

Improved Hardness Results for Profit Maximization Pricing Problems with Unlimited Supply

Parinya Chalermsook^{*†} Julia Chuzhoy[‡] Sampath Kannan[§] Sanjeev Khanna[¶]

Abstract

We consider profit maximization pricing problems, where we are given a set of m customers and a set of n items. Each customer c is associated with a subset $S_c \subseteq [n]$ of items of interest, together with a budget B_c , and we assume that there is an unlimited supply of each item. Once the prices are fixed for all items, each customer c buys a subset of items in S_c , according to its buying rule. The goal is to set the item prices so as to maximize the total profit.

We study the unit-demand min-buying pricing (UDP_{MIN}) and the single-minded pricing (SMP) problems. In the former problem, each customer c buys the cheapest item $i \in S_c$, if its price is no higher than the budget B_c , and buys nothing otherwise. In the latter problem, each customer c buys the whole set S_c if its total price is at most B_c , and buys nothing otherwise. Both problems are known to admit $O(\min\{\log(m+n), n\})$ -approximation algorithms. We prove that they are $\log^{1-\epsilon}(m+n)$ hard to approximate for any constant ϵ , unless $\text{NP} \subseteq \text{DTIME}(n^{\log^\delta n})$, where δ is a constant depending on ϵ . Restricting our attention to approximation factors depending only on n , we show that these problems are $2^{\log^{1-\delta} n}$ -hard to approximate for any $\delta > 0$ unless $\text{NP} \subseteq \text{ZPTIME}(n^{\log^{\delta'} n})$, where δ' is some constant depending on δ . We also prove that restricted versions of UDP_{MIN} and SMP , where the sizes of the sets S_c are bounded by k , are $k^{1/2-\epsilon}$ -hard to approximate for any constant ϵ .

We then turn to the Tollbooth Pricing problem, a special case of SMP , where each item corresponds to an edge in the input graph, and each set S_c is a simple path in the graph. We show that Tollbooth Pricing is at least as hard to approximate as the Unique Coverage problem, thus obtaining an $\Omega(\log^\epsilon n)$ -hardness of approximation, assuming $\text{NP} \not\subseteq \text{BPTIME}(2^{n^\delta})$, for any constant δ , and some constant ϵ depending on δ .

^{*}Department of Computer Science, University of Chicago, Chicago, IL and IDSIA, Lugano, Switzerland. Email: parinya@cs.uchicago.edu. Supported in part by NSF CAREER grant CCF-0844872, Swiss National Science Foundation project 200020-122110/1, and Hasler Foundation Grant 11099

[†]Part of this work was done while at University of Chicago.

[‡]Toyota Technological Institute, Chicago, IL 60637. Email: cjulia@ttic.edu. Supported in part by NSF CAREER grant CCF-0844872 and Sloan Research Fellowship.

[§]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA. Email: kannan@cis.upenn.edu. Supported in part by the National Science Foundation grant CCF-1137084

[¶]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA. Email: sanjeev@cis.upenn.edu. Supported in part by the National Science Foundation grants CCF-1116961 and IIS-0904314

1 Introduction

We study profit maximization pricing problems in the unlimited supply model. In these problems, we are given a set of m customers and a set of n items, where each customer c is associated with a budget B_c , and a subset $S_c \subseteq [n]$ of items it is interested in. Our goal is to set a price $p(i)$ for each item $i \in [n]$, so as to maximize the total revenue. Once the prices for the items are set, each customer c chooses a subset of items in S_c to buy, using its *buying rule*. We assume that we are given an unlimited supply of each item.

One of the most natural buying rules is the unit-demand min-buying rule, where each customer $c \in [m]$ buys the cheapest item $i \in S_c$ (breaking ties arbitrarily), provided that the price $p(i) \leq B_c$. We refer to the corresponding pricing problem as UDP_{MIN} . This problem was first introduced by Rusmevichientong et al. [22, 23], and subsequently Aggarwal et al. [1] have shown an $O(\log m + \log n)$ -approximation algorithm for it.

The second problem that we consider is Single-Minded Pricing (SMP). Here, each customer c buys the whole set S_c of items if its total price does not exceed its budget B_c , and buys nothing otherwise. This problem was introduced by Guruswami et al. [17], who also show that the techniques of [1] can be used to obtain an $O(\log m + \log n)$ -approximation algorithm for SMP. Hartline and Koltun [18] gave a $(1 + \epsilon)$ -approximation algorithm for both UDP_{MIN} and SMP when the number of items n is constant.

We remark that for pricing problems, it is natural to assume that the number of customers is much higher than the number of items, that is, $m \gg n$. Even though both UDP_{MIN} and SMP admit logarithmic approximation algorithms in terms of $(m + n)$, if we restrict ourselves to approximation factors depending only on n , nothing better than the trivial $O(n)$ -approximation is known.

On the negative side, Briest [5] has shown that both UDP_{MIN} and SMP are $\max\{n^\delta, \log^\delta(m + n)\}$ -hard to approximate for some (small) $\delta > 0$, assuming that no randomized polynomial-time algorithms can approximate constant-degree Balanced Bipartite Independent Set to within arbitrarily small constant factors. He also showed similar results under an assumption that slightly strengthens Feige’s Random 3SAT hypothesis [13].

In this paper, we show that both UDP_{MIN} and SMP are $\log^{1-\epsilon}(m + n)$ hard to approximate for any constant ϵ , unless $\text{NP} \subseteq \text{DTIME}(n^{\log^{\epsilon'} n})$ for some constant ϵ' depending only on ϵ . If we restrict our attention to approximation factors as a function of n , then we show that both these problems are $2^{\log^{1-\delta} n}$ hard to approximate for any constant δ , under the assumption that $\text{NP} \not\subseteq \text{ZPTIME}(n^{\log^{\delta'} n})$, for some constant δ' depending only on δ .

We next turn to restricted versions of UDP_{MIN} and SMP, denoted by kUDP_{MIN} and kSMP respectively, where the sizes of the sets S_c are bounded by k . The kSMP problem is known to be APX-hard even for $k = 2$ [17], and Balcan and Blum [4] have shown an $O(k)$ -approximation for kUDP_{MIN} , improving on an independent work of Briest and Krysta [6], who achieved an $O(k^2)$ -approximation for the problem. As for negative results, Briest [5] has proved that kSMP is k^ϵ -hard to approximate for some constant ϵ , assuming Feige’s random 3SAT hypothesis [13], and Khandekar et al. [19] showed that the problem is $\Omega(k)$ hard to approximate for constant k , assuming the Unique Games Conjecture of Khot [20]. We show that both kUDP_{MIN} and kSMP are $k^{1/2-\epsilon}$ -hard to approximate for any constant ϵ unless $\text{P} = \text{NP}$.

Finally, we consider a special case of the SMP problem called the Tollbooth Pricing problem, where we are given a graph G , and items correspond to the edges of G . The item set S_c of every customer c is some simple path in graph G , and the goal is to set the prices of the edges, so as to maximize the revenue. Since the Tollbooth Pricing problem is a special case of SMP, it admits an $O(\log m + \log n)$ approximation [17]. The problem is APX-hard [17], and from the results of Khandekar et al. [19], it is $(2 - \epsilon)$ hard to approximate even on star graphs, assuming the Unique Games Conjecture. We show that the Tollbooth Pricing problem is at least as hard to approximate as the Unique Coverage problem (to within a constant factor). In the Unique Coverage problem, we are given a collection U of n elements, and a family \mathcal{S} of subsets of elements of U . The goal is to find a family $\mathcal{S}' \subseteq \mathcal{S}$ of element subsets, maximizing the number of elements that are covered by exactly one subset in \mathcal{S}' . The problem was introduced and studied by Demaine et al. [10], who showed that for any arbitrarily small constant δ , if $\text{NP} \not\subseteq \text{BPTIME}(2^{n^\delta})$, then Unique Coverage is hard to approximate to within a factor of $\Omega(\log^\epsilon n)$, where ϵ is some constant depending on δ . They also showed that the problem is hard to approximate to within $\Omega(\log^{1/3-\epsilon} n)$ for any ϵ assuming the Random 3SAT Hypothesis of Feige [13], and proved additional hardness results using a hypothesis about Balanced Bipartite Independent Set. Our reduction immediately implies similar hardness results for the Tollbooth Pricing problem.

Related Work. Briest and Krysta [6] considered a more general version of UDP_{MIN} , where customers are allowed to have different budgets (valuations) for different items. They show an $\Omega(\log^\epsilon n)$ -hardness for this problem for some constant ϵ , unless $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$, and an n^ϵ -hardness for some constant $\epsilon > 0$, unless $\text{NP} \subseteq \text{DTIME}(2^{O(n^\delta)})$ for all $\delta > 0$.

A special case of the Tollbooth Pricing problem, called the Highway Problem, where the input graph is restricted to be a path, has received a significant amount of attention. Elbassioni et al. [11] showed that the problem is strongly NP-hard. On the algorithmic side, Balcan and Blum [4] have shown an $O(\log n)$ -approximation algorithm, and Elbassioni et al. [12] have proposed a QPTAS. Subsequently, Grandoni and Rothvoss [16] have shown a PTAS for the problem. For the special case of the Tollbooth Pricing problem where the input graph is a tree, the best known approximation ratio is $O(\log n / \log \log n)$, due to Gamzu and Segev [15]. However, when the number of leaves in the tree is bounded by a constant, the problem admits a PTAS [16].

Pricing problems with limited supply have also received a considerable amount of attention; Please refer to, e.g., [7, 9, 8] and references therein.

Our Results. We start by formally stating the pricing problems we consider. We are given a set of m customers and a set of n items, where each customer $c \in [m]$ is associated with a set $S_c \subseteq [n]$ of items and a budget B_c . Given a setting $\{p(i)\}_{i \in [n]}$ of item prices, every customer selects a subset $S'_c \subseteq S_c$ of items to buy according to its buying rule, and our goal is to maximize the total profit, $\sum_{c \in [m]} \sum_{i \in S'_c} p(i)$. In the UDP_{MIN} problem, the buying rule of the customers is defined as follows. Each customer $c \in [m]$ buys the cheapest item $i \in S_c$, breaking ties arbitrarily, if $p(i) \leq B_c$, and buys nothing otherwise.

In the SMP problem, each customer $c \in [m]$ purchases the whole set S_c if $\sum_{i \in S_c} p(i) \leq B_c$, and purchases nothing otherwise. Our main result is summarized in the following theorem.

Theorem 1 UDP_{MIN} and SMP are $\log^{1-\epsilon}(m+n)$ -hard to approximate for any constant $\epsilon > 0$, unless $\text{NP} \subseteq \text{DTIME}(n^{(\log n)^{\epsilon'}})$, where ϵ' is some constant depending only on ϵ . Moreover, assuming that $\text{NP} \not\subseteq \text{ZPTIME}(n^{(\log n)^{\delta'}})$, both problems are hard to approximate to within a factor of $2^{\log^{1-\delta} n}$ for any constant δ , where δ' is some constant depending only on δ .

We next turn to special cases of both problems, denoted by kUDP_{MIN} and kSMP respectively, where the sizes of the sets S_c are bounded by k and prove the following theorem.

Theorem 2 Let $\epsilon > 0$ be any constant. Then for infinitely many constants k , both kUDP_{MIN} and kSMP are $k^{1/2-\epsilon}$ -hard to approximate unless $\text{P} = \text{NP}$.

Finally we turn to the Tollbooth Pricing problem. In this problem, we are given a graph $G = (V, E)$, and a set of m simple paths P_1, \dots, P_m , where each path P_c is associated with a customer c and a budget B_c . Once the price function $p : E \rightarrow \mathbb{R}$ on the edges is set, each customer c buys all edges on the path P_c if $\sum_{e \in P_c} p(e) \leq B_c$, and buys nothing otherwise. The goal is to compute the edge prices $p(e)$ so as to maximize the total profit. It is clear that Tollbooth Pricing is a special case of SMP , and notice that the number of items is $n = |E(G)|$.

We perform a reduction from the Unique Coverage problem to the Tollbooth Pricing. In the Unique Coverage problem, we are given a set U of elements and a family \mathcal{S} of subsets of U as input. A solution is a sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ of the input sets. We say that element $u \in U$ is *satisfied* by the solution if and only if it belongs to exactly one set in \mathcal{S}' . Our goal is to choose \mathcal{S}' so as to maximize the number of satisfied elements. Demaine et. al. [10] have shown that for any arbitrarily small constant δ , if $\text{NP} \not\subseteq \text{BPTIME}(2^{n^\delta})$, then Unique Coverage is hard to approximate to within a factor of $\Omega(\log^\epsilon n)$, for some constant ϵ depending on δ . They also showed that, under the assumption of Feige [13] that refuting random instances of 3SAT is hard, Unique Coverage is hard to approximate to within a factor of $\Omega(\log^{1/3-\epsilon} n)$ for any $\epsilon > 0$. We prove the following theorem:

Theorem 3 If there is a factor α -approximation algorithm for the Tollbooth Pricing problem, for any approximation factor $\alpha \leq O(\log n)$, then there is a randomized $O(\alpha)$ -approximation algorithm for the Unique Coverage problem.

Combining this with the result of [10], we obtain the following corollary.

Corollary 4 For any arbitrarily small constant δ , if $\text{NP} \not\subseteq \text{BPTIME}(2^{n^\delta})$, Tollbooth Pricing is hard to approximate to within a factor of $\Omega(\log^\epsilon n)$ for some constant ϵ depending on δ . Moreover, under Feige's random 3SAT assumption, this problem is hard to approximate to within a factor of $\Omega(\log^{1/3-\epsilon} n)$ for any $\epsilon > 0$.

2 Hardness of UDP_{MIN} and SMP

In this section we prove Theorems 1 and 2. We focus here on the UDP_{MIN} problem only. The hardness results for SMP are obtained using similar ideas and are deferred to Appendix.

We start with the following theorem, due to Trevisan [25]. Since we use slightly different parameters, we provide the proof in the Appendix.

Theorem 5 *Given an n -variable 3SAT formula φ , any sufficiently small constant $\epsilon > 0$ and any integer $\lambda > 0$, there is a randomized algorithm to construct a graph G with maximum degree at most $\Delta = 2^{\lambda \text{poly}(\frac{1}{\epsilon})}$ such that w.h.p.:*

- (YES-INSTANCE:) *If φ is satisfiable, then G has an independent set of size $|V(G)|/\Delta^\epsilon$.*
- (NO-INSTANCE:) *If φ is not satisfiable, then G has no independent set of size $|V(G)|/\Delta^{1-\epsilon}$.*

The construction size is $|V(G)| = n^{\lambda \text{poly}(\frac{1}{\epsilon})}$, and the reduction runs in time $n^{\lambda \text{poly}(\frac{1}{\epsilon})}$. Moreover, the algorithm can be made deterministic with running time $2^{O(\Delta)} n^{\lambda \text{poly}(\frac{1}{\epsilon})}$.

We remark that this theorem allows us to adjust parameter λ . To prove Theorem 1, we will use $\lambda = O(\log \log n)$, while we set $\lambda = O(1)$ for Theorem 2.

2.1 The Construction

Let $G = (V, E)$ be the instance of Maximum Independent Set obtained from Theorem 5, where the value of λ (and Δ) will be fixed later. We first define an intermediate instance of UDP_{MIN} , which is then converted into a final instance.

The intermediate instance is defined as follows. The set of items contains, for each vertex $v \in V$, for each index $y \in [\Delta]$, an item $i(v, y)$. That is, the set of items is $\mathcal{I} = \{i(v, y) \mid v \in V, y \in [\Delta]\}$.

Similarly, the set of customers contains, for each vertex $v \in V$, for each index $x \in [\Delta]$, a customer $c(v, x)$. That is, the set of customers is $\mathcal{C} = \{c(v, x) \mid v \in V, x \in [\Delta]\}$.

The item set $S_{c(v, x)}$ for the customer $c(v, x)$, contains the item $i(v, x)$, and additionally, for each neighbor u of vertex v in graph G , for each index $y \in [\Delta]$, item $i(u, y)$ belongs to $S_{c(v, x)}$. Formally, $S_{c(v, x)} = \{i(u, y) \mid (u, v) \in E, y \in [\Delta]\} \cup \{i(v, x)\}$. Notice that $|S_{c(v, x)}| \leq \Delta^2 + 1$ for all customers $c(v, x) \in \mathcal{C}$. Moreover for each item $i(v, y) \in \mathcal{I}$, there are at most $\Delta^2 + 1$ customers $c' \in \mathcal{C}$ such that $i(v, y) \in S_{c'}$.

We partition the set \mathcal{C} of customers into Δ subsets $\mathcal{C}_1, \dots, \mathcal{C}_\Delta$, such that for each $1 \leq h \leq \Delta$, set \mathcal{C}_h contains customers $c(v, h)$ for all $v \in V$. Finally, for each $1 \leq h \leq \Delta$, each customer $c \in \mathcal{C}_h$ is assigned budget $B_c = 1/2^h$.

This finishes the definition of the intermediate instance. For convenience, we call the customers in set \mathcal{C} *virtual customers*. In our final instance, we replace each virtual customer with a number of new customers.

In order to define the final instance, for each $1 \leq h \leq \Delta$, we replace each virtual customer $c \in \mathcal{C}_h$ with a set $\mathcal{G}(c) = \{c(1), \dots, c(2^h)\}$ of 2^h identical new customers. Each new customer $c(h')$, for $1 \leq h' \leq 2^h$ has budget $B_{c(h')} = B_c$ and $S_{c(h')} = S_c$. The final set of customers is $\mathcal{C}' = \bigcup_{c \in \mathcal{C}} \mathcal{G}(c)$ and the final set of items remains unchanged, $\mathcal{I}' = \mathcal{I}$. The number of customers in the final instance is $\tilde{m} = |\mathcal{C}'| = O(2^\Delta |\mathcal{C}|) = |V| \cdot \Delta \cdot 2^{O(\Delta)} = |V| \cdot 2^{O(\Delta)}$, while the number of items is $\tilde{n} = |V| \cdot \Delta$. Moreover, for each customer $c \in \mathcal{C}'$, we have $|S_c| \leq \Delta^2 + 1$. This completes the construction description.

2.2 Analysis

We analyze the construction in the following two lemmas.

Lemma 6 *In the YES-INSTANCE, there is a solution to the UDP_{MIN} problem instance whose value is at least $|V|\Delta^{1-\epsilon}$.*

Proof: Let $U \subseteq V$ be a maximum independent set of size $|V|/\Delta^\epsilon$ in G . We set the prices of the items $i(u, y) \in \mathcal{I}'$ as follows. If $u \notin U$, then the price of $i(u, y)$ is set to ∞ . Otherwise, if $u \in U$, then we set the price of $i(u, y)$ to $1/2^y$. Notice that, since $|U| \cdot \Delta \geq |V| \cdot \Delta^{1-\epsilon}$, there are $|V| \cdot \Delta^{1-\epsilon}$ items of finite prices. We now show that this solution has value at least $|U|\Delta \geq |V| \cdot \Delta^{1-\epsilon}$.

Indeed, for each vertex $u \in U$ and an index $y \in [\Delta]$, consider the virtual customer $c' = c(v, y) \in \mathcal{C}_y$. Notice that $S_{c'}$ contains item $i(v, y)$ whose price is $1/2^y$, but all other items in $S_{c'}$ have price ∞ . Therefore, each customer $c \in \mathcal{G}(c')$ buys the item $i(v, y)$, and pays $1/2^y$ for it. The total profit collected from customers in $\mathcal{G}(c')$ is 1, and so the total profit collected from all customers is at least $|U|\Delta \geq |V| \cdot \Delta^{1-\epsilon}$. \square

Lemma 7 *In the NO-INSTANCE, the value of the optimal solution is at most $O(|V| \cdot \Delta^\epsilon)$.*

Proof: Let p^* be an optimal solution, and let r^* be its revenue. We first argue that we can assume w.l.o.g. that for each item $i \in \mathcal{I}'$, either $p^*(i) \in \{1/2^h \mid 1 \leq h \leq \Delta\}$, or $p^*(i) = \infty$.

Indeed, suppose there is an item $i \in \mathcal{I}'$ with $p^*(i) \in (1/2^h, 1/2^{h-1})$. Then any customer who buys item i must have budget at least $1/2^{h-1}$, so increasing $p^*(i)$ to $1/2^{h-1}$ does not affect these customers, and may only increase the revenue. Therefore, from now on we assume that for each item $i \in \mathcal{I}'$, $p^*(i) \in \{1/2^h \mid 1 \leq h \leq \Delta\} \cup \{\infty\}$.

Notice that for each virtual customer $c \in \mathcal{C}$, all customers in $\mathcal{G}(c)$ contribute the same amount to the total revenue. Let k_c denote this amount. We now let $\mathcal{C}^* \subseteq \mathcal{C}$ be the set of virtual customers for which $k_c = B_c$. Equivalently,

$$\mathcal{C}^* = \left\{ c \in \mathcal{C} : \min_{i \in S_c} \{p^*(i)\} = B_c \right\}$$

Claim 8 *The customers in $\bigcup_{c' \in \mathcal{C}^*} \mathcal{G}(c')$ contribute at least $r^*/2$ to the total revenue.*

Proof: Assume otherwise. Then the customers in $\bigcup_{c \in \mathcal{C} \setminus \mathcal{C}^*} \mathcal{G}(c)$ contribute more than $r^*/2$ to the total revenue.

We partition the set $\mathcal{C} \setminus \mathcal{C}^*$ of virtual customers into two subsets: \mathcal{C}_1 contains all virtual customers $c \in \mathcal{C} \setminus \mathcal{C}^*$ with $\min_{i \in S_c} \{p^*(i)\} > B_c$, while \mathcal{C}_2 contains all virtual customers $c \in \mathcal{C} \setminus \mathcal{C}^*$ with $\min_{i \in S_c} \{p^*(i)\} \leq B_c/2$. Notice that $\mathcal{C} \setminus \mathcal{C}^* = \mathcal{C}_1 \cup \mathcal{C}_2$.

Define a new price $\tilde{p} = 2p^*$, where the prices of all items are scaled up by factor 2. For each virtual customer $c \in \mathcal{C}_1$, the total contribution of the customers in $\mathcal{G}(c)$ remains 0. For each virtual customer $c \in \mathcal{C}_2$, the total contribution of the customers in \mathcal{G} now doubles. Therefore, the value of the solution \tilde{p} is greater than r^* , a contradiction. \square

Notice that $|\mathcal{C}^*| \geq r^*/2$, since for each virtual customer $c \in \mathcal{C}^*$, the total budget of all customers in $\mathcal{G}(c)$ is 1.

From now on, we focus on finding an independent set U in graph G of size at least $(r^*/2 - |V|)/\Delta$ from \mathcal{C}^* . Since in the NO-INSTANCE, G does not contain an independent set of size more than $|V|/\Delta^{1-\epsilon}$, this implies that $(r^*/2 - |V|)/\Delta \leq |V|/\Delta^{1-\epsilon}$, and hence $r^* \leq O(|V| \Delta^\epsilon)$.

We construct an independent set $U \subseteq V(G)$, together with a partition $(\mathcal{C}^1, \mathcal{C}^2)$ of \mathcal{C}^* , as follows. Start with $U, \mathcal{C}^1, \mathcal{C}^2 = \emptyset$. We then perform Δ iterations, where in iteration y , we consider each virtual customer $c(v, y)$ in $\mathcal{C}^* \cap \mathcal{C}_y$, and do the following:

- If vertex v is already in U , we add virtual customer $c(v, y)$ into \mathcal{C}^1 .
- If vertex v is not in U and $U \cup \{v\}$ remains an independent set, we add vertex v to set U and add $c(v, y)$ to \mathcal{C}^1 . We say that $c(v, y)$ is *responsible* for adding vertex v into U .
- Otherwise, $v \notin U$, but there is a vertex $u \in U$ such that $(u, v) \in E(G)$. We add $c(v, y)$ to \mathcal{C}^2 in this case and say that vertex u prevents the algorithm from adding v into U .

In the end, when all customers in \mathcal{C}^* are processed, each virtual customer in \mathcal{C}^* is added to either \mathcal{C}^1 or \mathcal{C}^2 , so $\mathcal{C}^* = \mathcal{C}^1 \cup \mathcal{C}^2$. Moreover, for each virtual customer $c(v, y)$ in \mathcal{C}^1 , the corresponding vertex v belongs to U , so $|U| \geq |\mathcal{C}^1|/\Delta$. The following claim will complete the proof of the lemma.

Claim 9 $|\mathcal{C}^2| \leq |V|$, and so $|U| \geq |\mathcal{C}^* \setminus \mathcal{C}^2|/\Delta \geq (r^*/2 - |V|)/\Delta$.

Proof: It is sufficient to show that for each vertex $v \in V$, no two virtual customers $c(v, y), c(v, y')$ with $y \neq y'$ belong to \mathcal{C}^2 . Assume otherwise, and let $c(v, y), c(v, y') \in \mathcal{C}^2$. By our construction, we have $c(v, y) \in \mathcal{C}_y$ and $c(v, y') \in \mathcal{C}_{y'}$. Assume w.l.o.g. that $y < y'$, so $c(v, y)$ was processed before $c(v, y')$.

Let $u \in U$ be a vertex such that $(u, v) \in E(G)$, and vertex u prevents the algorithm from adding v to set U . Let $c(u, x)$ be the customer responsible for adding u to U . Then $c(u, x)$ was processed before $c(v, y)$, and so $x \leq y < y'$.

Notice that the item $i(v, y')$ belongs to $S_{c(u, x)}$. The price of $i(v, y')$ then must be set to at least $B_{c(u, x)} = 1/2^x > 1/2^{y'} = B_{c(v, y')}$, since otherwise the customers in $\mathcal{G}(c(u, x))$ would have paid below $B_{c(u, x)}$ for item $i(v, y')$, contradicting the fact that $c(u, x) \in \mathcal{C}^*$. But then customer $c(v, y')$ must buy some item $i' \neq i(v, y')$. Assume that $i' = i(w, z)$. Then w must be a neighbor of v in G , $w \neq v$, and so $i' \in S_{c(v, y)}$ must hold. But the price of i' must be $B_{c(v, y')} = 1/2^{y'} < 1/2^y = B_{c(v, y)}$, and so the customers in $\mathcal{G}(c(v, y))$ should have paid below $B_{c(v, y)}$ for item i' , contradicting the fact that $c(v, y) \in \mathcal{C}^*$.

□

□

Hardness factors: The gap between YES-INSTANCE and NO-INSTANCE costs is $\Delta^{1-2\epsilon}$, while the number of customers in the instance is $\tilde{m} = |V(G)| \cdot 2^{O(\Delta)}$, and the number of items is $\tilde{n} = |V(G)| \cdot \Delta$.

We first prove Theorem 1. We choose the parameter $\lambda = O(\log \log n)$ such that $\Delta = (\log n)^b$, where $b > \frac{1}{2\epsilon}$. The hardness factor then becomes $g = \Delta^{1-2\epsilon} \geq \log^{b-1} n$, while $\tilde{m} + \tilde{n} = |V(G)| 2^{O(\Delta)} \leq$

$2^{O(\Delta \log n)} \leq 2^{\log^{b+2} n} \leq 2^{g^{1+O(\epsilon)}}$. Taking logarithm on both sides will give $g = \log^{1-O(\epsilon)}(\tilde{m} + \tilde{n})$. The deterministic reduction takes time $2^{O(\Delta)} = n^{(\log n)^{f(\epsilon)}}$ for some function f , so we have proved the first part of Theorem 1.

To prove the second part, we use the randomized version of Theorem 5, and choose $\lambda = (\log n)^b$ for some large constant b , while ϵ is set to be any small enough constant for which Theorem 5 works. In this case, we have $\Delta = 2^{O((\log n)^b)}$ and $\tilde{n} \leq |V(G)|\Delta \leq 2^{(\log n)^{b+2}}$, while $g = \Delta^{1-2\epsilon} \geq 2^{O((\log n)^b)}$. It is easy to check that $g \geq 2^{\log^{1-O(1/b)} \tilde{n}}$, as desired. Since we use the randomized reduction, the running time of the reduction is $2^{(\log n)^{\tilde{O}(b)}}$, and so the result holds under the assumption that $\text{NP} \not\subseteq \text{ZPTIME}(n^{(\log n)^{O(b)}})$.

To prove Theorem 2, we choose λ in Theorem 5 to be any sufficiently large constant. Denote by $k = \max_{c \in \mathcal{C}'} |S_c|$. Since the construction guarantees that $k \leq 2\Delta^2$, we have the hardness factor of $\Delta^{1-2\epsilon} \geq k^{1/2-\epsilon}$. In this case, the deterministic reduction only takes polynomial time, so this hardness result holds under the assumption that $\text{P} \neq \text{NP}$.

3 Tollbooth Pricing

In this section we prove Theorem 3. It will be useful to introduce the notion of fractional coverage and show how to convert fractional coverage to an integral one. Given an instance of Unique Coverage and a fractional solution that assigns a non-negative weight $w(S)$ to every set $S \in \mathcal{S}$, we say that an element $u \in U$ is *fractionally covered* if and only if $1/4 \leq \sum_{S:u \in S} w(S) \leq 1$. We argue that any good fractional coverage can be converted into a good integral coverage with a constant loss in the solution value. The proof of the following lemma appears in Appendix.

Lemma 10 *There is an efficient randomized algorithm, that, given a fractional solution of value βn to any instance of the Unique Coverage problem, w.h.p. finds an integral solution of value $\Omega(\beta n)$ to the Unique Coverage instance.*

3.1 Construction

Let (U, \mathcal{S}) be an instance of Unique Coverage, where $|U| = n$ and $|\mathcal{S}| = m$. We construct an instance of Tollbooth Pricing as follows. Graph $G = (V, E)$ consists of $m+1$ vertices v_0, \dots, v_m . Let $h = \lceil \log m \rceil$. For each consecutive pair (v_{i-1}, v_i) of vertices, $0 < i \leq m$, we add $h+1$ parallel edges e_0^i, \dots, e_h^i . These edges are viewed as representing the set $S_i \in \mathcal{S}$. We now define the set of paths (or customers) in the graph. All paths start from v_0 and end at v_m . For each element $u \in U$, for each $j : 1 \leq j \leq h$, we have a set $\mathcal{P}(u, j)$ of 2^{h-j} paths. The budget of each path in $\mathcal{P}(u, j)$ is 2^j , the source vertex is v_0 , and the sink is v_m . Each path in $\mathcal{P}(u, j)$ consists of edges $e_{i_1}^1, e_{i_2}^2, \dots, e_{i_m}^m$, where for all $1 \leq \ell \leq m$, if $u \in S_\ell$ then $i_\ell = j$, or otherwise $i_\ell = 0$. This completes the description of the construction. Notice that the total budget is $\mathcal{B} = nh2^h$. Let \tilde{m} and \tilde{n} denote the number of customers (i.e. the number of paths) and items, respectively. Notice that $\tilde{m} \leq O(nm \log m)$, and $\tilde{n} \leq nh \leq O(n \log m) \leq O(n^2)$, since we can assume w.l.o.g. that $|\mathcal{S}| \leq 2^n$.

3.2 Analysis

The analysis consists of two parts. First we show that if there is a solution to Unique Coverage that satisfies a β -fraction of the elements, then there is a solution to Tollbooth Pricing of value at least $\beta \cdot \mathcal{B}$. In the second part, we show an efficient randomized algorithm, that, given any solution to Tollbooth Pricing instance G of value $\alpha \cdot \mathcal{B}$, w.h.p. finds a solution to the Unique Coverage problem that satisfies $\Omega(\alpha n)$ elements.

Lemma 11 *If there is a solution to the Unique Coverage instance (U, \mathcal{S}) that satisfies at least βn -elements, then there is a solution to the Tollbooth Pricing instance of value $\beta \mathcal{B}$.*

Proof: Let $\mathcal{S}' \subseteq \mathcal{S}$ be a solution to the Unique Coverage problem, and let $U' \subseteq U$ be the set of elements uniquely covered by \mathcal{S}' , $|U'| \geq \beta n$. For each $S_i \in \mathcal{S}'$, for each $j : 1 \leq j \leq h$, we set the price of the edge e_j^i to 2^j . The prices of all other edges (including the edges e_0^i for all i) are set to 0. For each $u \in U'$ and $j : 1 \leq j \leq h$, we consider the revenue collected from the paths in $\mathcal{P}(u, j)$. Let S_i be the set that uniquely covers u in the solution. Then for each path in $\mathcal{P}(u, j)$, exactly one edge e_j^i on the path has a non-zero price. This price is 2^j - the same as the budget of the path, while all other edges have price 0. Therefore, each such path contributes 2^j to the solution value, and the total contribution of the paths in $\mathcal{P}(u, j)$ is 2^h . This implies the lemma. \square

Lemma 12 *There is an efficient randomized algorithm, that, given any solution to the Tollbooth Pricing instance G of value $\alpha \mathcal{B}$, w.h.p. finds a solution to the Unique Coverage instance (U, \mathcal{S}) that satisfies $\Omega(\alpha n)$ of the elements.*

Proof: Let $p^* : E \rightarrow \mathbb{R}_{\geq 0}$ be any solution of value $\alpha \mathcal{B}$ to the Tollbooth Pricing problem. Let \mathcal{P}_1 be the set of paths, such that each $P \in \mathcal{P}_1$ contributes at least half of its budget to the solution. Our first observation is that the profit collected from the paths in \mathcal{P}_1 must be at least $\alpha \mathcal{B}/2$ (otherwise, we can multiply the price of each edge by a factor of two and get a better solution). From now on, we will only focus on paths in \mathcal{P}_1 and we will discard all other paths. We say that a path $P \in \mathcal{P}_1$ is of type 1 if at least half the cost it pays goes to edges in set $E_0 = \{e_0^i : 1 \leq i \leq m\}$, and it is of type 2 otherwise. Let \mathcal{P}' and \mathcal{P}'' denote the set of paths of type 1 and 2 respectively. We distinguish between two cases.

Case 1: Paths of type 1 contribute at least $\alpha \mathcal{B}/4$ to the solution value. We claim that in this case the solution value is at most $O(\mathcal{B}/\log m)$, and therefore it is sufficient to find a solution to Unique Coverage instance that satisfies a $\Omega(1/\log m)$ -fraction of the elements. We then show an algorithm to find such a solution.

Indeed, consider some element $u \in U$. Recall that, for all j , every path in the sets $\mathcal{P}(u, j)$ traverses all edges in the set $E_0(u) = \{e_0^i : u \notin S_i\}$, and these are the only edges from E_0 traversed by these paths. Let $C_u = \sum_{e \in E_0(u)} p^*(e)$ be the total price of these edges. A path $P \in \mathcal{P}(u, j)$ can belong to \mathcal{P}' only if $2^j/4 \leq C_u \leq 2^j$. This means that there are at most 3 values of $j : 1 \leq j \leq h$ for which $\mathcal{P}(u, j) \cap \mathcal{P}' \neq \emptyset$, so for each $u \in U$, the paths in set $\bigcup_{j=1}^h \mathcal{P}(u, j)$ only contribute at most an $O(1/h) = O(1/\log m)$ -fraction of their total budget to the solution. Therefore, the solution value is at most $O(\mathcal{B}/h) = O(\mathcal{B}/\log m)$. Now we show an algorithm for the Unique Coverage problem

instance that satisfies an $\Omega(1/\log m)$ -fraction of the elements. From Lemma 10, it is enough to construct a fractional solution of value $\Omega(n/\log m)$. For each element $u \in U$, let $\delta(u)$ be the number of sets in \mathcal{S} to which element u belongs. We partition the elements into $h = \lceil \log m \rceil$ classes C_1, \dots, C_h where class C_j contains elements u with $2^j \leq \delta(u) \leq 2^{j+1}$. Let j^* be the class containing the maximum number of elements, so $|C_{j^*}| \geq \Omega(n/\log m)$. We set the weight of every set S to be $w(S) = 1/2^{j^*+1}$. This ensures that all elements in C_{j^*} are fractionally covered. Applying Lemma 10, we obtain an integral solution of value $\Omega(n/\log m)$.

Case 2: Assume now that the paths in \mathcal{P}'' contribute at least $\alpha\mathcal{B}/4$ to the solution value. Let r'' denote the total revenue collected from these paths by edges in $E_1 = E \setminus E_0$. Then we have that $r'' \geq \Omega(\alpha\mathcal{B}) = \Omega(\alpha n h 2^h)$. Notice that by the definition of set \mathcal{P}'' , each path $P \in \mathcal{P}''$ pays at least $1/4$ of its budget for the edges in set E_1 that lie on path P .

We now partition the paths in \mathcal{P}'' into sets $\mathcal{P}''_1, \dots, \mathcal{P}''_h$ where set \mathcal{P}''_j contains all type-2 paths whose budget is 2^j . Let j^* be the index for which the profit contributed by the paths in \mathcal{P}''_{j^*} is maximized. This profit is at least $\alpha n 2^h$.

We say that element u is good if $2^{j^*}/4 \leq \sum_{i:u \in S_i} p^*(e_{j^*}^i) \leq 2^{j^*}$. From the above arguments, for each path $P \in \mathcal{P}''$, if $P \in \mathcal{P}(u, j^*)$, then the corresponding element u must be good. Moreover, if u is good, then all paths in $\mathcal{P}(u, j^*)$ belong to \mathcal{P}''_{j^*} . Therefore, at least $\Omega(\alpha n)$ of the elements in U must be good. We now define a fractional solution to the Unique Coverage problem, where every the weight of every set $S_i \in \mathcal{S}$ is set to $w(S_i) = p(e_{j^*}^i)/2^{j^*}$. Notice that all good elements are fractionally covered, thus giving us a fractional solution where $\Omega(\alpha n)$ elements are satisfied. We finally invoke Lemma 10 to complete the proof. \square

Acknowledgement: The first author thanks Khaled Elbassioni for introducing him to pricing problems and for explaining the differences between various pricing models. He also thanks Danupon Nanongkai and Khaled Elbassioni for useful discussions about UDP_{MIN} and SMP .

References

- [1] Gagan Aggarwal, Tomás Feder, Rajeev Motwani, and An Zhu. Algorithms for Multi-product Pricing. In *ICALP*, pages 72–83, 2004.
- [2] Matthew Andrews, Julia Chuzhoy, Venkatesan Guruswami, Sanjeev Khanna, Kunal Talwar, and Lisa Zhang. Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs. *Combinatorica*, 30(5):485–520, 2010.
- [3] Matthew Andrews and Lisa Zhang. Logarithmic hardness of the undirected edge-disjoint paths problem. *J. ACM*, 53(5):745–761, 2006.
- [4] Maria-Florina Balcan and Avrim Blum. Approximation algorithms and online mechanisms for item pricing. *Theory of Computing*, 3(1):179–195, 2007.
- [5] Patrick Briest. Uniform budgets and the envy-free pricing problem. In *ICALP (1)*, pages 808–819, 2008.

- [6] Patrick Briest and Piotr Krysta. Buying cheap is expensive: hardness of non-parametric multi-product pricing. In *SODA*, pages 716–725, 2007.
- [7] Ning Chen and Xiaotie Deng. Envy-free pricing in multi-item markets. In *ICALP (2)*, pages 418–429, 2010.
- [8] Ning Chen, Arpita Ghosh, and Sergei Vassilvitskii. Optimal envy-free pricing with metric substitutability. *SIAM J. Comput.*, 40(3):623–645, 2011.
- [9] Maurice Cheung and Chaitanya Swamy. Approximation algorithms for single-minded envy-free profit-maximization problems with limited supply. In *FOCS*, pages 35–44. IEEE Computer Society, 2008.
- [10] Erik D. Demaine, Uriel Feige, MohammadTaghi Hajiaghayi, and Mohammad R. Salavatipour. Combination can be hard: Approximability of the unique coverage problem. *SIAM J. Comput.*, 38(4):1464–1483, 2008.
- [11] Khaled M. Elbassioni, Rajiv Raman, Saurabh Ray, and René Sitters. On profit-maximizing pricing for the highway and tollbooth problems. In *SAGT*, pages 275–286, 2009.
- [12] Khaled M. Elbassioni, René Sitters, and Yan Zhang. A quasi-ptas for profit-maximizing pricing on line graphs. In *ESA*, pages 451–462, 2007.
- [13] Uriel Feige. Relations between average case complexity and approximation complexity. In *STOC*, pages 534–543, 2002.
- [14] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43:268–292, March 1996.
- [15] Iftah Gamzu and Danny Segev. A sublogarithmic approximation for highway and tollbooth pricing. In *ICALP (1)*, pages 582–593, 2010.
- [16] Fabrizio Grandoni and Thomas Rothvoß. Pricing on paths: A ptas for the highway problem. In *SODA*, pages 675–684, 2011.
- [17] Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. On profit-maximizing envy-free pricing. In *SODA*, pages 1164–1173, 2005.
- [18] Jason D. Hartline and Vladlen Koltun. Near-optimal pricing in near-linear time. In *In 9th Workshop on Algorithms and Data Structures*, pages 422–431, 2005.
- [19] Rohit Khandekar, Tracy Kimbrel, Konstantin Makarychev, and Maxim Sviridenko. On hardness of pricing items for single-minded bidders. In *APPROX*, pages 202–216, 2009.
- [20] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, STOC '02, pages 767–775, New York, NY, USA, 2002. ACM.
- [21] Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *FOCS*, pages 3–13. IEEE Computer Society, 2000.

- [22] Paat Rusmevichientong. A non-parametric approach to multi-product pricing: Theory and application. *Ph. D. thesis, Stanford University*, 2003.
- [23] Paat Rusmevichientong, Benjamin Van Roy, and Peter W. Glynn. A nonparametric approach to multiproduct pricing. *Oper. Res.*, 54:82–98, January 2006.
- [24] Alex Samorodnitsky and Luca Trevisan. A pcg characterization of np with optimal amortized query complexity. In *STOC*, pages 191–199, 2000.
- [25] Luca Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *STOC*, pages 453–461, 2001.

A Proof of Lemma 10

Let U' be the set of elements that are fractionally covered. We perform randomized rounding of the fractional solution in two stages. In the first stage, each set S is selected independently with probability $\min\{1, 8w(S)\}$. Let \mathcal{S}' be the collection of selected sets. For each element u that is covered by the fractional solution, let $x_u = |\{S \in \mathcal{S}' : u \in S\}|$. We have that $1 \leq \mathbf{E}[x_u] \leq 8$. Moreover, from Markov inequality, $\Pr[x_u \geq 32] \leq 1/4$. We also need to bound the probability that $x_u = 0$. Let $\mathcal{S}_u = \{S \in \mathcal{S} : u \in S\}$. Then $x_u = 0$ can only happen if, for all $S \in \mathcal{S}_u$, $w(S) < 1/8$. In this case, $\mu = \mathbf{E}[x_u] = 8 \sum_{S \in \mathcal{S}_u} w(S) \geq 2$. If we denote by y_S the indicator variable of whether or not S is chosen, then $\mathbf{Var}(x_u) = \sum_{S \in \mathcal{S}_u} \mathbf{Var}(y_S) \leq \sum_{S \in \mathcal{S}_u} \mathbf{E}[y_S] \leq \mu$. Therefore,

$$\Pr[x_u = 0] \leq \Pr[|x_u - \mu| \geq \mu] \leq \frac{\mathbf{Var}(x_u)}{\mu^2} \leq 1/\mu \leq 1/2$$

Let U'' be the subset of elements for which $1 \leq x_u \leq 32$. For $u \in U'$, the probability that $u \in U''$ is at least $1/4$ and thus the expected number of elements in U'' is at least $|U'|/4$. As a second step of the rounding process, each set $S \in \mathcal{S}'$ is randomly selected to be in \mathcal{S}'' with probability $1/32$. The final output is set \mathcal{S}'' . It is easy to see that, for each element $u \in U''$, the probability that u is covered by exactly one set in \mathcal{S}'' is a constant. Thus, the number of elements uniquely covered in the final solution is $\Omega(|U'|)$.

B Proof of Theorem 5

The proof here is almost identical to [25], except that we need to allow super-constant vertex degrees, and slightly change the parameters. We present the proof for completeness. We note that the presentation in this section follows [2, 3].

B.1 Samorodnitsky-Trevisan PCP

The reduction starts from the PCP characterization of NP, due to Samorodnitsky and Trevisan [24]. Let φ be the instance of 3SAT on n variables. For any constant $\ell > 0$, the ST construction gives a PCP verifier that uses $r = O(\log n)$ random bits to generate $q = \ell^2$ locations to probe the proof.

The verifier reads these q bits in the given proof Π and decides whether or not φ is satisfiable. Given a random string r of the verifier, let $b_1(r), \dots, b_q(r)$ be the indices of the proof bits read (we abuse the notation by using r to denote both a random string and the total number of random bits). A configuration is (r, a_1, \dots, a_q) , where $a_1, \dots, a_q \in \{0, 1\}$ are values of $\Pi_{b_1(r)}, \dots, \Pi_{b_q(r)}$. We say that a configuration (r, a_1, \dots, a_q) is accepting if, for a random string r of the verifier and the values a_1, \dots, a_q of the corresponding proof bits, the verifier accepts. The properties of the PCP are summarized in the following theorem.

Theorem 13 ([24]) *For any constant $\mu > 0, \ell > 0$, there is a PCP verifier for 3SAT with the following properties:*

- *If φ is a YES-INSTANCE, accepts with probability at least $1 - \mu$, for some proof Π .*
- *If φ is a NO-INSTANCE, accepts with probability at most $\mu + 2^{-\ell^2}$, for any proof Π .*
- *Read $2\ell + \ell^2$ query bits and tosses $r = O(\log n)$ random coins.*
- *For each random string r , there are $2^{2\ell}$ accepting configurations.*
- *For each random string r , for every $j : 1 \leq j \leq q$, the number of accepting configurations where the value of $\Pi_{b_j(r)} = 0$ equals the number of accepting configurations where $\Pi_{b_j(r)} = 1$.*

We fix the value $\mu = 2^{-\ell^2}$ and ℓ to be a large enough constant. Therefore, in the YES-INSTANCE, we have the acceptance probability of at least $1/2$, while the acceptance probability in the NO-INSTANCE is at most $2 \cdot 2^{-\ell^2}$.

Observe that the length of the proof is bounded by $2^r(2\ell + \ell^2)$. For convenience, we would like to ensure that each proof bit participates in many accepting configurations, and this can be done as follows. The verifier works as before, except that now it also randomly chooses one additional proof bit and accepts if and only if the original verifier accepts. Observe now that the length of the random string becomes $r' \leq r + r + 3 \log \ell$, and the number of query bits read is $\ell^2 + 2\ell + 1$. Let Π_j be some proof bit and n_j be the number of accepting configurations in which the value of Π_j is zero. We have that $n_j \geq 2^r 2^{2\ell-1} \geq 2^{r'/2}$. We summarize the properties of the PCP construction below (after substituting the values of r and ℓ by the new values r' and $\ell' = \ell + 1$).

- The verifier reads at most ℓ^2 query bits and tosses $r = O(\log n)$ random coins.
- (YES-INSTANCE:) The acceptance probability is at least $1/2$.
- (NO-INSTANCE:) The acceptance probability is at most $2^{-\ell^2/2}$.
- For each random string, there are $2^{2\ell-1}$ accepting configurations.
- For each random string r , for each $j : 1 \leq j \leq q$, the number of accepting configurations where the value of $\Pi_{b_j(r)} = 1$ equals the number of accepting configurations where the value of $\Pi_{b_j(r)} = 0$.
- For each proof index j , we have $n_j \geq 2^{r/2}$

We repeat this protocol independently λ times, and the verifier accepts if and only if the original verifier accepts in each protocol repetition. The resulting PCP has the following properties:

- **Random Bits:** λr . Let \mathcal{R} denote the set of all possible random strings, $|\mathcal{R}| = 2^{\lambda r}$.
- **Query Bits:** $q = \lambda \ell^2$. Assume w.l.o.g. that the verifier reads exactly q bits of proof for every random string.
- **Completeness:** YES-INSTANCE is accepted with probability at least $2^{-\lambda}$.
- **Soundness:** NO-INSTANCE is accepted with probability at most $2^{-\lambda \ell^2/2}$.
- For each random string, there are $2^{\lambda(2\ell-1)}$ accepting configurations.
- For each proof index j , let Z_j be the set of all accepting configurations in which proof bit Π_j participates with value 0 and O_j be the set of all accepting configurations in which Π_j participates with value 1. We denote $n_j = |Z_j| = |O_j| \geq 2^{\lambda r/2}$.

B.2 Construction and Analysis

Following [25, 3], we perform the reduction of Feige et al. [14], to construct the graph $G = (V, E)$. For each random string $r \in \mathcal{R}$, for each accepting configuration $C = (r, a_1, \dots, a_q)$, we add a vertex $v(r, C)$ to G , representing this configuration. Therefore, the number of the graph vertices, $N = |V(G)| = |\mathcal{R}| 2^{\lambda(2\ell-1)} = 2^{\lambda(r+2\ell-1)}$. For each index j in the proof Π , we let O_j and Z_j be the subsets of vertices of G representing the configurations where the value of the bit Π_j is 1 and 0 respectively. We add a set E_j of edges between the vertices of O_j and the vertices of Z_j , using the following lemma, due to Andrews and Zhang [3]:

Lemma 14 (Lemma 11 in [3]) *For all $\gamma > 0$ and sufficiently large n' , we construct a bipartite graph $([n'], [n'], E')$ where the edges in E' are obtained by taking $\delta = \frac{3}{\gamma} \log \frac{1}{\gamma}$ random bipartite matchings. We say that the algorithm succeeds if for all subsets $A, B \subseteq [n']$, $|A|, |B| \geq \gamma n'$, there is an edge $(i, j) \in E'$ connecting $i \in A$ and $j \in B$. Then the above algorithm succeeds with probability at least $1 - e^{-n'\gamma(\log(1/\gamma)-2)}$.*

For each proof index j , we invoke the above lemma to construct the set of edges E_j connecting O_j and Z_j . The final set of edges is $E = \bigcup_j E_j$. This completes the construction of graph $G = (V, E)$. We first argue that the algorithm succeeds with high probability. Recall that $n_j \geq 2^{\lambda r/2} \geq n^{\Omega(\lambda)}$. Our choice of γ will be $2^{-O(\lambda)}$ (to be made precise later), so the lemma only fails with exponentially small probability for each proof bit Π_j . Since there is a polynomial number of proof bits, we can apply the union bound to ensure that the algorithm succeeds for all proof bits with high probability. Notice that, each vertex has degree at most $O(\frac{q}{\gamma} \log(1/\gamma))$, because for each random string r , only q proof bits are read.

Claim 15 *In the YES-INSTANCE, there is an independent set of size at least $2^{\lambda(r-1)}$.*

Proof: Fix the proof Π^* that is optimal in the YES-INSTANCE. We say that a random string $r \in \mathcal{R}$ is good if it causes the verifier to accept given the proof Π . Recall that the number of good random

strings is at least $|\mathcal{R}|2^{-\lambda} = 2^{\lambda(r-1)}$. For each good random string $r \in \mathcal{R}$, we can add, to set S , one vertex $v(r, C)$ where configuration $C = (r, \Pi_{b_1(r)}^*, \dots, \Pi_{b_q(r)}^*)$ encodes the corresponding answers. So we have $|S| \geq 2^{\lambda(r-1)}$. Since the edges in E only connect conflicting configurations, set S must be an independent set. \square

Claim 16 *In the NO-INSTANCE, any independent set has size at most $2^{\lambda(r-\ell^2/2)} + q\gamma N$.*

Proof: Assume that $S \subseteq V$ is an independent set of size larger than $2^{\lambda(r-\ell^2/2)} + q\gamma N$. For each j , we know that $\min\{|S \cap Z_j|, |S \cap O_j|\} \leq \gamma n_j$ because otherwise (from Lemma 14) there would have been an edge between $S \cap Z_j$ and $S \cap O_j$. If $|S \cap Z_j| < |S \cap O_j|$, we remove vertices in $S \cap Z_j$ from S and define the value $\Pi'_j = 1$; otherwise, we remove vertices in $S \cap O_j$ from S and define $\Pi'_j = 0$. Once we have processed all the proof bits, we have removed at most $\sum_j \gamma n_j \leq q\gamma N$ vertices from set S , and have obtained the proof Π' . The resulting set \tilde{S} satisfies $|\tilde{S}| > 2^{\lambda(r-\ell^2/2)}$. We now define set $\mathcal{R}' = \{r \in \mathcal{R} : (\exists C)v(r, C) \in \tilde{S}\}$, so we have $|\mathcal{R}'| \geq |\tilde{S}| > 2^{\lambda(r-\ell^2/2)}$. Notice that the verifier accepts proof Π' whenever $r \in \mathcal{R}'$. Hence the verifier accepts proof Π' with probability $|\mathcal{R}'|/|\mathcal{R}| > 2^{-\ell^2/2}$, a contradiction. \square

Finally we analyze the parameters. We set $\ell = 8/\epsilon$, where ϵ is the parameter from the statement of Theorem 5. Recall that $N = 2^{\lambda(r+2\ell-1)} = n^{\lambda \text{poly}(\frac{1}{\epsilon})}$. We choose $\gamma = 2^{\lambda(r-\ell^2/2)}/qN = 2^{\lambda(-\ell^2/2-2\ell)}/q$, so we have $q\gamma N = 2^{\lambda(r-\ell^2/2)}$. We write all parameters in terms of ϵ and λ below:

- **Degree:** Recall that the maximum vertex degree in graph G is $\Delta = O(\frac{q}{\gamma} \log(1/\gamma))$. By plugging in the values $\gamma = \frac{2^{\lambda(-\ell^2/2-2\ell)}}{q}$ and $q = \lambda\ell^2$, we have $\Delta = O(\lambda^3\ell^6) \cdot 2^{\lambda\ell/2+2\lambda\ell-1}$, so for a large enough λ , $2^{\lambda(\ell^2/2+\ell)} \leq \Delta \leq 2^{\lambda(\ell^2/2+2\ell)}$.
- **Completeness:** The value of YES-INSTANCE is $2^{\lambda(r-1)} = N/2^{2\lambda\ell}$. From the choice of $\ell = 8/\epsilon$, we have $\Delta^\epsilon \geq 2^{\epsilon\lambda(\ell^2/2+2\ell)} \geq 2^{\lambda(2\ell+1)}$, so completeness parameter is at least $N/2^{2\lambda\ell} \geq N/\Delta^\epsilon$.
- **Soundness:** The value of the NO-INSTANCE is at most $2 \cdot 2^{\lambda(r-\ell^2)} \leq N/2^{\lambda(\ell^2/2+2\ell+1)}$. Since $\Delta^{1-\epsilon} \leq 2^{\lambda(\ell^2/2+2\ell)(1-8/\ell)} \leq 2^{\lambda\ell^2/2}$, the soundness value is at most $N/\Delta^{1-\epsilon}$.
- **Running time:** Creating the vertex set in FGLSS graph can be done in time $2^{\lambda(r+2\ell-1)} = n^{\lambda \text{poly}(\frac{1}{\epsilon})}$. Adding edges into the graph can be done in time polynomial in N , so the total running time is $n^{\lambda \text{poly}(\frac{1}{\epsilon})}$.

Derandomization: We can make this reduction deterministic by replacing Lemma 14 with the following deterministic counterpart.

Lemma 17 ([21], as stated in Lemma 8 in [25]) *For any γ and sufficiently large n , there is a bipartite graph $([n'], [n'], E)$ of degree $O(\frac{1}{\gamma} \text{poly} \log(1/\gamma))$ such that, for each subsets $A, B \subseteq [n']$, $|A|, |B| \geq \gamma n'$, there is an edge $(i, j) \in E$ connecting $i \in A$ and $j \in B$. Such a graph can be constructed in time $\text{poly}(n', 2^{1/\gamma})$.*

Using this lemma will cause the running time of $2^{1/\gamma} \text{poly } n = 2^{O(\Delta)} \text{poly } n$.

C Hardness of SMP

The construction here is very similar to the hardness construction of UDP_{MIN} , except that here the budgets of virtual customers are defined differently. For SMP , each virtual customer $c \in \mathcal{C}_h$ has $B_c = |\mathcal{C}|^{-2h}$, and $\mathcal{G}(c) = \{c(1), \dots, c(|\mathcal{C}|^{2h})\}$. For completeness, we describe the construction and its analysis below.

C.1 Construction

Let $G = (V, E)$ be the instance of **Maximum Independent Set** obtained from Theorem 5, where the value of λ (and Δ) will be fixed later. We first define an intermediate instance of SMP , which is then converted into a final instance.

The intermediate instance is defined as follows. The set of items contains, for each vertex $v \in V$, for each index $y \in [\Delta]$, an item $i(v, y)$. That is, $\mathcal{I} = \{i(v, y) \mid v \in V, y \in [\Delta]\}$.

Similarly, the set of customers contains, for each vertex $v \in V$, for each index $x \in [\Delta]$, a customer $c(v, x)$. That is, $\mathcal{C} = \{c(v, x) \mid v \in V, x \in [\Delta]\}$.

The item set $S_{c(v, x)}$ for the customer $c(v, x)$, contains the item $i(v, x)$, and additionally, for each neighbor u of vertex v in graph G , for each index $y \in [\Delta]$, item $i(u, y)$ belongs to $S_{c(v, x)}$. Formally, $S_{c(v, x)} = \{i(u, y) \mid (u, v) \in E, y \in [\Delta]\} \cup \{i(v, x)\}$. Notice that $|S_{c(v, x)}| \leq \Delta^2 + 1$ for all customers $c(v, x) \in \mathcal{C}$. Moreover for each item $i(v, y) \in \mathcal{I}$, there are at most $\Delta^2 + 1$ customers $c' \in \mathcal{C}$ such that $i(v, y) \in S_{c'}$.

We partition the set \mathcal{C} of customers into Δ subsets $\mathcal{C}_1, \dots, \mathcal{C}_\Delta$, such that for each $1 \leq h \leq \Delta$, set \mathcal{C}_h contains customers $c(v, h)$ for all $v \in V$. Finally, for each $1 \leq h \leq \Delta$, each customer $c \in \mathcal{C}_h$ is assigned budget $B_c = 1/|\mathcal{C}|^{2h}$.

This finishes the definition of the intermediate instance. For convenience, we call the customers in set \mathcal{C} *virtual customers*. In our final instance, we replace each virtual customer with a number of new customers.

In order to define the final instance, for each $1 \leq h \leq \Delta$, we replace each virtual customer $c \in \mathcal{C}_h$ with a set $\mathcal{G}(c) = \{c(1), \dots, c(|\mathcal{C}|^{2h})\}$ of $|\mathcal{C}|^{2h}$ identical new customers. Each new customer $c(h')$, for $1 \leq h' \leq |\mathcal{C}|^{2h}$ has budget $B_{c(h')} = B_c$ and $S_{c(h')} = S_c$. The final set of customers is $\mathcal{C}' = \bigcup_{c \in \mathcal{C}} \mathcal{G}(c)$ and the final set of items remains unchanged, $\mathcal{I}' = \mathcal{I}$. The number of customers in the final instance is $\tilde{m} = |\mathcal{C}'| = O(|\mathcal{C}|^{O(\Delta)} |\mathcal{C}|) = |V| \cdot \Delta \cdot |\mathcal{C}|^{O(\Delta)} = |V|^{O(\Delta)}$, while the number of items is $\tilde{n} = |V| \cdot \Delta$. Moreover, for each customer $c \in \mathcal{C}'$, we have $|S_c| \leq \Delta^2 + 1$.

C.2 Analysis

The analysis is very similar to the analysis of UDP_{MIN} .

Lemma 18 *In the YES-INSTANCE, there is a price function that collects revenue at least $|V| \Delta^{1-\epsilon}$.*

Proof: Let $U \subseteq V$ be a maximum independent set of size $|V|/\Delta^\epsilon$. For each item $i(u, y)$, $u \in U, y \in [\Delta]$, we set the price $p(i(u, y)) = B_{c(u, y)} = |\mathcal{C}|^{-2y}$. We set the prices of other items to 0. Now we

argue that the total revenue made by price function p is at least $|V|\Delta^{1-\epsilon}$.

Fix $v \in U, y \in [\Delta]$. Consider virtual customer $c' = c(v, y) \in \mathcal{C}_y$. Notice that $S_{c'}$ contains item $i(v, y)$ but not any other item of positive price. Therefore each customer in $\mathcal{G}(c')$ buys the set $S_{c'}$, and we can collect the total revenue of 1 from customers in $\mathcal{G}(c')$. The total profit collected from customers in \mathcal{C}' is at least $|U|\Delta \geq |V|\Delta^{1-\epsilon}$. \square

Lemma 19 *In the NO-INSTANCE, the value of the optimal solution is at most $O(|V| \cdot \Delta^\epsilon)$.*

Proof: Let p^* be the optimal solution and r^* be its revenue. We say that a virtual customer $c \in \mathcal{C}$ is *good* if and only if $B_c \geq \sum_{i \in S_c} p^*(i) \geq \frac{B_c}{2|\mathcal{C}|}$. Denote by $\mathcal{C}^* \subseteq \mathcal{C}$ the set of good virtual customers.

We first claim that $|\mathcal{C}^*| \geq r^*/2$. Indeed, for each bad virtual customer $c \in \mathcal{C} \setminus \mathcal{C}^*$, customers in $\mathcal{G}(c)$ altogether contribute at most $1/(2|\mathcal{C}|)$ to the profit, and therefore the total revenue from customers in $\bigcup_{c' \in \mathcal{C} \setminus \mathcal{C}^*} \mathcal{G}(c')$ is at most $1/2 \leq r^*/2$. Hence the revenue of at least $r^*/2$ must come from customers in $\bigcup_{c' \in \mathcal{C}^*} \mathcal{G}(c')$, and so $|\mathcal{C}^*| \geq r^*/2$.

From now on, we focus on constructing an independent set U in graph G of size at least $(r^*/2 - |V|)/\Delta$ from \mathcal{C}^* . Since in the NO-INSTANCE, G does not contain an independent set of size more than $|V|/\Delta^{1-\epsilon}$, this implies that $(r^*/2 - |V|)/\Delta \leq |V|/\Delta^{1-\epsilon}$, and hence $r^* \leq O(|V| \Delta^\epsilon)$.

We construct an independent set $U \subseteq V(G)$, together with a partition $(\mathcal{C}^1, \mathcal{C}^2)$ of \mathcal{C}^* , as follows. Start with $U, \mathcal{C}^1, \mathcal{C}^2 = \emptyset$. We then perform Δ iterations where, in iteration h , we consider all virtual customers in $\mathcal{C}_{\Delta-h+1}$. When customer $c(v, y)$ in $\mathcal{C}^* \cap \mathcal{C}_y$ is considered, we do the following:

- If vertex v is already in U , we add virtual customer $c(v, y)$ into \mathcal{C}^1 .
- If vertex v is not in U and $U \cup \{v\}$ remains an independent set, we add vertex v to set U and add $c(v, y)$ to \mathcal{C}^1 . We say that $c(v, y)$ is *responsible* for adding vertex v into U .
- Otherwise, $v \notin U$, but there is a vertex $u \in U$ such that $(u, v) \in E(G)$. We add $c(v, y)$ to \mathcal{C}^2 in this case and say that vertex u prevents the algorithm from adding v into U .

In the end, when all customers in \mathcal{C}^* are processed, each virtual customer in \mathcal{C}^* is added to either \mathcal{C}^1 or \mathcal{C}^2 , so $\mathcal{C}^* = \mathcal{C}^1 \cup \mathcal{C}^2$. Moreover, for each virtual customer $c(v, y)$ in \mathcal{C}^1 , the corresponding vertex v belongs to U , so $|U| \geq |\mathcal{C}^1|/\Delta$. The following claim will complete the proof of the lemma.

Claim 20 $|\mathcal{C}^2| \leq |V|$, and so $|U| \geq |\mathcal{C}^* \setminus \mathcal{C}^2|/\Delta \geq (r^*/2 - |V|)/\Delta$.

Proof: It is sufficient to show that for each vertex $v \in V$, no two virtual customers $c(v, y), c(v, y')$ with $y \neq y'$ belong to \mathcal{C}^2 . Assume otherwise and let $c(v, y), c(v, y') \in \mathcal{C}^2$. By our construction, we have $c(v, y) \in \mathcal{C}_y$ and $c(v, y') \in \mathcal{C}_{y'}$. Assume w.l.o.g. that $y > y'$, so $c(v, y)$ was processed before $c(v, y')$.

Let $u \in U$ be a vertex such that $(u, v) \in E(G)$, and vertex u prevents the algorithm from adding v to set U . Let $c(u, x)$ be the customer responsible for adding vertex u to U . Then $c(u, x)$ was processed in iteration $x : x \geq y > y'$.

Notice that the item $i(v, y')$ belongs to $S_{c(u, x)}$. The price of item $i(v, y')$ must be at most $B_{c(u, x)} \leq B_{c(v, y')}/|\mathcal{C}|^2$. Now consider virtual customer $c(v, y')$. Since it is a good customer and $|S_{c(v, y')}| \leq 2\Delta^2$,

some item in $S_{c(v,y')}$ must cost at least $B_{c(v,y')}/(2\Delta^2|\mathcal{C}|)$, and this item cannot be $i(v, y')$. Let i' be such an item and assume that $i' = i(w, z)$. Then w is a neighbor of v in G , $w \neq v$, and so $i' \in S_{c(v,y)}$ must hold. Since the price of i' is $p^*(i') \geq B_{c(v,y')}/(2\Delta^2|\mathcal{C}|) > B_{c(v,y)}$, customers in $\mathcal{G}(c(v, y))$ buy nothing in this solution, contradicting the fact that $c(v, y) \in \mathcal{C}^*$. \square

\square

As in the analysis presented in Section 2, the gap that we obtain is $g = \Delta^{1-2\epsilon}$, while the sizes of the sets S_c are bounded by $\Delta^2 + 1$, and the number of items is $\tilde{n} = |V| \cdot \Delta$. The number of customers is slightly higher, $\tilde{m} = |V|^{O(\Delta)}$. It is still immediate to verify that the same analysis of the parameters as in Section 2 works for this construction, giving the required hardness of approximation factors.