

Inequity Aversion Pricing over Social Networks: Approximation Algorithms and Hardness Results*

Georgios Amanatidis^{a,b}, Peter Fulla^c, Evangelos Markakis^d, Krzysztof Sornat^{e,f}

^aUniversity of Essex, Department of Mathematical Sciences; United Kingdom

^bUniversity of Amsterdam, Institute for Logic, Language and Computation; The Netherlands

^cUniversity of Oxford, Department of Computer Science; United Kingdom

^dAthens University of Economics and Business, Department of Informatics; Greece

^eBen-Gurion University, Department of Industrial Engineering and Management; Israel

^fUniversity of Wrocław, Institute of Computer Science; Poland

Abstract

We study a revenue maximization problem in the context of social networks. Namely, we generalize a model introduced by Alon, Mansour, and Tennenholtz [2] that captures *inequity aversion*, i.e., it captures the fact that prices offered to neighboring vertices should not differ significantly. We first provide approximation algorithms for a natural class of instances, where the total revenue is the sum of *single-value revenue functions*. Our results improve on the current state of the art, especially when the number of distinct prices is small. This applies, for instance, to settings where the seller will only consider a fixed number of discount types or special offers. To complement our positive results, we resolve one of the open questions posed in [2] by establishing APX-hardness for the problem. Surprisingly, we further show that the problem is NP-complete even when the price differences are allowed to be large, or even when the number of allowed distinct prices is as small as three. Finally, we study extensions of the model regarding the demand type of the clients.

Keywords: revenue maximization, inequity aversion, social networks, pricing, approximation algorithms

*A preliminary conference version of this work appeared in MFCS 2016 [3].

Email addresses: g.d.amanatidis@gmail.com (Georgios Amanatidis), peter.fulla@gmail.com (Peter Fulla), markakis@gmail.com (Evangelos Markakis), krzysztof.sornat@cs.uni.wroc.pl (Krzysztof Sornat)

Published in Theoretical Computer Science, DOI:10.1016/j.tcs.2021.04.012

1. Introduction

We study a differential pricing optimization problem in the presence of network effects. Differential pricing is very common in practice and refers to offering different prices to different potential customers for the same service or good. Examples include offering cheaper prices when launching a new product, making special offers to gold and silver members of an airline miles program, and offering discounts at stores during selected periods.

Imagine a network connecting individuals—who are seen as potential clients here—with their friends, family, colleagues or other people who can exert some influence on them. Of course, one may also have in mind other forms of abstract networks, e.g., a node could represent a geographic region, a neighborhood within a city, a type of profession, a social class, and edges can represent interactions or proximity. The presence of such a network creates *externality effects*, meaning that the decision of a node to acquire a new product or a new service is affected by the fact that some other nodes within her social circle (i.e., her neighborhood in the graph) already did so. A typical example of positive externalities is when someone becomes more likely to buy a new product due to the positive reviews by a friend who already bought it in the past. Modeling positive externalities has led to a series of works that study marketing strategies for maximizing the diffusion of a new product [12, 22] or the total revenue achieved [21]; see also the Related Work section.

However, there also exist negative externality effects that can arise in a network. One example is the purchase of products with the intention to show off and be a locally unique owner, e.g., a new expensive car model or designer clothes (also referred to as *invidious consumption* [8]). In such a case, a node may be deterred from buying the same product, if a neighboring node already did so. A second example of negative externalities, which arises from differential pricing and is the focus of this work, is *inequity aversion* [7, 13]. The idea behind inequity aversion is that a customer may be discontent if she realizes that other people within her social circle were offered a better deal for the same service. Hence, significant price differences can create a negative response by some customers towards the target product. Inequity aversion can also arise under a different, but equally applicable, interpretation: nodes may correspond to retail stores and an edge can signify proximity, so that clients could choose among these stores. Again, having significantly different prices to the same products is often not desirable.

To capture the need for avoiding such phenomena, the work of Alon, Mansour, and Tennenholtz [2] introduced a model for pricing nodes over a social network. The main idea is to impose constraints on each edge, specifying that the price difference between two neighbors should be bounded by some endogenous parameter determined by the two neighbors. On top of this, the seller is also allowed to not make a price offer to some nodes (referred to as introducing *discontinuities*, see the definition in Section 2), in which case the difference constraints do not apply for the edges incident to these nodes.

The notion of discontinuities can also be interpreted as follows: Suppose we had no discontinuities but at the same time we were allowed to make offers that violate the price constraints along some edges. In that case, consider a node v affected by such a violation, i.e., a node who has been offered a price she considers to be too high given the prices offered to her neighbors. As v would feel envious of some neighbor, she would choose not to buy the product. Hence, the seller does not hope to extract any revenue from

nodes like v . This is equivalent to just setting a discontinuity on v and only caring for the edge constraints not involving v . For convenience, we will stick to the terminology of [2], with allowing discontinuities, rather than making price offers that violate the constraints. Occasionally, in related works, e.g., [6], the equivalent notion of *buyer preselection* is used instead. That is, the seller has the freedom to first remove a subset of potential buyers (these are the nodes where the discontinuities would be introduced here) and then solve the pricing problem without discontinuities on the remaining network.

Assuming a finite set of available prices, unit-demand users, and digital goods (i.e., the supply can cover all the demand) the problem is to find a feasible price vector that satisfies the edge constraints and maximizes the total revenue. For general revenue functions the problem was shown to be NP-complete by Alon et al. [2], and exact or approximation algorithms were derived for some special cases. However, several questions remained open regarding the approximability status of the problem.

Contribution. We revisit the model introduced by Alon et al. [2] (namely Model II in their work, which is the most general therein) and study the approximability of the underlying revenue maximization problem. In fact, we generalize the model to allow prices from a set of k arbitrary integers, rather than using price sets of the form $\{1, 2, \dots, k\}$ as done in [2]. Essentially all our positive results hold for this general model. We focus on the natural class of the so-called *single-value revenue functions*. Simply put, this means that the revenue extracted by each node is exactly the price offered to her, as long as the price does not exceed her valuation for the product (i.e., the usual assumption made in almost all other pricing problems as well).

We start in Section 4 with exact algorithms, and we completely resolve the case of two distinct prices by showing that there is a polynomial-time exact algorithm (Theorem 4.1). As exhibited later by the results of Section 6, this is the only case where we can have an efficient exact algorithm. We therefore continue in Section 5 with designing approximation algorithms. We first provide an improved approximation algorithm for the case where the price set is $\{1, 2, \dots, k\}$ (Theorem 5.1). Then, in Section 5.2, we extend these results to arbitrary price sets (Theorem 5.3). The improvement we get is stronger when the number of distinct prices is small. This applies, for example, to many settings where the seller will only consider a fixed number of discount types or special offers to selected customers. As the number of customers and the number of available prices both become large, the performance of our algorithms degrades to a logarithmic approximation.

Regarding the problem’s complexity, we resolve one of the main open questions posed in [2] about single-value revenue functions. We establish APX-hardness for this case (Theorem 6.7), and we further show that the problem remains hard even for cases that could be thought easier to handle. In particular, it is NP-complete even when the price differences are allowed to be almost as large as the maximum price (Theorem 6.1), or when we only have three prices allowed (Theorem 6.11). The latter strikes a contrast to the polynomial-time solvable case of two distinct prices.

Finally, in Section 7, we consider the generalization of the problem to a multi-unit demand setting. Surprisingly, we show that all positive results from the single-unit setting hold here, as long as the demands are polynomially bounded (Theorem 7.1).

Remark 1.1. In comparison to the preliminary conference version of this work [3], there are two main additional contributions which resolve the problem’s complexity for a small number of prices, namely Theorems 4.1 and 6.11. That is, the NP-hardness for three prices

(previously left as an open question) and the polynomial-time algorithm for two prices (previously we provided a 0.8-approximation algorithm for this case). This algorithm is then used as a subroutine to improve the approximation ratio for general price sets in Section 5.2. Additionally, here we include proofs omitted from the conference version. In particular, we provide a full proof of the APX-hardness for single-value revenue functions.

Related Work. Price discrimination is well studied in various domains in economics and is also being applied to numerous real life scenarios. The algorithmic problem of differential pricing over social networks is a more recent topic, initiated by Hartline et al. [21]. This work studied a model with positive externalities, where the valuation of an agent may increase as more friends acquire a good, and analyzed the performance of a very intuitive class of pricing strategies. Further improvements on the performance of such strategies were obtained later on by Fotakis and Siminelakis [16]. Akhlaghpour et al. [1] also consider a pricing problem but in an iterative fashion, where the seller is allowed to reprice a good in future rounds. Revenue maximization under a mechanism design approach was taken by Haghpanah et al. [20] under positive network externalities. Finally, positive externalities have been used to model the diffusion of products on a network, see, e.g., the exposition by Kleinberg [24].

Negative externalities within networks, as we focus on here, are less studied in the literature. For the concept of inequity aversion, see the work of Bolton and Ockenfels [7] and Fehr and Schmidt [13]. Most closely related to ours is the work of Alon et al. [2], who introduced the model considered here. Efficient algorithms were obtained for the case where discontinuities are not allowed (even for more general revenue functions), and also for networks with bounded treewidth. An approximation ratio of $1/(\Delta + 1)$ was also provided, where Δ is the maximum degree. Similar results were shown for a stochastic version of the model. Finally, other types of negative externalities have been considered e.g., by Bhattacharya et al. [5] and Cao et al. [8], who study the effects of invidious consumption.

There are also some further recent results on price discrimination models related to ours, in the sense that they consider the presence of a social network or they allow buyer preselection (i.e., excluding some buyers from being eligible to receive a price offer). The work of Bilò et al. [6] studies pricing problems with buyer preselection but without the presence of a social network. They consider limited supply of multiple goods and general valuation functions, while we consider a single type of good with unlimited supply and unit-demand users. Flammini et al. [14] study a multi-unit demand model with limited supply, similar to the model in [6], but assuming a directed social network as we do. One of the main differences to our model is not allowing buyer preselection. Furthermore, Flammini et al. [14] look for item envy-free outcomes with fair pricing (where fairness here prohibits offering lower prices to the neighbors). This is a more stringent model in terms of the price differences between neighbors; in contrast, we allow envy (and hence more flexibility in pricing), up to given model parameters. Finally, Flammini et al. [15] study a similar model to [14] (multi-unit demand with limited supply, presence of social network, no buyer preselection) but instead of using the social network to define fair pricing, they use it to determine the subset of agents with respect to whom an agent should be bundle envy-free.

2. Preliminaries

We assume there is a social network that is represented as an undirected graph $G = (V, E)$ with $|V| = n$. The nodes are the potential customers, and there is also a provider of some good or service, who has a finite set P of available prices that she could offer to the nodes. The price set $P = \{p_1, p_2, \dots, p_k\}$ can be any set of k distinct positive integers (in increasing order). Note that the original model of Alon et al. [2] assumes that the set of available prices is $P = [k] = \{1, 2, \dots, k\}$. All our positive results (except for Theorem 5.1 which is nevertheless generalized in Section 5.2) are true for general prices, while our hardness results (in particular, Theorem 6.11) hold even for the price set $P = [k]$.

We further assume that every node has a unit-demand for the same product and that the supply of the seller is enough to cover the demand of all nodes. For every node $v \in V$, we associate a revenue function $R_v : P \rightarrow \mathbb{N}$, that maps an offered price p_v to the revenue that the provider gains from this offer. In this work, we focus on a simple and intuitive class of revenue functions, also studied by Alon et al. [2]. In particular, for a node $v \in V$, R_v is called a *single-value revenue function*, if there exists a value $\text{val}(v)$ such that when offered a price p_v :

$$R_v(p_v) = \begin{cases} p_v, & \text{if } \text{val}(v) \geq p_v; \\ 0, & \text{if } \text{val}(v) < p_v. \end{cases}$$

We assume that every node has a single-value revenue function. We can also assume, without loss of generality, that $\text{val}(v) \in P$, for every $v \in V$. To see this, note that for revenue maximization (which is the objective here) nodes with $\text{val}(v) > p_k$ can only yield a revenue equal to some element of P and their value could be replaced by $\text{val}(v) = p_k$, i.e., the highest possible price. From values that are between p_i and p_{i+1} , for some $i \in [k-1]$, we can again extract only revenue equal to some element of $\{p_1, \dots, p_i\}$, so we could round them down to p_i . Finally, any node v with $\text{val}(v) < p_1$ can be deleted without affecting the optimal revenue (see the concept of *discontinuity* defined below), so we can completely ignore such nodes to begin with. Therefore, we consider only instances with $\text{val}(v) \in P$, for all $v \in V$.

Given a vector $\mathbf{p} = (p_v)_{v \in V}$ of prices offered to the nodes, the total revenue is $R(\mathbf{p}) = \sum_{v \in V} R_v(p_v)$. Our goal is to find a price vector that maximizes the total revenue. At the same time, however, we want to capture the effect of *inequity aversion* [7, 13] in social networks, namely the fact that significant price differences create negative externalities among users. To avoid such price differences the model of Alon et al. [2] introduces a constraint on each edge stating that the price difference between two neighbors u, v is bounded. That is, $p_u - p_v \leq \alpha(u, v)$ and $p_v - p_u \leq \alpha(v, u)$, for every $(u, v) \in E$. Here, $\alpha(\cdot, \cdot) \geq 0$ is integer-valued (given that the prices are also integers), and it can be non-symmetric. Furthermore, the seller is allowed not to make an offer to certain nodes. Formally, this is captured by having one more price option, which we denote by \perp , with $R_v(\perp) = 0$. Setting $p_v = \perp$ to a node means that the provider does not make any offer to v (or alternatively, makes a very high offer to v that cannot be accepted) and there is no price restriction on the edges that are incident to v . We can essentially think about this as deleting these vertices from the graph. We will refer to setting $p_v = \perp$ to a node $v \in V$ as *introducing a discontinuity on v* . Avoiding making an offer can be thought of as choosing not to promote a product or service within a certain

region or within a certain social group. In terms of optimization, discontinuities can help the seller produce much higher revenue as Proposition 3.3 states.

Given this model, the set of feasible price vectors is $\mathcal{F} = \{\mathbf{p} : \forall v \in V, p_v \in P \cup \{\perp\}, \text{ and } \forall (u, v) \in E, p_u \neq \perp \wedge p_v \neq \perp \Rightarrow p_u - p_v \leq \alpha(u, v) \wedge p_v - p_u \leq \alpha(v, u)\}$. Therefore, the problem we study is:

Inequity Aversion Pricing: Given a graph G with edge constraints, a set P of available prices, and a single-value revenue function R_v for each node v , find a feasible price vector \mathbf{q} that maximizes the total revenue. That is, find $\mathbf{q} \in \arg \max_{\mathbf{p} \in \mathcal{F}} \sum_{v \in V} R_v(p_v)$, where \mathcal{F} is defined as above.

Some cases of this problem, as well as the variant where no discontinuities are allowed, are already known to be polynomial time solvable [2]. Regarding hardness, although the problem is NP-hard for general revenue functions, it was posed as an open question whether NP-hardness still holds for single-value revenue functions.

3. Warmup: Basic Facts and Single-Price Solutions

In this section, we present a simple algorithm and some basic observations which we use later on, in Section 5.

Let $\mu = \max_{v \in V} \text{val}(v)$ be the maximum value any potential customer has for the good, and $\text{VAL} = \sum_{v \in V} \text{val}(v)$ be the total value for it. Given an instance of the problem, we denote by OPT the revenue of an optimal solution. The quantity VAL is clearly an upper bound on the optimal revenue, i.e., $\text{OPT} \leq \text{VAL}$.

We will refer to a solution as being a *single-price* solution, if it charges the same price to every node without introducing discontinuities. This is always a feasible solution since all the edge constraints are satisfied. The revenue extracted by the single-price solution that uses the common price p is equal to $p \cdot |\{v \in V : \text{val}(v) \geq p\}|$.

To understand whether a single-price solution can be of any help for our setting, we may examine the performance of the *best* possible single price. The following observation suggests that we do not need to try too many values, even if μ is very large.

Lemma 3.1. *In order to find the optimal single-price solution, it suffices to check at most $\min\{n, \tau\}$ possible prices, where $\tau = |\{p \in P : p \leq \mu\}|$.*

Proof. There are at most $\min\{n, \tau\}$ different values in the set $\{\text{val}(v) : v \in V\}$. It is never optimal to use any price $p \notin \{\text{val}(v) : v \in V\}$. Indeed, if $p \in (\text{val}(v_1), \text{val}(v_2))$, where $\text{val}(v_1)$ and $\text{val}(v_2)$ are two consecutive distinct values for some nodes $v_1, v_2 \in V$, then it is strictly better to set the price to $\text{val}(v_2)$. For the same reason, it is suboptimal to set a price that is less than the minimum value across nodes. Finally, we gain no revenue by using a price $p > \mu$. \square

Hence, in $O(n)$ steps we can find an optimal single-price solution. Let us denote by R_{sp} the revenue raised by such a solution. The performance of R_{sp} has been analyzed in a different context by Goldberg et al. [18], where it was shown that it achieves a $\Theta(\ln n)$ -approximation.¹ Here we give a somewhat tighter and more general statement which we utilize later on in Section 5.

¹Goldberg et al. [18] studied an auction pricing problem without any underlying network structure.

Recall that $p_1 < p_2 < \dots < p_k$. In what follows, $P_j = \sum_{i=1}^j \frac{p_i - p_{i-1}}{p_i}$, where $p_0 = 0$. Essentially, P_j generalizes the j th harmonic number H_j , when the price set P does not coincide with $[k]$.

Theorem 3.2. *Let $\tau \in [k]$ be such that $\mu = p_\tau$. Then the optimal single-price solution achieves a ρ -approximation, where $\rho = 1/\min\{H_n, P_\tau\}$. In particular,*

$$R_{\text{sp}} \geq \frac{\text{VAL}}{\min\{H_n, P_\tau\}} \geq \frac{\text{OPT}}{\min\{H_n, P_\tau\}},$$

and this approximation guarantee is tight.

Proof. By Lemma 3.1, we know that we only need to consider at most $\min\{n, \tau\}$ possible prices that correspond to the distinct values of the nodes. Let a_j be the number of vertices with value p_j and R_i be the revenue obtained by setting the price of all nodes equal to p_i , i.e.,

$$a_j = |\{v \in V : \text{val}(v) = p_j\}| \quad \text{and} \quad R_i = \sum_{v \in V} R_v(i) = p_i \sum_{j=i}^k a_j = p_i \sum_{j=i}^{\tau} a_j.$$

The last equality follows from observing that $a_j = 0$ for $j > \tau$.

Let R_{sp} be the revenue achieved by the optimal one-price algorithm. Then $R_i \leq R_{\text{sp}}$ and we have

$$\text{VAL} = \sum_{v \in V} \text{val}(v) = \sum_{i=1}^k \left((p_i - p_{i-1}) \sum_{j=i}^k a_j \right) = \sum_{i=1}^{\tau} \frac{(p_i - p_{i-1}) R_i}{p_i} \leq R_{\text{sp}} P_\tau.$$

So, we obtain

$$R_{\text{sp}} \geq \frac{\text{VAL}}{P_\tau} \geq \frac{\text{OPT}}{P_\tau}. \quad (1)$$

We now sort the vertices of V with respect to $\text{val}(v)$ in ascending order, say v_1, \dots, v_n . Let $R_{(i)}$ be the revenue obtained from the vertices $\{v_i, v_{i+1}, \dots, v_n\}$ by setting to all of them the price $\text{val}(v_i)$, i.e.,

$$R_{(i)} = \sum_{v \in \{v_i, v_{i+1}, \dots, v_n\}} R_v(\text{val}(v_i)) = (n - i + 1) \text{val}(v_i).$$

Clearly, $R_{(i)} \leq R_{\text{sp}}$ and we have

$$\text{VAL} = \sum_{i=1}^n \text{val}(v_i) = \sum_{i=1}^n \frac{R_{(i)}}{n - i + 1} \leq R_{\text{sp}} \sum_{i=1}^n \frac{1}{n - i + 1} = R_{\text{sp}} H_n.$$

Hence, we obtain

$$R_{\text{sp}} \geq \frac{\text{VAL}}{H_n} \geq \frac{\text{OPT}}{H_n}. \quad (2)$$

Putting inequalities (1) and (2) together completes the proof of the approximation guarantee.

To see that this is tight, consider the following family of graphs. For any n take $G(n)$ to be a clique on $\{v_1, v_2, \dots, v_n\}$ and let $\text{val}(v_i) = n!/i$ and $\alpha(u, v) = k = n!$ for every edge. Then

$$\text{OPT} = \text{VAL} = \sum_{i=1}^n \frac{n!}{i} = n!H_n \quad \text{and} \quad R_{n!/i} = \sum_{j=1}^i \frac{n!}{j} = n!, \quad \forall i \in \{1, 2, \dots, n\}.$$

Therefore, $\frac{R_{\text{sp}}}{\text{OPT}} = \frac{\max_{i \in \{1, 2, \dots, n\}} R_{n!/i}}{n!H_n} = \frac{1}{H_n}$.

In fact, tightness holds even when $P_\tau \leq H_n$. Consider an instance where $p_i = i, \forall i \in [k]$ and $n = k!$. Define $G(k)$ to be a clique on $\bigcup_{i=1}^k V_i$, where $V_i = \{v \in G(k) : \text{val}(v) = i\}$ and

$$\forall i \in [k-1], |V_i| = \frac{k!}{i(i+1)} \quad \text{and} \quad |V_k| = \frac{k!}{k}.$$

Notice that here $\tau = k$. Like before, $\alpha(u, v) = k$ for every edge. It is easy to verify that $\sum_{i=1}^k |V_i| = n$. Then

$$\text{OPT} = \text{VAL} = \sum_{i=1}^k i|V_i| = k \frac{n}{k} + \sum_{i=1}^{k-1} i \frac{n}{i(i+1)} = nH_k = nP_k,$$

while

$$\begin{aligned} \forall i \in [k], \quad R_i &= \sum_{j=i}^k i|V_j| = i n \left(\frac{1}{k} + \sum_{j=i}^{k-1} \frac{1}{j(j+1)} \right) \\ &= i n \left(\frac{1}{k} + \sum_{j=i}^{k-1} \left(\frac{1}{j} - \frac{1}{j+1} \right) \right) = i n \frac{1}{i} = n. \end{aligned}$$

Therefore, $\frac{R_{\text{sp}}}{\text{OPT}} = \frac{\max_{i \in \{1, 2, \dots, k\}} R_i}{nP_k} = \frac{1}{P_k}$. □

One interesting point here is that single-price solutions do not use any discontinuities. If R_{nd} is the optimal revenue without using any discontinuities, clearly $R_{\text{nd}} \geq R_{\text{sp}}$. As we mentioned in Section 2, it is possible to compute R_{nd} in polynomial time [2]. But then, why use the optimal single-price solution, which might be worse, instead of using directly R_{nd} ? The reason is that, besides being harder to argue about, R_{nd} turns out to be as bad an approximation as R_{sp} in the worst case, when allowing discontinuities.

The proposition below provides some further justification for the model with discontinuities, at least from the seller's point of view. In particular, it reveals that introducing discontinuities can cause a significant increase to the optimal revenue compared to what can be achieved without discontinuities.

Proposition 3.3. *The optimal solution with no discontinuities achieves a $1/H_r$ -approximation, where $r = \min\{n, \mu\}$, and this approximation guarantee is tight.*

Proof. The approximation guarantee follows from the fact that $R_{\text{nd}} \geq R_{\text{sp}}$, since single-price solutions do not use any discontinuities. To see that without using any discontinuities

one cannot always do better, it suffices to slightly modify the tightness examples from the proof of Theorem 3.2. In each case, we connect a new vertex v with value 1 to every vertex u and set $\alpha(u, v) = \alpha(v, u) = 0$. The optimal solution is to put a discontinuity on v and maximize the revenue of every other vertex. When discontinuities are not allowed though, a solution cannot do better than R_{sp} since all prices have to be equal. It is easy to see that we still get the same ratios, namely $1/H_n$ and $(n+1)/nH_\mu$. \square

4. An Exact Algorithm for Two Prices

We begin with an exact algorithm for the case of two arbitrary integer prices, i.e., $P = \{p_1, p_2\}$, for $p_1, p_2 \in \mathbb{N}$. As we later see in Section 6, this is the *only* case where we can hope for an exact algorithm, since the problem becomes hard for three prices, even when $P = \{1, 2, 3\}$. The natural interpretation of instances with two available prices is that a seller may be willing to offer only one type of discount to selected customers, e.g., 30% off the regular price, rather than using a bigger set of possible prices.

Having $P = \{p_1, p_2\}$, the available prices are p_1, p_2 and \perp . Recall that our standard assumption (without loss of generality) is that $p_1 < p_2$ and that for every node $v \in V$, $\text{val}(v) \in \{p_1, p_2\}$; see also the discussion in Section 2. Even when $p_1 = 1$ and $p_2 = 2$, the problem still remains non-trivial. For such instances we already have a $\frac{2}{3}$ -approximation by Theorem 3.2 that does not use discontinuities. The difficulty in improving the revenue is in finding a way of selecting appropriate nodes to set to \perp .

Before we formally state our algorithm, let us illustrate the main idea. Consider an instance of the problem on a graph $G = (V, E)$. We construct an appropriate bipartite graph H , such that feasible price vectors for G correspond to independent sets of H . Hence, the problem reduces to finding a maximum weighted independent set in a bipartite graph; the latter is well known to be solvable in polynomial time (see, e.g., [19]). To be more specific, denote by $A, B \subseteq V$ the partition of the vertices of G into vertices with $\text{val}(v) = p_1$ and $\text{val}(v) = p_2$ respectively. To construct the bipartite graph $H = (V_1 \cup V_2, E')$, we will be using a superscript for every node to clarify whether it belongs to V_1 or V_2 . For each $a \in A$, the graph H will have a vertex $a^1 \in V_1$; for each $b \in B$, H will have two vertices $b^1 \in V_1$ and $b^2 \in V_2$ connected with an edge. Additionally, we include an edge between x^2 and y^1 for every $x \in B, y \in V$ for which $(x, y) \in E(G)$ and $\alpha(x, y) < p_2 - p_1$. Note that H is bipartite, since there is no edge between vertices of the same superscript.

Algorithm 1: Exact algorithm when $P = \{p_1, p_2\}$

- 1 Given the graph $G = (V, E)$, construct the bipartite graph H with
 - $V(H) = \{v^1 \mid v \in V(G)\} \cup \{v^2 \mid v \in V(G) \text{ and } \text{val}(v) = p_2\}$ and
 - $E(H) = E_1 \cup E_2$, where
 - $E_1 = \{(v^1, v^2) \mid v \in V(G) \text{ and } \text{val}(v) = p_2\}$ and
 - $E_2 = \{(u^1, v^2) \mid (u, v) \in E(G), \text{val}(v) = p_2 \text{ and } \alpha(v, u) < p_2 - p_1\}$.
 - 2 Find a maximum weight independent set on H , say $S \subseteq V(H)$.
 - 3 For every $u^i \in S$, offer a price of p_i to the corresponding vertex u of $V(G)$.
 - 4 Set \perp to all the remaining vertices of $V(G)$.
 - 5 Return the resulting price vector.
-

Now consider any independent set S of H . We interpret $u^1 \in S$ as offering price p_1 to vertex u in the original instance and $u^2 \in S$ as offering price p_2 . Note that it cannot be the case that both u^1 and u^2 belong to S . If neither u^1 nor u^2 belong to S , we interpret this as introducing a discontinuity on u . To see that this is a feasible price vector, we only have to worry about edges (x, y) in the graph G for which $\alpha(x, y) < p_2 - p_1$, since all other constraints are trivially satisfied. But by construction, the graph H has an edge (x^2, y^1) whenever $\alpha(x, y) < p_2 - p_1$. Hence, we cannot include both x^2 and y^1 in S , and this ensures that either we offer the same price to such nodes, or one of them will have a discontinuity, implying that the resulting price vector is feasible.

Conversely, it is easy to see that any feasible price vector corresponds to an independent set of H . Given such a vector for the vertices of G , if price $p_i \in \{p_1, p_2\}$ is offered to v , we include the vertex v^i in the independent set of H .

We can further make H weighted by setting weight $p_i \in \{p_1, p_2\}$ to each vertex v^i , for $i = 1, 2$. Then, the total weight of an independent set equals the total revenue of the corresponding price vector and vice versa. Thus, in order to solve the Inequity Aversion Pricing on the original instance, it suffices to find a maximum weight independent set of H . The next theorem summarizes the above discussion.

Theorem 4.1. *If $P = \{p_1, p_2\}$, then Algorithm 1 solves optimally the Inequity Aversion Pricing problem in polynomial time.*

5. Approximation Algorithms

In this section we present new approximation algorithms for the problem by exploiting ways in which setting discontinuities in certain nodes can help. We start with $P = \{1, 2, \dots, k\}$, which is the most general price set studied before our work [2], and we propose an approximation algorithm with a ratio of $(H_k - 0.5)^{-1}$. Even though asymptotically this is no better than the optimal single-price algorithm, it is a significant improvement for instances where k is a small constant (see Table 1). We then generalize this result for arbitrary price sets.

5.1. An Algorithm for $P = [k]$

In order to improve the approximation guarantee of Theorem 3.2, we reduce the problem to the case of $k = 2$ and use the results of the previous section. Later, in Section 5.2, we show how to generalize the algorithm given here to arbitrary price sets. Towards this goal, we believe that working out first the case of $P = [k]$ facilitates the presentation.

Consider an instance of the problem, with available prices in $[k] \cup \{\perp\}$ and, as usual, $\text{val}(v) \in [k]$ for every $v \in V$. We create now another instance, where we set the value of every node with $\text{val}(v) > 1$ to be equal to 2. We can then run Algorithm 1 from Subsection 4 on this new instance. At the same time, we can also compute the optimal single-price solution for the original instance and pick the best among these two solutions. This yields Algorithm 2, described below.

Clearly, Algorithm 2 runs in polynomial time. Note that the solution returned by the algorithm is feasible. Any single-price solution is always feasible, while Algorithm 1 will produce a price vector that is feasible for I' , and therefore for I since the edge restrictions in the two instances are the same.

Algorithm 2: Algorithm for $P = [k]$

- 1 Given an instance I , construct the new instance I' , where for every $v \in V$, $\text{val}'(v) = \min\{\text{val}(v), 2\}$; everything else remains unchanged.
 - 2 Run Algorithm 1 on instance I' and let R_* be the revenue obtained.
 - 3 Compute the optimal single-price solution without discontinuities on the original instance I , as described in Section 3, with revenue R_{sp} .
 - 4 Return the solution that achieves $\max\{R_*, R_{\text{sp}}\}$.
-

Theorem 5.1. *Algorithm 2 achieves a $1/(H_\mu - 0.5)$ -approximation for Inequity Aversion Pricing when $P = [k]$, with $k \geq 2$, where $\mu = \max_{v \in V} \text{val}(v)$.*

Proof. The proof is by induction on μ . For $\mu = 2$ the result follows from Theorem 4.1 since $H_2 - 0.5 = 1$.

Now assume we have an instance I where $\mu = j > 2$. As usual, let OPT denote the optimal revenue for I and ALG the revenue returned by Algorithm 2. Also, let R_j be the revenue extracted by setting price j at every node, and $V_j = \{v \in V : \text{val}(v) = j\}$. We consider two cases.

Case (i): $\text{OPT} \leq (H_j - 0.5)j|V_j|$.

Then we have

$$\frac{\text{ALG}}{\text{OPT}} \geq \frac{R_j}{\text{OPT}} = \frac{j|V_j|}{\text{OPT}} \geq \frac{\frac{1}{H_j - 0.5} \text{OPT}}{\text{OPT}} = \frac{1}{H_\mu - 0.5}.$$

Case (ii): $\text{OPT} > (H_j - 0.5)j|V_j|$.

Let I^* be an instance derived from I by setting $\text{val}^*(v) = \min\{\text{val}(v), j - 1\}$, i.e., we only reduce the valuation of nodes v with $\text{val}(v) = \mu$ by 1. Let OPT^* and ALG^* denote the optimal revenue and the revenue returned by Algorithm 2 respectively, given I^* . By the inductive hypothesis we have $\text{OPT}^* \leq (H_{j-1} - 0.5)\text{ALG}^*$.

Furthermore, notice that the set of vertices with valuation greater than 1 is the same in both instances. So, Algorithm 2 on input I^* considers exactly the same price vectors as it does on input I , with the exception of the single-price solution that universally uses j . We conclude that $\text{ALG}^* \leq \text{ALG}$. Next, we prove the following useful claim.

Claim 5.2. $\text{OPT}^* \geq \text{OPT} - |V_j|$.

Proof of Claim. Let \mathbf{p} be an optimal price vector for I . Construct the price vector \mathbf{p}^* by decreasing any price that is at least j to $j - 1$. It is straightforward to see that in instance I we have $R(\mathbf{p}^*) \geq R(\mathbf{p}) - |V_j| = \text{OPT} - |V_j|$, while in both instances $R(\mathbf{p}^*)$ is the same. What is left to show is that \mathbf{p}^* is feasible for I^* . Observe, however, that the two instances have exactly the same edge restrictions and that \mathbf{p}^* did not increase the price difference between any two vertices compared to \mathbf{p} . Thus, $\text{OPT}^* \geq R(\mathbf{p}^*) \geq \text{OPT} - |V_j|$. \square

Now, we can write

$$\frac{\text{ALG}}{\text{OPT}} \geq \frac{\text{ALG}^*}{\text{OPT}} \geq \frac{\frac{1}{H_{j-1} - 0.5} \text{OPT}^*}{\text{OPT}} \geq \frac{\frac{1}{H_{j-1} - 0.5} (\text{OPT} - |V_j|)}{\text{OPT}}$$

$$\begin{aligned}
&\geq \frac{1}{H_{j-1} - 0.5} \left(1 - \frac{\frac{1}{j(H_j - 0.5)} \text{OPT}}{\text{OPT}} \right) = \frac{1}{H_{j-1} - 0.5} \cdot \frac{jH_j - 0.5j - 1}{j(H_j - 0.5)} \\
&= \frac{1}{H_{j-1} - 0.5} \cdot \frac{j(H_{j-1} - 0.5)}{j(H_j - 0.5)} = \frac{1}{H_j - 0.5},
\end{aligned}$$

which concludes the proof. \square

5.2. General Price Sets

We now extend our results from Section 5.1 to general price sets, i.e., to the case where $P = \{p_1, p_2, \dots, p_k\}$. This can be seen as a more realistic model, especially for small values of k . One could try to directly apply Theorem 3.2 or Theorem 5.1 for $P' = \{1, 2, 3, \dots, p_k\}$. However, this may produce a very poor approximation when k is small but n and p_k are large, and feasibility is not guaranteed either.

We are now ready to state the generalization of Theorem 5.1 for arbitrary price sets.

Theorem 5.3. *Let $P = \{p_1, p_2, \dots, p_k\}$. There exists a polynomial time $1/(P_k + \frac{p_1}{p_2} - 1)$ -approximation algorithm for the Inequity Aversion Pricing problem, where $P_k = \sum_{i=1}^k \frac{p_i - p_{i-1}}{p_i}$ and $p_0 = 0$.*

Proof. The algorithm is very similar to Algorithm 2. The only difference is in the definition of the instance I' (step 1 of Alg. 2). Namely, given an instance I , we define I' as the new instance where for every $v \in V$, $\text{val}'(v) = \min\{\text{val}(v), p_2\}$, for all $v \in V$, while the constraints remain the same.

The proof is by induction on k . For $k = 2$ the algorithm is equivalent to Algorithm 1 that gives an exact solution (Theorem 4.1). Note that we have

$$P_k + \frac{p_1}{p_2} - 1 = \frac{p_1 - p_0}{p_1} + \frac{p_2 - p_1}{p_2} + \frac{p_1}{p_2} - 1 = 1.$$

Now assume we have an instance I where $k > 2$. We use the notation of the proof of Theorem 5.1, but now R_k is the revenue extracted by setting price p_k at every node and $V_k = \{v \in V : \text{val}(v) = p_k\}$.

Case (i): $\text{OPT} \leq (P_k + \frac{p_1}{p_2} - 1) p_k |V_k|$.

Then we have

$$\frac{\text{ALG}}{\text{OPT}} \geq \frac{R_k}{\text{OPT}} = \frac{p_k |V_k|}{\text{OPT}} \geq \frac{1}{P_k + \frac{p_1}{p_2} - 1}.$$

Case (ii): $\text{OPT} > (P_k + \frac{p_1}{p_2} - 1) p_k |V_k|$.

Let I^* be an instance derived from I by setting $\text{val}^*(v) = \min\{\text{val}(v), p_{k-1}\}$. By the inductive hypothesis, we have $(P_{k-1} + \frac{p_1}{p_2} - 1) \text{ALG}^* \geq \text{OPT}^*$. It is easy to see that $\text{ALG}^* \leq \text{ALG}$. Also, we can prove an analog of Claim 5.2 (using the same arguments), namely that $\text{OPT}^* \geq \text{OPT} - (p_k - p_{k-1}) |V_k|$. Putting everything together we get

$$\frac{\text{ALG}}{\text{OPT}} \geq \frac{\text{ALG}^*}{\text{OPT}} \geq \frac{\frac{1}{P_{k-1} + \frac{p_1}{p_2} - 1} \text{OPT}^*}{\text{OPT}} \geq \frac{\frac{1}{P_{k-1} + \frac{p_1}{p_2} - 1} (\text{OPT} - (p_k - p_{k-1}) |V_k|)}{\text{OPT}}$$

$$\begin{aligned}
&> \frac{1}{P_{k-1} + \frac{p_1}{p_2} - 1} \left(1 - \frac{\frac{p_k - p_{k-1}}{p_k (P_k + \frac{p_1}{p_2} - 1)} \text{OPT}}{\text{OPT}} \right) \\
&= \frac{1}{P_{k-1} + \frac{p_1}{p_2} - 1} \cdot \frac{\frac{p_k}{p_k - p_{k-1}} P_k + \frac{p_k}{p_k - p_{k-1}} (\frac{p_1}{p_2} - 1) - 1}{\frac{p_k}{p_k - p_{k-1}} (P_k + \frac{p_1}{p_2} - 1)} \\
&= \frac{1}{P_{k-1} + \frac{p_1}{p_2} - 1} \cdot \frac{\frac{p_k}{p_k - p_{k-1}} (P_{k-1} + \frac{p_1}{p_2} - 1)}{\frac{p_k}{p_k - p_{k-1}} (P_k + \frac{p_1}{p_2} - 1)} = \frac{1}{P_k + \frac{p_1}{p_2} - 1},
\end{aligned}$$

which concludes the proof. \square

Table 1 contains examples of obtained approximation ratios. It is interesting to note that our results are robust to price scaling. For instance, the price set $\{10, 20, 30\}$ can be seen as scaled up version of $\{1, 2, 3\}$.

P	$\{1, 2\}$	$\{1, 2, 3\}$	$\{1, \dots, 10\}$	$\{1, \dots, 100\}$	$\{10, 20, 30\}$	$\{70, 80, 90, 100\}$
$1/H_k$	0.666	0.545	0.341	0.192	0.286	0.192
Alg. 2	1	0.750	0.412	0.213	–	–
Thm. 5.3	1	0.750	0.412	0.213	0.750	0.825

Table 1: The approximation ratios of Algorithm 2 (whenever it can be applied) and the algorithm implied by Theorem 5.3 for different sets of prices P . As a reference we also include the factor $1/H_k$ for the smallest k such that $P \subseteq [k]$.

6. Hardness for Single-Value Revenue Functions

Alon et al. [2] show an $n^{1-\varepsilon}$ inapproximability result for Inequity Aversion Pricing, but for general revenue functions and $\alpha(u, v) = 1$ for every edge. An NP-hardness proof is also given for these edge constraints when a mix of single-value and constant revenue functions is allowed. The NP-hardness of Inequity Aversion Pricing as we study it here, i.e., allowing only single-value revenue functions, was left as an open question. We resolve this question by proving that the problem remains NP-complete. Our reduction implies that the result holds even when the price differences are allowed to be comparable to the maximum possible price (which could presumably make the problem easier). Further, when $\alpha(u, v) = 0$ for every edge, we are able to show APX-hardness, as well as NP-hardness even for $P = \{1, 2, 3\}$, in contrast to the case of two possible prices.

To facilitate the presentation, in this section we assume that all prices up to the maximum allowed price are available, i.e., $P = \{1, 2, \dots, k\}$. Our reductions are self-contained but we assume some familiarity with the different types of reductions we use, like linear or PTAS reductions. For more details see, e.g., [4].

The reduction of Theorem 6.1 is from the decision version of *3-Terminal Node Cut*: Given a graph $G(V, E)$, a set $S = \{v_1, v_2, v_3\} \subseteq V$, and an integer q , is there a subset of q vertices that can be deleted, so that any two vertices of S are in different connected components of the resulting graph? The NP-completeness of the weighted version of 3-Terminal Node Cut is discussed by Cunningham [10], while the APX-hardness of the unweighted version we use here is discussed by Garg et al. [17]. The NP-completeness result we need follows from Theorem 6.4 as well (see the discussion before the statement of Theorem 6.4).

Theorem 6.1. *Let $\varepsilon > 0$ be any small constant. The decision version of Inequity Aversion Pricing for single-value revenue functions is NP-complete, even when $\alpha(u, v)$ is as large as $k^{1-\varepsilon}$ for all $(u, v) \in E(G)$, where k is the maximum possible price.*

Proof. It is immediate that the problem is in NP. To facilitate the presentation, we prove the NP-hardness when $\alpha(\cdot, \cdot)$ is upper bounded by $k^{1/3}/3$. As discussed at the end of the proof, the reduction can be easily adjusted when the upper bound of $\alpha(\cdot, \cdot)$ is $k^{1-\varepsilon}$, for constant ε .

Let us consider an instance of 3-Terminal Node Cut, i.e., a graph $G(V, E)$ with $|V(G)| = n$, a set $S = \{v_1, v_2, v_3\}$ of non adjacent vertices of G and an integer q . We may assume that $q \leq n - 3$, otherwise the question is trivial. Next we give a construction of an appropriate instance for Inequity Aversion Pricing.

Let H be the graph obtained from G as follows. We replace every vertex $v \in S$ by n^3 vertices, where each such vertex has the same neighbors as v , i.e., if u_v is a vertex in the bundle of vertices replacing v , then for every edge $(v, x) \in E(G)$ we add the edge (u_v, x) to $E(H)$. For any $v \in S$, we call such a set of vertices in H a v -bundle. The set of prices is $\{1, 2, \dots, k\}$, where $k = n^3 + n^2$. Finally, for any $(u, v) \in E(H)$ we set $\alpha(u, v)$ and $\alpha(v, u)$ arbitrarily, as long as they are at most $k^{1/3}/3$. Note that $|V(H)| = n - 3 + 3n^3$, and $|E(H)| \leq |E(G)| + 3(n - 1)n^3 \leq 3n^4$.

Next we define the single-value revenue functions for the vertices of H . For every $v \in V(G) \setminus S$, let $\text{val}(v) = n^3 + n^2$, and for every $v_i \in S$, let $\text{val}(u_{v_i}) = n^3 + \frac{i-1}{2}n^2$ for all u_{v_i} in the v_i -bundle. We show below that G has a subset of at most q vertices that separate all the vertices of S , if and only if there is a feasible choice of prices for the vertices of H that gives revenue at least R_q , where $R_q = (n - 3 - q)n^3 + \sum_{i=1}^3 n^3 \left(n^3 + \frac{i-1}{2}n^2\right)$.

One direction is easy. Let A be a subset of at most q vertices of G that separate the three vertices of S . For all $v \in A$ we put a discontinuity on the corresponding v in H . If we think of these vertices as removed from H , this creates several connected components. For any other vertex $u \in V(H)$, if u is in the same component as some v_i -bundle (or itself is one of the vertices of the v_i -bundle), set its price to $n^3 + \frac{i-1}{2}n^2$, otherwise set its price to $n^3 + n^2$. Notice that any vertex without a discontinuity produces revenue at least n^3 , while any vertex u_{v_i} in a v_i -bundle with $v_i \in S$ produces revenue exactly $n^3 + \frac{i-1}{2}n^2$. Now, it is straightforward to check that this price vector \mathbf{p} is feasible and gives enough revenue: $R(\mathbf{p}) = \sum_{u \in V(H)} R(u) \geq (n - 3 - q)n^3 + \sum_{i=1}^3 n^3 \left(n^3 + \frac{i-1}{2}n^2\right) = R_q$.

For the opposite direction we begin with a couple of observations. Assume that there is a price vector \mathbf{p}_* that gives revenue at least R_q . We claim that \mathbf{p}_* can have only a few discontinuities.

Claim 6.2. *There is no feasible price vector \mathbf{p} with $R(\mathbf{p}) \geq R_q$ and more than q discontinuities.*

Proof of Claim. Let \mathbf{p} be a feasible price vector with at least $q + 1$ discontinuities. Notice that any vertex without a discontinuity produces revenue at most $n^3 + n^2$ and, in particular, any vertex u_{v_i} in a v_i -bundle with $v_i \in S$ produces revenue at most $n^3 + \frac{i-1}{2}n^2$. The maximum possible revenue for \mathbf{p} is

$$\begin{aligned} R(\mathbf{p}) &\leq (n - 3)(n^3 + n^2) + \sum_{i=1}^3 n^3 \left(n^3 + \frac{i-1}{2}n^2\right) - (q + 1)n^3 \\ &= R_q + (n - 3)n^2 - n^3 < R_q, \end{aligned}$$

thus proving the claim. \square

One immediate implication of Claim 6.2 is that for any $v \in S$ not every vertex in the v -bundle has price \perp . This holds because the v -bundle has n^3 vertices and only $q \leq n - 3$ of them can get \perp . This is crucial, because if we think of the vertices with price \perp as removed from H , then no two vertices are separated because of discontinuities in the v -bundles. In particular, we can completely ignore those discontinuities with respect to connectivity.

Let $D_{\mathbf{p}} = \{v \in V(G) \setminus S : \mathbf{p}_v = \perp\}$, i.e., $D_{\mathbf{p}}$ is the set of non terminal vertices in G that their corresponding vertices in H have discontinuities in \mathbf{p} . So far, by Claim 6.2, we have that $|D_{\mathbf{p}}| \leq q$. What is left to be shown is that these discontinuities separate the v -bundles, for any $v \in S$.

Claim 6.3. *There is no feasible price vector \mathbf{p} such that $R(\mathbf{p}) \geq R_q$, and for some $v_i, v_j \in S$ vertices from both the v_i -bundle and the v_j -bundle are in the same connected component of the graph $H' = H - \{v \in V(H) : v \text{ is not in a bundle and } \mathbf{p}_v = \perp\}$.*

Proof of Claim. Let \mathbf{p} be a feasible price vector and assume that there exist $v_i, v_j \in S$ such that vertices from both the v_i -bundle and the v_j -bundle are in the same component of H' . First notice that all the vertices in the v_i -bundle and the v_j -bundle are in the same component, since vertices in a bundle share the same neighbors. We are going to upper bound the maximum possible revenue for such a price vector. Without loss of generality, assume $i < j$. If all the vertices in the v_i -bundle are assigned prices in $\{\perp, n^3 + \frac{i-1}{2}n^2 + 1, \dots, k\}$, then they contribute 0 to the total revenue. On the other hand, if there is some vertex in the v_i -bundle with price at most $n^3 + \frac{i-1}{2}n^2$, then by the feasibility of \mathbf{p} we have that any vertex in the v_j -bundle has its revenue upper bounded by $n^3 + \frac{i-1}{2}n^2 + \frac{k^{1/3}}{3}n$. To see the latter, notice that if any two vertices from two distinct bundles are connected by a path, then this path has length at most n (like it would in G) and therefore their prices can differ by $\frac{k^{1/3}}{3}n$ at most. We conclude that the loss, compared to the sum of the maximum revenues per vertex, is lower bounded by either $n^3 (n^3 + \frac{i-1}{2}n^2)$ or $n^3 \left(\frac{j-i}{2}n^2 - \frac{k^{1/3}}{3}n \right)$ and therefore by $n^3 \left(\frac{1}{2}n^2 - \frac{k^{1/3}}{3}n \right)$. For $n \geq 10$, we have

$$\begin{aligned} n^3 \left(\frac{n^2}{2} - \frac{(n^3 + n^2)^{1/3} n}{3} \right) &\geq n^3 \left(\frac{n^2}{2} - \frac{1.1^{1/3} n^2}{3} \right) > 0.15 n^2 n^3 \\ &> (n-2)n^3 \geq (q+1)n^3 \end{aligned}$$

and we get $R(\mathbf{p}) < R_q$ in exactly the same way as in the proof of Claim 6.2. \square

We conclude that $D_{\mathbf{p}_*}$ is a set of at most q vertices of G that separate all the vertices of S . This completes the proof for the case where $\alpha(\cdot, \cdot)$ is upper bounded by $k^{1/3}/3$.

The above reduction, however, generalizes for $\alpha(\cdot, \cdot)$ upper bounded by $k^{1-\varepsilon}$ for any positive constant ε . Let $c \in \mathbb{N}$ with $c > 4/\varepsilon$. If we multiply by n^c all the relevant quantities, i.e., the size of the bundles, k , R_q , and $\text{val}(v)$ for all $v \in V(H)$, then the reduction is identical up to the last part of the proof of Claim 6.3. Now, the loss is lower bounded by $n^{c+3} (n^{c+2}/2 - nk^{1-\varepsilon})$ and it suffices for this quantity to be at least $(q+1)n^{c+3}$ for things to work out. So, we need $n^{c+2}/2 - nk^{1-\varepsilon} \geq n-2$ (since $n-2 \geq q+1$), and it is only a matter of simple calculations to check that this holds. \square

For the special case where all the differences are 0, we show that the problem is APX-hard. In doing so, we prove that 3-Terminal Node Cut is MAX SNP-hard, and thus APX-hard. As noted already, MAX SNP-hardness of 3-Terminal Node Cut is discussed—but not explicitly proved—by Garg et al. [17]. Here, having this reduction is crucial, and we have therefore obtained an explicit construction, since eventually we need to show that 3-Terminal Node Cut restricted in a specific set of instances is MAX SNP-hard (Corollary 6.5).

Theorem 6.4. *Multi-Terminal Node Cut is MAX SNP-hard even for 3 terminals and all the weights equal to 1.*

Proof. We prove the result for 3 terminals. The extension to more follows immediately. Proofs of MAX SNP-hardness involve linear reductions. Let A and B be two optimization problems. We say that A linearly reduces to B if there are two polynomial time computable functions f and g and constants $c_\alpha, c_\beta > 0$ such that

- Given an instance a of A , f produces an instance $b = f(a)$ of B such that $\text{OPT}_B(b) \leq c_\alpha \text{OPT}_A(a)$, and
- Given a , $b = f(a)$, and any solution y of b , g produces a solution x of a such that $|\text{cost}_A(x) - \text{OPT}_A(a)| \leq c_\beta |\text{cost}_B(y) - \text{OPT}_B(b)|$.

The reduction is from the unweighted version of *3-Terminal Cut*: Given a graph $G(V, E)$ and a set $S = \{v_1, v_2, v_3\} \subseteq V$, find a minimum cardinality set of edges that can be deleted, so that any two vertices of S are in different connected components of the resulting graph. Dahlhaus et al. [11] showed that 3-Terminal Cut is MAX SNP-hard even when all the weights equal to 1, which is essentially the unweighted version defined above.

Consider an instance of 3-Terminal Cut, i.e., a graph $G(V, E)$ with $|V(G)| = n$ and a set of non adjacent terminals $S = \{v_1, v_2, v_3\}$. We first describe the function f in the definition of the linear reduction. Let H be the graph obtained from G as follows:

1. Replace each edge e by a path of length two, the middle vertex of which we denote by v_e .
2. Replace every “old” vertex v by a v -bundle of $\deg_G(v) + 1$ vertices (see also the proof of Theorem 6.1), where each such vertex has the same neighbors as v in the graph constructed at step 1. That is, put an edge between u_v and v_e if u_v is a vertex in the v -bundle and e is incident to v .

Also, let $S' = \{u_1, u_2, u_3\}$, where u_i is an arbitrarily chosen vertex from the v_i -bundle. Define $f((G, S)) = (H, S')$. Clearly, f is polynomial time computable.

Next we define the function g in the definition of the linear reduction. Given a vertex cut Y in H that separates the vertices in S' , first we transform it to an appropriate vertex cut Y' that separates the vertices in S' and contains no vertices from any v -bundle.

1. While there is a whole v -bundle contained in the vertex cut, remove those vertices from the cut and add all of their neighbors instead.
2. While there is some vertex from a v -bundle in the cut, just remove this vertex from the cut.

Notice that in one iteration of step 1 the connectivity is not improved and the size of the vertex cut is reduced. The latter holds because $\deg_G(v) + 1$ vertices were removed from the cut and at most $\deg_G(v)$ were added. Similarly, in one iteration of step 2 the connectivity is not improved and the size of the vertex cut is reduced. Now the latter is obvious, but to see that the connectivity is not improved, notice that the removal of vertices in some v -bundle has an effect in connectivity only if the whole v -bundle is removed. Since in step 2 there are no v -bundles completely contained in the vertex cut (this was fixed in step 1), the vertices removed from the cut were not disconnecting anything to begin with. We conclude that Y' is indeed a vertex cut that separates the vertices in S' and moreover $|Y'| \leq |Y|$.

Now that Y' contains only vertices outside the v -bundles, i.e., only vertices that correspond to edges of G , it is straightforward to define an edge cut in G that separates the vertices in S . Let $X = \{e \in E(G) : v_e \in Y'\}$, i.e., X is the set of edges in G that their corresponding vertices in H are in the vertex cut. Define $g((G, S), (H, S'), Y)$ to be equal to X ; clearly, g is polynomial time computable. It remains to be shown that X separates the vertices in S . Assume not; then there exists some $v_i - v_j$ path $p = (v_i, x_1, x_2, \dots, x_k, v_j)$ in $G - X$ for $v_i, v_j \in S$, with $i \neq j$. This, however, directly transforms to a $u_i - u_j$ path $p' = (u_i, v_{(v_i, x_1)}, x'_1, v_{(x_1, x_2)}, x'_2, \dots, x'_k, v_{(x_k, v_j)}, u_j)$ in $H - Y'$, where x'_ℓ is an arbitrary vertex in the x_ℓ -bundle. This is a contradiction. Thus, X is a cut that separates the vertices in S .

Next, we prove that $\text{OPT}_{3\text{TNC}}(H) \leq \text{OPT}_{3\text{TC}}(G)$ (to improve readability we drop the subscripts). Notice that any cut A in G that separates the vertices of S gives the vertex cut $B = \{v_e \in V(H) : e \in A\}$ in H that separates the vertices of S' . Since $|B| = |A|$, and by taking $|A|$ to be an optimal cut, we have $\text{OPT}(H) \leq \text{OPT}(G)$. This also implies that $c_\alpha = 1$ works.

Finally, since $|X| = |Y'|$, we have $|X| - \text{OPT}(G) \leq |Y'| - \text{OPT}(H) \leq |Y| - \text{OPT}(H)$, i.e., $c_\beta = 1$ works. We conclude that the unweighted version of 3-Terminal Node Cut is MAX SNP-hard. \square

As proved by Khanna et al. [23], APX is the closure of MAX SNP under PTAS reductions (introduced by Crescenzi and Trevisan [9]). Therefore, any MAX SNP-hard problem is also APX-hard. Let \mathcal{I} be the set of instances of 3-Terminal Node Cut that can be the result of the composition of the reduction of Theorem 6.4 with the linear reduction from Max Cut to 3-Terminal Cut, presented by Dahlhaus et al. [11]. The next corollary follows directly.

Corollary 6.5. *3-Terminal Node Cut is MAX SNP-hard, and thus APX-hard, even when restricted on instances in \mathcal{I} .*

Corollary 6.5 is a crucial step towards our goal, since instances in \mathcal{I} are guaranteed to have only “large” vertex cuts that separate the terminals.

Lemma 6.6. *Let $(G, S, q) \in \mathcal{I}$. Then, any feasible 3-Terminal Node Cut solution for (G, S, q) has size greater than $\frac{1}{14}|V(G)|$.*

Proof. Let G_0 be a graph with n_0 vertices and m_0 edges. The reduction of Dahlhaus et al. [11] adds 3 terminals and, furthermore, for each edge adds 4 new vertices and 102 new edges. In fact, each edge is replaced with the gadget shown in Figure 1 (Figure 11 of Dahlhaus et al. [11]), where s_1, s_2, s_3 are identified with the terminals and x, y with the

endpoints of the edge. Then, each of the 12 edges with weight 4 is replaced by 4 paths of length 2. The resulting graph G_1 , has $n_1 = n_0 + 3 + 52m_0$ vertices and $m_1 = 102m_0$ edges.

Our reduction adds 1 new vertex for each edge, and then replaces each one of the old vertices with $\deg_{G_1}(v) + 1$ new vertices. The number of vertices of the resulting graph G_2 is $n_2 = \sum_{v \in V(G_1)} (\deg_{G_1}(v) + 1) + m_1 = n_1 + 3m_1 = n_0 + 3 + 358m_0 < 378m_0$.

By the proof of Theorem 3 in Dahlhaus et al. [11], we have that any cut in G_1 that separates the 3 terminals has size at least $27m_0$. Using g from our reduction, however, we can transform a vertex cut that separates the 3 terminals in G_2 into a cut of the same cardinality that separates the 3 terminals in G_1 . Thus, any vertex cut that separates the 3 terminals in G_2 has size at least $27m_0$. To complete the proof, notice that $27m_0 > 27n_2/378 = n_2/14$. \square

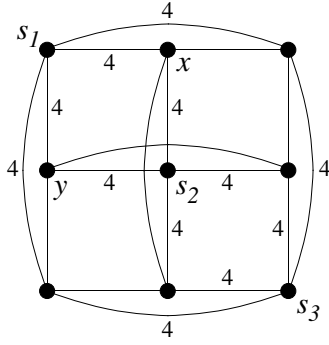


Figure 1: The gadget that “replaces” every edge in the linear reduction of from Max Cut to 3-Terminal Cut [11].

Theorem 6.7. *Inequity Aversion Pricing for single-value revenue functions is APX-hard when $\alpha(e) = 0$ for all $e \in E(G)$.*

Proof. We use a PTAS reduction to prove the APX-hardness. Let A and B be two NPO problems. Here assume that A is a minimization and B is a maximization problem. We say that A is PTAS-reducible to B if there exist three computable functions f , g , and c such that

- For any instance x of A and any $r > 1$, $f(x, r)$ is an instance of B computable in time polynomial in $|x|$.
- For any instance x of A , any $r > 1$, and any feasible solution y of $f(x, r)$, $g(x, y, r)$ is a feasible solution of A computable in time polynomial in both $|x|$ and $|y|$.
- $c : (1, \infty) \rightarrow (0, 1)$
- For any instance x of A , any $r > 1$, and any feasible solution y of $f(x, r)$,

$$\text{cost}_B(y) \geq c(r) \cdot \text{OPT}_B(f(x, r)) \text{ implies } \text{cost}_A(g(x, y, r)) \leq r \cdot \text{OPT}_A(x).$$

The reduction is from the restriction of 3-Terminal Node Cut on \mathcal{I} . It is similar to the reduction in the proof of Theorem 6.1, but here all the parameters are carefully tuned. Consider an instance of 3-Terminal Node Cut, i.e., a graph $G(V, E)$ with $|V(G)| = n$, a set of non adjacent terminals $S = \{v_1, v_2, v_3\}$, and an integer q , such that $(G, S, q) \in \mathcal{I}$. We describe the function f in the definition of a PTAS reduction.

For $r > 1$, let $\varepsilon = \min\{0.5, r - 1\}$ and $t = \lceil \frac{42}{\varepsilon} \rceil$. Also, let $H = f(G, r)$ be the graph obtained from G by replacing every vertex $v \in S$ by a v -bundle of $4tn$ vertices, each such vertex having the same neighbors as v . The set of prices is $\{\perp, 1, 2, \dots, t\}$. To define the single-value revenue functions, for every $v \in V(G) \setminus S$, let $\text{val}(v) = t$, and for every $v_i \in S$, let $\text{val}(u_{v_i}) = t + i - 3$ for all u_{v_i} in the v_i -bundle. We define $f((G, S, q), r)$ to be the above instance. Clearly, f is computable in polynomial time in n .

Next we define the function g in the definition of a PTAS reduction. Given a feasible price vector \mathbf{p} for H , first we transform it to an appropriate feasible price vector \mathbf{p}' .

1. While there is a whole v -bundle only with discontinuities, set price $\text{val}(u_v)$ to all the vertices u_v in this v -bundle and \perp to all of their neighbors.
2. Consider the graph after we remove all the vertices with price \perp . While there are i, j (assume $i < j$) such that vertices from both the v_i -bundle and the v_j -bundle are in the same connected component:
 - If all the vertices in the v_i -bundle are assigned prices in $\{\perp, \text{val}(u_{v_i}) + 1, \dots, t\}$, then set price $\text{val}(u_{v_i})$ to all the vertices in the v_i -bundle and \perp to all of their neighbors.
 - Otherwise, set price $\text{val}(u_{v_j})$ to all the vertices in the v_j -bundle and \perp to all of their neighbors.

Then, we use this price vector \mathbf{p}' in order to define a solution D to the 3-Terminal Node Cut instance. Let $D = \{v \in V(G) \setminus S : \mathbf{p}'_v = \perp\}$, i.e., D is the set of non terminal vertices in G that their corresponding vertices in H have discontinuities. Again, it is straightforward to see that computing $g((G, S, q), \mathbf{p}, r)$ takes polynomial time in n .

It remains to determine the function c in the definition of the reduction; for any $r \in (1, \infty)$ let $c(r) = 1 - \frac{1}{20r^2}$. We need to show that

$$R(\mathbf{p}) \geq c(r) \cdot \text{OPT}(H) \implies \text{cost}(D) \leq r \cdot \text{OPT}(G).$$

Claim 6.8. *If $R(\mathbf{p}) \geq c(r)\text{OPT}(H)$, then $\mathbf{p}' = \mathbf{p}$. That is, there is no v -bundle only with discontinuities and every v -bundle is in a different connected component.*

Proof of Claim. For the first part, assume that there is a v -bundle where every single vertex gets price \perp . We get the following upper bound for $R(\mathbf{p})$:

$$R(\mathbf{p}) \leq (n - 3)t + \sum_{i=2}^3 4tn(t + i - 3) \leq 8t^2n - 3tn - 3t < 8t^2n.$$

On the other hand, there is a feasible price vector that sets all the prices equal to $t - 2$, and this way we get a lower bound on $\text{OPT}(H)$.

$$\text{OPT}(H) \geq (3 \cdot 4tn + n - 3)(t - 2) = 12t^2n - 23tn - 2n - 3t + 6 > 12t^2n - 28tn.$$

Notice that, since $\varepsilon \leq 0.5$, we have $t \geq 84$, and therefore $c(r) > 0.99$. So, we have $\frac{R(\mathbf{p})}{\text{OPT}(H)} < \frac{8t}{12t-28} < 0.8 < c(r)$, which is a contradiction.

For the second part, assume that there are two v -bundles in the same component. To arrive at a contradiction, it suffices to show that there exists a feasible price vector \mathbf{p}'' , such that $R(\mathbf{p}) < c(r)R(\mathbf{p}'')$ and therefore $R(\mathbf{p}) < c(r)\text{OPT}(H)$. Let \mathbf{p}'' be the price vector obtained after just one iteration of step 2 in the description of g . Assuming that we are talking about the v_i -bundle and the v_j -bundle, with $i < j$, the gain in revenue is at least $4tn(\text{val}(u_{v_j}) - \text{val}(u_{v_i})) \geq 4tn$ (see also the proof of Claim 6.3 in the proof of Theorem 6.1). On the other hand, the loss in revenue is upper bounded by $(n-3)(t+i-3) \leq tn$. So, $R(\mathbf{p}'') \geq R(\mathbf{p}) + 3tn$. Suppose $R(\mathbf{p}) \geq c(r)R(\mathbf{p}'')$. Then it is a matter of simple calculations to see that

$$R(\mathbf{p}) \geq c(r)(R(\mathbf{p}) + 3tn) \implies R(\mathbf{p}) \geq 60t^3n - 3tn > 57t^3n.$$

An obvious upper bound for $R(\mathbf{p})$ however, is to say that each vertex produces revenue at most t , i.e., $R(\mathbf{p}) \leq (12tn + n - 3)t < 13tn$. Combining the two, we get the contradiction $R(\mathbf{p}) > 57t^3n > 13tn > R(\mathbf{p})$. We conclude that $R(\mathbf{p}) < c(r)R(\mathbf{p}'')$ which leads to the contradiction $R(\mathbf{p}) < c(r)\text{OPT}(H)$. Hence, in the graph defined by removing the discontinuities of \mathbf{p} from H , every v -bundle is in a different connected component. \square

Claim 6.9. *If $R(\mathbf{p}) \geq c(r)\text{OPT}(H)$, then \mathbf{p} has less than $(1 + \varepsilon)\text{OPT}(G)$ discontinuities.*

Proof of Claim. Let \mathbf{p} be a feasible price vector with $R(\mathbf{p}) \geq c(r)\text{OPT}(H)$ and assume that \mathbf{p} has at least $(1 + \varepsilon)\text{OPT}(G)$ discontinuities. Also, consider the feasible price vector \mathbf{p}^* induced by an optimal cut in G , i.e., the price vector that sets \perp in every vertex that has a corresponding vertex removed by the cut in G and then uses optimal single price in each ‘‘connected component’’. To get a contradiction, we show that $R(\mathbf{p}) < c(r)R(\mathbf{p}^*)$ and therefore $R(\mathbf{p}) < c(r)\text{OPT}(H)$. To obtain a lower bound on $R(\mathbf{p}^*)$, notice that any vertex without a discontinuity produces revenue at least $t - 2$, while any vertex u_{v_i} in a v_i -bundle produces revenue exactly $t + i - 3$. So,

$$R(\mathbf{p}^*) \geq (n - 3 - \text{OPT}(G))(t - 2) + \sum_{i=1}^3 4tn(t + i - 3).$$

To get an upper bound for $R(\mathbf{p})$, notice that each vertex without a discontinuity produces revenue at least $t - 2$ and at most t , while any vertex u_{v_i} in a v_i -bundle produces revenue exactly $t + i - 3$, i.e.,

$$R(\mathbf{p}) \leq (n - 3)t - (1 + \varepsilon)\text{OPT}(G)(t - 2) + \sum_{i=1}^3 4tn(t + i - 3).$$

We consider the difference $R(\mathbf{p}) - c(r)R(\mathbf{p}^*)$, and show it is negative. Recall that Lemma 6.6 implies that $\text{OPT}(G) \geq n/14$.

$$\begin{aligned} R(\mathbf{p}) - c(r)R(\mathbf{p}^*) &\leq \frac{1}{20t^2} \left((n - 3 - \text{OPT}(G))(t - 2) + \sum_{i=1}^3 4tn(t + i - 3) \right) \\ &\quad + 2(n - 3) - \varepsilon \text{OPT}(G)(t - 2) \end{aligned}$$

$$\begin{aligned}
&< \frac{1}{20t^2} (nt + 12t^2n) + 2n - \varepsilon \frac{1}{14} n \left(\frac{42}{\varepsilon} - 2 \right) \\
&< \frac{13n}{20} + 2n - 2.9n < -0.25n < 0,
\end{aligned}$$

which leads to contradiction. Thus, \mathbf{p} has less than $(1 + \varepsilon)\text{OPT}(G)$ discontinuities. \square

By combining Claim 6.8, Claim 6.9, and the fact that $1 + \varepsilon \leq r$, we directly get that a $c(r)$ -approximate solution for H gives an r -approximate solution for G , thus concluding the proof. \square

Remark 6.10. The maximum price k in the instance constructed in the proof of Theorem 6.7 does not depend on the size of the problem. Given that there is some constant ρ beyond which it is hard to approximate 3-Terminal Node Cut, this means that there exists some constant k^* for which Inequity Aversion Pricing does not have a PTAS (unless $P = NP$). Note that for such a k^* we do have a constant factor approximation, with factor $H_{k^*}^{-1}$.

6.1. Hardness when $k = 3$

We close this section by showing that Inequity Aversion Pricing remains hard even when we only have three possible prices and $\alpha(e) = 0$ for all edges. This identifies the transition from polynomial time solvability, which we have when $k = 2$, to NP-hardness with respect to the number of available prices.

Theorem 6.11. *Inequity Aversion Pricing for single-value revenue functions is NP-complete when $\alpha(e) = 0$ for all $e \in E(G)$, even if the price set is $P = \{1, 2, 3\}$.*

The theorem follows from the fact that the problem is trivially in NP and the next three lemmas, each consisting of a simple reduction. We begin with the definition of two intermediate problems used in those reductions.

Definition 6.12. *The Tripartite Independent Set problem is the restriction of Independent Set on tripartite graphs. In particular, given a tripartite graph, a tripartition of its vertices, and an integer q , is there an independent set of size at least q ?*

The next problem is a stricter version of our problem, regarding the price that we are allowed to offer to each node.

Definition 6.13. *The Strict Inequity Aversion Pricing problem is a variant of Inequity Aversion Pricing in which $\alpha(e) = 0$ for all $e \in E(G)$ and the seller is disallowed to offer a customer a price different from the customer's valuation, i.e. $p_v \in \{\text{val}(v), \perp\}$ for each node v .*

Lemma 6.14. *Tripartite Independent Set is NP-hard.*

Proof. We reduce the general Independent Set problem to Tripartite Independent Set using a construction by Poljak [25].

Given a graph G with n vertices and m edges, we 2-subdivide its edges, i.e., replace each edge with a path of length 3, to obtain a graph H , which is clearly tripartite. We call the vertices added by 2-subdivisions *new* as opposed to the *old* vertices coming from G . Now G has an independent set of size q if and only if H has an independent set of size $q + m$: Starting from an independent set of G , we can add to it one of the

two new vertices on each 2-subdivided edge. Conversely, every independent set of H can be transformed into one that is not smaller and contains precisely m new vertices (one for each 2-subdivided edge). The old vertices of this independent set then form an independent set of G . \square

Lemma 6.15. *Strict Inequity Aversion Pricing with price set $P = \{1, 2, 3\}$ is NP-hard.*

Proof. We give a reduction from Tripartite Independent Set. Given a tripartite graph G , a tripartition V_1, V_2, V_3 of its vertices, and an integer q , we construct an instance of Strict Inequity Aversion Pricing as follows: For each vertex $v \in V_i$, we have a bundle of $6/i$ nodes v' with $\text{val}(v') = i$, for $i = 1, 2, 3$. For each edge (u, v) of G , we add constraints between all pairs (u', v') of nodes associated with u and v respectively, setting $\alpha(u', v') = \alpha(v', u') = 0$. Call H the resulting graph and let $R_q = 6q$.

We claim that G has an independent set of size q if and only if there is a feasible price vector for the above instance that guarantees revenue R_q . One direction is straightforward. For every vertex v in an independent set of size q in G , we set price $\text{val}(v')$ to every vertex v' of the corresponding v -bundle in H . This way no constraint is violated, since we started with an independent set, and each bundle contributes to the total revenue either a value of 6, if it corresponds to a vertex in the independent set, or 0, for a total of $6q$.

Conversely, suppose that there is a feasible price vector \mathbf{p} for H that guarantees revenue R_q , for the strict version of Inequity Aversion Pricing. Because all nodes v' in a bundle of H have the same neighborhood, they may be given the same offer (i.e., $\text{val}(v')$ or \perp). If this is not already the case for \mathbf{p} , we can find such a feasible price vector \mathbf{p}' by using in each bundle the maximum price that \mathbf{p} uses on any vertex of this bundle. Since the new prices only go up (without affecting feasibility), \mathbf{p}' guarantees revenue $R' \geq R_q$. Moreover, under \mathbf{p}' , a bundle of nodes contributes to the total revenue either 6 or 0, regardless of which part V_i their associated vertex v belongs to. Let us denote by S the set of vertices of G such that their associated nodes were not assigned \perp . From the construction it follows that S is an independent set in G , since we have that $\alpha(e) = 0$ for every edge e , and therefore, for an edge (u, v) in G , it cannot be the case that the associated nodes for both u and v in H were not assigned \perp . The size of S is $R'/6 \geq R_q/6 = q$, which concludes the proof. \square

Lemma 6.16. *Inequity Aversion Pricing with price set $P = \{1, 2, 3\}$ is NP-hard.*

Proof. We give a reduction from Strict Inequity Aversion Pricing with price set $\{1, 2, 3\}$. Consider an instance of the decision version of the problem, i.e., a graph G with edge constraints $\alpha(\cdot, \cdot) = 0$, and a single-value revenue function for each node, along with a positive integer t . Let $n_i = |\{v \in V(G) : \text{val}(v) = i\}|$ for $i \in \{1, 2, 3\}$. We construct an instance of Inequity Aversion Pricing as follows: For each node $v \in G$, we add $\text{val}(v)$ new nodes v' with $\text{val}(v') = \text{val}(v)$ and impose constraints $\alpha(v, v') = \alpha(v', v) = 0$, forming a star with v at its center. Call H the resulting graph and let $t' = t + n_1 + 4n_2 + 9n_3$.

We first observe that if there is a feasible price vector \mathbf{p} for G that produces revenue t , then we can use it to set the price on the old nodes of H , while for each new node v' we set its price to $\text{val}(v')$, and the resulting price vector \mathbf{p}' is still feasible. The feasibility of \mathbf{p}' follows from the fact that for every old node v , \mathbf{p} sets a price of $\text{val}(v)$ or \perp . By the construction of H , it is straightforward to see that this way we extract revenue t' .

Conversely, suppose that there is a feasible price vector \mathbf{p}' for H that produces revenue $t' \geq t + n_1 + 4n_2 + 9n_3$. We will construct a feasible price vector \mathbf{p} for G as follows. For each $v \in G$, $\mathbf{p}_v = \perp$ if $\mathbf{p}'_v \neq \text{val}(v)$, and $\mathbf{p}_v = \mathbf{p}'_v$ otherwise. Feasibility follows from the feasibility of \mathbf{p}' for H (we have only introduced more \perp s). We next show that \mathbf{p} gives revenue at least t .

To do so, we construct a feasible price vector \mathbf{p}'' for H that produces revenue $t'' \geq t'$ and agrees with \mathbf{p} on all old nodes. For each old node $v \in H$ such that $\mathbf{p}'_v \neq \text{val}(v)$, we set $\mathbf{p}''_v = \perp$, and for all new nodes v' that are in a star with v we set $\mathbf{p}''_{v'} = \text{val}(v') = \text{val}(v)$. This way we increase the revenue by at least 1 without sacrificing feasibility. Now the revenue extracted using \mathbf{p}'' on H is at least t' and a total of at most $n_1 + 4n_2 + 9n_3$ is due to new nodes. That is, the revenue extracted from old nodes in H using \mathbf{p}'' is at least $t' - (n_1 + 4n_2 + 9n_3) = t$. Since \mathbf{p}'' agrees with \mathbf{p} on all old nodes, the revenue extracted using \mathbf{p} on G is at least t as well. \square

7. A Generalization to Multi-Demand Users

So far, we have always assumed that each node has demand for only one copy of the product. A natural generalization is to consider multi-demand users who are interested in receiving a certain number of copies if the price is affordable. For example, someone might want to buy either a certain number of licenses of a video game (because she wants to play the game with her friends) or no license at all. This would correspond to a type of *inelastic multi-unit demand* in the terminology of auctions. Assume again that there is enough supply of copies to satisfy all the demand. Then there is a natural way to generalize single-value revenue functions to capture such simple scenarios.

A revenue function $R_v(\cdot)$ is called a *multi-demand single-value revenue function* if there exist an integer s_v (the number of copies demanded) and a value $\text{val}(v)$ such that:

$$R_v(p_v) = \begin{cases} s_v p_v, & \text{if } \text{val}(v) \geq p_v; \\ 0, & \text{if } \text{val}(v) < p_v. \end{cases}$$

The intuition here is the same as for the single-value revenue functions.

Given a multi-demand single-value revenue function for each node, the objective is again to find a feasible price vector \mathbf{p} that maximizes the total revenue. We call this problem *Multi-Demand Inequity Aversion Pricing*. As this is a generalization of Inequity Aversion Pricing, it is immediate that any negative result for the latter yields the same negative result for the former. In particular, by Theorems 6.1 and 6.7, Multi-Demand Inequity Aversion Pricing is NP-hard and APX-hard for the corresponding edge constraints.

Quite surprisingly, we also prove that when the number of demanded copies is polynomially bounded, there is a strict reduction from Multi-Demand Inequity Aversion Pricing to Inequity Aversion Pricing. This directly implies that any approximation factor achieved for the latter is also achieved for the former. Therefore, we establish that the two problems are equivalent in terms of approximability. Note that the theorem holds for general edge constraints.

Theorem 7.1. *Let q be any polynomial. There exists a strict reduction from Multi-Demand Inequity Aversion Pricing with demands bounded by $q(n)$ to Inequity Aversion Pricing.*

Proof. Suppose we are given an instance I of Multi-Demand Inequity Aversion Pricing, i.e., a graph $G(V, E)$, an edge restriction function $\alpha(\cdot, \cdot)$, and for each node v her valuation $\text{val}(v)$ and her demand s_v . We are going to construct an equivalent instance I' of Inequity Aversion Pricing. The reduction creates s_v copies of v for each $v \in V$ and connects them to each other to create a clique K_{s_v} . Edges inside such a clique have $\alpha = 0$. For every edge $(u, v) \in E$ all the edges between the vertices of the u -clique and the v -clique are added with the same restrictions as the original edge. Let $G' = (V', E')$ be the resulting graph. If $s_{\max} = \max_{v \in V} s_v$ then we have $|V'| \leq ns_{\max}$ and $|E'| \leq (n + m)s_{\max}^2$.

We use OPT' and OPT to denote the optimal revenue of this instance and of the original, respectively. Our goal is to show that for any price vector \mathbf{p}' for I' we can efficiently find a feasible price vector \mathbf{p} for I with such that $\frac{R(\mathbf{p})}{\text{OPT}} \geq \frac{R'(\mathbf{p}')}{\text{OPT}'}$. We begin by proving that $\text{OPT} = \text{OPT}'$.

Claim 7.2. *An optimal price vector \mathbf{p}' for I' sets the same price for all vertices inside each v -clique.*

Proof of Claim. Note that $\alpha = 0$ inside each v -clique, so all these vertices have the same common price p'_v or \perp . If there were x, y in a v -clique such that $p'_x \neq \perp \wedge p'_y = \perp$ then by setting $p'_y = p'_x$ we would obtain a new feasible price vector for I' that gives greater revenue than \mathbf{p}' . This would contradict the optimality of \mathbf{p}' . \square

By Claim 7.2, we directly obtain a feasible solution for I with revenue OPT by setting p_v equal to the common price from the v -clique. Therefore, $\text{OPT} \geq \text{OPT}'$.

On the other hand, each feasible price vector \mathbf{p} for I can be adopted as a feasible price vector \mathbf{p}' for I' with the same revenue. To see that, just set the same price $p'_{u_v} = p_v$ for each copy u_v of v in the v -clique of G' . All edge constraints are satisfied, so the solution is feasible, and it clearly gives the same revenue. By taking \mathbf{p} to be an optimal price vector for I , the above implies that $\text{OPT}' \geq \text{OPT}$. We conclude that $\text{OPT} = \text{OPT}'$.

Finally, we need the following.

Claim 7.3. *Each feasible price vector \mathbf{p}' for I' can be transformed into a feasible price vector \mathbf{p} for I with at least the same revenue.*

Proof of Claim. For each $u \in V$, if V_u is the set of vertices in the u -clique of G' , define $u^* = \arg \max_{x \in V_u} p'_x$. Then, set $p_u = p'_{u^*}$. Such a \mathbf{p} is feasible for I because for all $(u, v) \in E$, $\alpha(u, v) = \alpha(u^*, v^*)$, where v^* is any vertex in V_v , and the constraint $\alpha(u^*, v^*)$ is already satisfied by \mathbf{p}' . It is straightforward that $R(\mathbf{p}) \geq R'(\mathbf{p}')$. \square

For the price vector described in the proof of Claim 7.3, we have

$$\frac{R(\mathbf{p})}{\text{OPT}} \geq \frac{R'(\mathbf{p}')}{\text{OPT}} = \frac{R'(\mathbf{p}')}{\text{OPT}'},$$

which completes the proof. \square

It would be interesting to determine whether the hardness of the problem changes when the demands are not polynomially bounded, although such functions are not very realistic in our setting. Notice, however, that even in that case it is not hard to obtain a $1/H_{p_k}$ -approximation in polynomial time by using the best single-price solution. In fact, we would still have a $1/P_\tau$ -approximation as in Theorem 3.2.

8. Concluding Remarks

We studied a revenue maximization problem under inequity aversion for the natural class of single-value revenue functions. Apart from establishing the first hardness results for this class, we also derived approximation algorithms based on combinatorial and graph-theoretic tools, which improve the state of the art when the set of available prices is small. We find this to be a realistic setting as special price offers are usually small in number, derived by specific discount and promotion policies. Clearly, the most interesting open problem is to resolve the approximability for general prices. That is, can we close the gap between the $O(1/\min\{P_k, H_n\})$ -approximation and the APX-hardness? Exploring further models of negative externalities is another attractive direction that has not been given as much attention as the case of positive externalities.

Acknowledgments

G. Amanatidis and E. Markakis were supported by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES. K. Sornat was supported by the National Science Centre, Poland, grant numbers 2015/17/N/ST6/03684, 2018/28/T/ST6/00366 and by the Foundation for Polish Science (FNP) within the START programme. P. Fulla was supported by a Royal Society Research Grant.

References

- [1] H. Akhlaghpour, M. Ghodsi, N. Haghpanah, H. Mahini, V. S. Mirrokni, and A. Nikzad. Optimal iterative pricing over social networks (extended abstract). In *6th Workshop on Internet and Network Economics (WINE 2010)*, pages 415–423, 2010.
- [2] N. Alon, Y. Mansour, and M. Tennenholtz. Differential pricing with inequity aversion in social networks. In *14th ACM Conference on Economics and Computation (EC 2013)*, pages 9–24, 2013.
- [3] G. Amanatidis, E. Markakis, and K. Sornat. Inequity aversion pricing over social networks: Approximation algorithms and hardness results. In *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *LIPICs*, pages 9:1–9:13, 2016.
- [4] G. Ausiello, A. Marchetti-Spaccamela, P. Crescenzi, G. Gambosi, M. Protasi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.
- [5] S. Bhattacharya, J. Kulkarni, K. Munagala, and X. Xu. On allocations with negative externalities. In *7th Workshop on Internet and Network Economics (WINE 2011)*, pages 25–36, 2011.
- [6] V. Bilò, M. Flammini, G. Monaco, and L. Moscardelli. Pricing problems with buyer preselection. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, pages 47:1–47:16, 2018.
- [7] G. E. Bolton and A. Ockenfels. A theory of equity, reciprocity and competition. *American Economic Review*, 100:166–193, 2000.
- [8] Z. Cao, X. Chen, X. Hu, and C. Wang. Approximation algorithms for pricing with negative network externalities. *J. Comb. Optim.*, 33(2):681–712, 2017.
- [9] P. Crescenzi and L. Trevisan. On approximation scheme preserving reducibility and its applications. *Theory Comput. Syst.*, 33(1):1–16, 2000.
- [10] W. H. Cunningham. The optimal multiterminal cut problem. In *Reliability Of Computer And Communication Networks, DIMACS Workshop*, pages 105–120, 1989.
- [11] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

- [12] P. Domingos and M. Richardson. Mining the network value of customers. In *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001)*, pages 57–66, 2001.
- [13] E. Fehr and K. M. Schmidt. A theory of fairness, competition and co-operation. *Quarterly Journal of Economics*, 114:817–868, 1999.
- [14] M. Flammini, M. Mauro, and M. Tonelli. On fair price discrimination in multi-unit markets. In *27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 247–253, 2018.
- [15] M. Flammini, M. Mauro, and M. Tonelli. On social envy-freeness in multi-unit markets. *Artif. Intell.*, 269:1–26, 2019.
- [16] D. Fotakis and P. Siminelakis. On the efficiency of influence-and-exploit strategies for revenue maximization under positive externalities. *Theor. Comput. Sci.*, 539:68–86, 2014.
- [17] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.
- [18] A. Goldberg, J. Hartline, A. Karlin, M. Saks, and A. Wright. Competitive auctions. *Games and Economic Behavior*, 55(2):242–269, 2006.
- [19] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.
- [20] N. Haghpanah, N. Immorlica, V. S. Mirrokni, and K. Munagala. Optimal auctions with positive network externalities. *ACM Trans. Economics and Comput.*, 1(2):13:1–13:24, 2013.
- [21] J. Hartline, V. S. Mirrokni, and M. Sundararajan. Optimal marketing strategies over social networks. In *17th International Conference on World Wide Web (WWW 2008)*, pages 189–198, 2008.
- [22] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.
- [23] S. Khanna, R. Motwani, M. Sudan, and U. V. Vazirani. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28(1):164–191, 1998.
- [24] J. Kleinberg. Cascading behavior in networks: Algorithmic and economic issues. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 24, pages 613–632. Cambridge University Press, 2007.
- [25] S. Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15(2):307–309, 1974.