(in_i + out_i + 1)$, or

$$p_{\sigma_i} \leq 2 + in_i + out_i \qquad \text{for } \alpha \leq i \leq \omega\,. \tag{2}$$

Combining (1) and (2) yields

$$p_{\sigma_i} + r_i \leq 2 + \ell_i + in_i - in_{i+1} \quad \text{for } \alpha \leq i \leq \omega\,. \tag{3}$$

Finally, we have

$$\begin{aligned} \sum_{i=\alpha}^{\omega} \Big(p_{\sigma_i} + r_i\Big) &\leq \sum_{i=\alpha}^{\omega} \Big(2 + \ell_i + in_i - in_{i+1}\Big) & (4)\\ &= 2(\omega - \alpha + 1) + \sum_{i=\alpha}^{\omega} \ell_i + in_\alpha - in_{\omega+1} & (5)\\ &= 2(\omega - \alpha + 1) + \sum_{i=\alpha}^{\omega} \ell_i, & (6) \end{aligned}$$

where (6) holds since start points σ_α and $\sigma_{\omega+1}$ are not entered via shortcuts and, thus, $in_\alpha = in_{\omega+1} = 0$. Recall that $\sigma_1, \ldots, \sigma_r$ denote the significant gadgets across all islands. Assume now that there is a total number I of islands with start and end points $\sigma_{\alpha(1)}, \sigma_{\omega(1)}, \ldots, \sigma_{\alpha(I)}, \sigma_{\omega(I)}$. Summing over all islands we get that overall revenue of price assignment p is bounded by

$$\sum_{j=1}^{I} \sum_{i=\alpha(j)}^{\omega(j)} p_{\sigma_i} + r_i \leq \sum_{j=1}^{I} \Big(2\big(\omega(j) - \alpha(j) + 1\big) + \sum_{i=\alpha(j)}^{\omega(j)} \ell_i \Big) \leq 2r + m,$$

where the last inequality follows from the fact that $\alpha(j) = \omega(j-1) + 1$ for $2 \leq j \leq I$, $\omega(I) = r$ and $\sum_{i=1}^{r} \ell_i \leq m$, since all shortcuts defining the ℓ_i are disjoint. Thus, we have $m + c \leq 2r + m$, or $r \geq c/2$.

So we have established a relation between the revenue of a price assignment and the number of its significant gadgets. It remains to show that a pricing's significant gadgets can be used to construct a label assignment satisfying a large number of the corresponding edges. Towards this end, consider the P-edges of the $\lceil r/2 \rceil$ gadgets $\sigma_1, \sigma_3, \sigma_5, \ldots$ and their corresponding label assignments (κ_i, λ_i). By definition, there are no shortcut edges between the P-edges of any of these gadgets and, thus, $(\kappa_1, \lambda_1), (\kappa_3, \lambda_3), \ldots$ define a non-conflicting label assignment satisfying at least $\lceil r/2 \rceil \geq c/4$ edges in G. More precisely, labels $(\kappa_1, \lambda_1), (\kappa_3, \lambda_3), \ldots$ can be extended into a complete label assignment satisfying $c/4$ edges by choosing all unspecified labels in an arbitrary fashion.

Finally, consider a label cover instance as in Theorem 2 and the path pricing instance resulting from our reduction above. If all edges can be satisfied, maximum path pricing revenue is $2m$. If no label assignment satisfies more than $m/2^{\log^{1-\varepsilon} m}$ edges, maximum path pricing revenue is bounded by $(1 + 4/2^{\log^{1-\varepsilon} m})m$. This completes the proof of Theorem 1. □

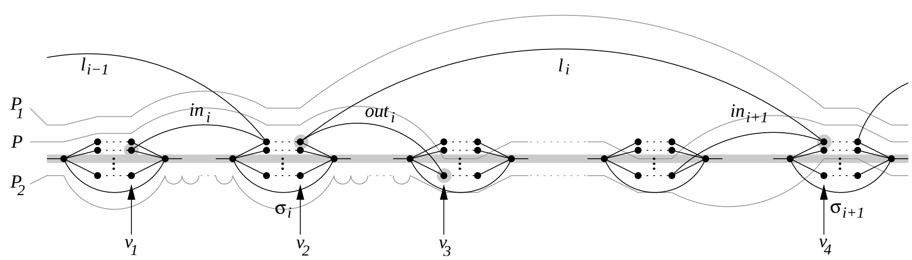

Fig. 3. Two consecutive significant gadgets σ_i, σ_{i+1} inside one island. The length of the shortcut edges used to enter and exit gadget σ_i (defined as 0 if no such shortcut exists) are denoted as in_i and out_i, respectively.

Tightness: We briefly mention that our analysis is tight in the following sense. It is easy to check that by assigning price 1 to all pricable edges we can make sure that w.l.o.g. the shortest s-t-path uses a pricable edge in each of the gadgets and, thus, we obtain revenue m. Since maximum possible revenue is bounded above by $2m$ (there is an s-t-path of that cost that does not use any pricable edges), it follows that it is trivial to achieve approximation guarantee 2 on the instances resulting from our reduction.

3 Shortcut Instances

In this section we take a closer look at the type of instances resulting from our reduction, which we believe present an important milestone in getting a further improvement in terms of hardness or algorithmic results. We will be interested in the family of so-called *shortcut instances*, which we define as follows.

We say that a *gadget* H consists of source and sink nodes connected by (i) a fixed cost edge from source to sink and (ii) node-disjoint paths of length three where each path is directed from the source to the sink, alternating between fixed-cost, pricable, and fixed-cost edges. For example, the graph in Figure 1 is a gadget with three node-disjoint paths of length three. The left and right nodes are source and sink, respectively. We call an input instance *shortcut instance*, if it can be constructed by the following two-step process:

1. Let $G_1, \ldots, G_{n'}$ be gadgets. Sequentially join them together by unifying the sink of each gadget G_i with the source of G_{i+1}. The source of G_1 is denoted as s, the sink of $G_{n'}$ as t.
2. For each pair of integers $i < j$, for each pricable edge (u, u') in G_i and each pricable edge (v, v') in G_j, we have a fixed cost edge (u', v). (Note, that we allow setting the price of edges to ∞, which is equivalent to removing them from the instance.) Edges created in this step are called *shortcuts*. The fixed cost edge of cost $j - i - 1$ in Figure 2 is an example of shortcut.

Clearly, the instances resulting from our reduction in the previous section are examples of shortcut instances. It is a natural question to ask whether one can exploit the special structure of shortcut instances to beat the $\mathcal{O}(\log n)$ approximation guarantee known for the general path pricing problem. In fact, getting a better approximation ratio for shortcut graphs would probably even yield insights into potential approaches to improving the general case.

It is, however, not clear at all how to exploit the structure of these seemingly simple instances. This is so, because in dealing with the shortcut graphs, one faces the same main barrier currently encountered in the general case: all known algorithms for the problem rely on the same upper bounding technique, which yields bounds as large as $\Theta(\log n \cdot \mathsf{OPT})$. Unfortunately, it turns out that this is also the case for shortcut graphs.

The upper bound used by previous algorithms is the quantity $f_\infty(G) - f_0(G)$ where $f_x(G)$, $x \in \mathbb{R}_0^+$, is defined to be the shortest path length in G when $p(e) = x$ for all pricable edges e.[2] It is known that $f_\infty(G) - f_0(G)$ can be as

[2] Intuitively, $f_\infty(G) - f_0(G) \geq \mathsf{OPT}$ follows from the fact that the customer will never pay more than $f_\infty(G)$ (and will do so when the pricable edges are very expensive) and part of this will be paid to the competitor who owns the fixed cost edges; the latter is at least $f_0(G)$ (when the pricable edges are very cheap).

large as $\Omega(\log n \cdot \mathsf{OPT})$ and therefore one cannot hope for a better approximation guarantee using this upper bound. We show that the same problem occurs in the family of shortcut instances.

Theorem 3. *For infinitely many n, there exists a shortcut graph G of n nodes with*

$$f_\infty(G) - f_0(G) = \Omega((\log n / \log\log n) \cdot \mathsf{OPT})\,.$$

We first describe an explicit construction of the shortcut graphs from Theorem 3 and then prove the claim.

Construction: Let $\alpha \geq 2$ be any integer and let $n = \alpha^\alpha$. We construct a graph G of $\Theta(n)$ nodes as follows.

- **Gadgets:** There are n gadgets, each of which has (i) a fixed cost edge of length 1 from source to sink and (ii) a path alternating between two fixed-cost edges and a pricable edge where fixed cost edges have price 0 (see Fig. 4).
- **Shortcuts:** For a shortcut from gadget i to gadget j, the price is $(j-i)\cdot(k/\alpha)$ where k is chosen such that $\alpha^{k-1} \leq j - i < \alpha^k$; in this case, we additionally say that the shortcut is of *type* k. Observe that $1 \leq k \leq \alpha$.

We denote by (a_i, b_i) the pricable edge in gadget i. For the ease of referencing in the future, we let b_0 and b_{n+1} denote s and t respectively. Moreover, we add a shortcut of type α (i.e., of cost n) from b_0 to b_{n+1}.

Analysis: If all pricable edges have cost ∞ then, since all shortcuts are blocked, the shortest path will consist of all the gadgets' fixed cost edges and, thus, $f_\infty(G) = n$. If all pricable edges have cost zero then the shortest path will use all pricable edges and, hence, $f_0(G) = 0$. Therefore, $f_\infty(G) - f_0(G) = n$. We now prove that $\mathsf{OPT} = O(n/\alpha)$. This yields the theorem since $n = \alpha^\alpha$ implies that $\alpha = \Omega(\log n / \log\log n)$.

Let p be any pricing and P be the shortest path purchased by the customer given this pricing. Let $\delta_1, \ldots, \delta_r$ be the indices of gadgets that contain pricable edges on P (P-edges), so the revenue is collected from edges $(a_{\delta_1}, b_{\delta_1}), \ldots, (a_{\delta_r}, b_{\delta_r})$. Let $\delta_0 = 0$ and $\delta_{r+1} = n + 1$.

The following lemma bounds the price of each pricable edge.

Lemma 1. *For any $1 \leq i \leq r$, $p(a_{\delta_i} b_{\delta_i}) = O((\delta_{i+1} - \delta_{i-1})/\alpha)$.*

The fact that $\mathsf{OPT} \leq O(n/\alpha)$ follows as an easy consequence of the lemma, since

$$\sum_{i=1}^{r} p(a_{\delta_i}, b_{\delta_i}) \leq \sum_{i=1}^{r} O\left(\frac{\delta_{i+1} - \delta_{i-1}}{\alpha}\right) \leq O(n/\alpha).$$

Proof of Lemma 1: Let P' be the subpath of P from $b_{\delta_{i-1}}$ to a_{δ_i}, P'' be the subpath from b_{δ_i} to $a_{\delta_{i+1}}$. Let k be the type of shortcut $(b_{\delta_{i-1}}, a_{\delta_{i+1}})$. Notice that $p(a_{\delta_i}, b_{\delta_i}) + c(P') + c(P'') \leq (\delta_{i+1} - \delta_{i-1})k/\alpha$, because the customer will buy shortcut $(b_{\delta_{i-1}}, a_{\delta_{i+1}})$ instead otherwise. Now proving the following claim yields the lemma.

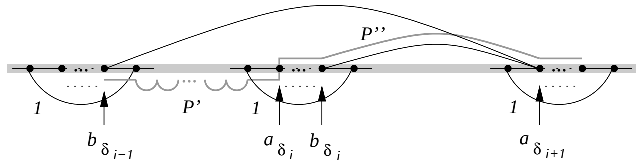

Fig. 4. Proof idea of Lemma 1.

Claim. $c(P') + c(P'') \geq \big((\delta_{i+1} - \delta_{i-1})(k-2)\big)/\alpha$.

Since both P' and P'' do not contain pricable edges, there are only two possibilities for each of them: path P' either takes the shortcut $(b_{\delta_{i-1}}, a_{\delta_i})$ or takes a sequence of fixed cost edges in gadgets $\delta_{i-1}+1, \ldots, \delta_i - 1$ (similarly for P''), and since the first option always costs less, we assume that both P' and P'' take the first option (i.e., take shortcuts).

Our first simple observation is that at least one of P' and P'' is of type at least $(k-1)$. To see this, note that assuming the contrary, we have that $(\delta_i - \delta_{i-1}) < \alpha^{k-2}$ and $(\delta_{i+1} - \delta_i) < \alpha^{k-2}$, so, adding them up, $\delta_{i+1} - \delta_{i-1} < \alpha^{k-1}$ (because $\alpha \geq 2$), contradicting the assumption that the shortcut $(b_{\delta_{i-1}}, a_{\delta_{i+1}})$ is of type k.

There are two cases to analyze now. The first case is when both P' and P'' are shortcuts of type at least $k-1$. In this case, we have,

$$\begin{aligned} c(P') + c(P'') &\geq (\delta_i - \delta_{i-1})(k-1)/\alpha + (\delta_{i+1} - \delta_i)(k-1)/\alpha \\ &\geq (\delta_{i+1} - \delta_{i-1})(k-1)/\alpha\,. \end{aligned}$$

In the second case, assume w.l.o.g. that P' is of type at most $k-2$. (This means that P'' is of type at least $k-1$ by our previous observation.) So, $\delta_i - \delta_{i-1} < \alpha^{k-2}$, while $\delta_{i+1} - \delta_{i-1} \geq \alpha^{k-1}$. Consequently, $\delta_i - \delta_{i-1} \leq \frac{1}{\alpha}(\delta_{i+1} - \delta_{i-1})$. Therefore, we get

$$\begin{aligned} \delta_{i+1} - \delta_i &= (\delta_{i+1} - \delta_{i-1}) - (\delta_i - \delta_{i-1}) \\ &\geq (1 - 1/\alpha)(\delta_{i+1} - \delta_{i-1})\,. \end{aligned}$$

Since P'' is of type at least $k-1$, we have

$$\begin{aligned} c(P'') &\geq \frac{(1-1/\alpha)(\delta_{i+1} - \delta_{i-1})(k-1)}{\alpha} \\ &\geq \frac{(\delta_{i+1} - \delta_{i-1})(k-2)}{\alpha}\,. \end{aligned}$$

The second inequality follows because $(k-1)(1-1/\alpha) = (k-1+1/\alpha - k/\alpha)$, and $k \leq \alpha$. $\square$

4 Conclusions

We have proven the first explicit approximation threshold for any Stackelberg pricing problem. Still, the approximation threshold for this kind of problem in general - and the shortest path version in particular - is far from settled. The following questions seem to constitute fertile ground for future research:

– Can we prove super-constant hardness of approximation results for any kind of Stackelberg pricing problem?
– Is it possible to achieve a better than logarithmic approximation guarantee for the Stackelberg shortest path pricing problem? Is there an interesting restricted set of graphs on which constant approximation factors are possible? In the context of the first of these questions, our discussion in Section 3 points to the obvious need of coming up with novel upper-bounding techniques even for restricted problem instances. Some progress towards answering the second of these questions has recently been made in [10], where polynomial-time algorithms are presented for the spanning-tree pricing problem in bounded-treewidth graphs.

References

1. G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for Multi-Product Pricing. In *Proc. of 31st ICALP*, 2004.
2. S. Arora and C. Lund. Hardness Of Approximations. In *Approximation Algorithms for NP-hard Problems*, PWS Publishing Company, 1996.
3. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and Hardness of Approximation Problems. *Journal of the ACM*, **45**(3):501–555, 1998.
4. N. Balcan, A. Blum, and Y. Mansour. Item Pricing for Revenue Maximization. In *Proc. of 9th EC*, 2008.
5. P. Briest. Uniform Budgets and the Envy-Free Pricing Problem. In *Proc. of 35th ICALP*, 2008.
6. P. Briest, L. Gualà, M. Hoefer, and C. Ventre. On Stackelberg Pricing with Computationally Bounded Consumers. In *Proc. of 5th WINE*, 2009.
7. P. Briest, M. Hoefer, and P. Krysta. Stackelberg Network Pricing Games. In *Proc. of 25th STACS*, 2008.
8. P. Briest and P. Krysta. Buying Cheap is Expensive: Hardness of Non-Parametric Multi-Product Pricing. In *Proc. of 18th SODA*, 2007.
9. J. Cardinal, E. Demaine, S. Fiorini, G. Joret, S. Langerman, I. Newman, and O. Weimann. The Stackelberg Minimum Spanning Tree Game. In *Proc. of 10th WADS*, 2007.
10. J. Cardinal, E. Demaine, S. Fiorini, G. Joret, I. Newman, and O. Weimann. The Stackelberg Minimum Spanning Tree Game on Planar and Bounded-Treewidth Graphs. In *Proc. of 5th WINE*, 2009.
11. J. Chuzhoy, S. Kannan and S. Khanna. Network Pricing for Multicommodity Flows. *Unpublished manuscript*, 2007.
12. E.D. Demaine, U. Feige, M.T. Hajiaghayi, and M.R. Salavatipour. Combination Can Be Hard: Approximability of the Unique Coverage Problem. In *Proc. of 17th SODA*, 2006.

13. V. Guruswami, J.D. Hartline, A.R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On Profit-Maximizing Envy-Free Pricing. In *Proc. of 16th SODA*, 2005.
14. G. Joret. Stackelberg Network Pricing is Hard to Approximate. *Networks n/a*, doi: 10.1002/net.20391, 2010.
15. R. Raz. A Parallel Repetition Theorem. *SIAM Journal on Computing*, **27**, 1998.
16. S. Roch, G. Savard, and P. Marcotte. An Approximation Algorithm for Stackelberg Network Pricing. *Networks*, 46(1): 57–67, 2005.
17. H. von Stackelberg. *Marktform und Gleichgewicht (Market and Equilibrium)*. Verlag von Julius Springer, Vienna, 1934.