Abstract—In this paper we study envy free pricing problem in general graphs where there is not a seller in every graph’s nodes. We assume unique establishment cost for initiating a store in each node and we wish to find an optimal set of nodes in which we would make the maximum profit by initiating stores in them. Our model is motivated from the observation that a same product has different prices in different locations and there is also an establishing cost for initiating any store. We consider both of these issues in our model: first where should we establish the stores, and second at what price should we sell our items in them to gain maximum possible profit. We prove that in a case of constant price our problem is NP-Hard and we present a $(1-\frac{1}{e})$-approximation algorithm for solving “Equal prices-Equal costs” and “Equal prices-Difference costs” versions of this problem.

I. INTRODUCTION

Consider you own a chain store and wish to build some branches in a city. You are well aware of your customers’ locations, the transportation cost and also customers’ valuations for the selling items. Now with this information in your hand you should first decide where to establish the stores and second at what price you should sell the items in each branch in order to maximize your revenue. You should also consider branch’s establishment cost into account. The point that we wish to make in this paper is pricing schemes in some situations related to the locations of the stores. This is the flavor of the pricing problems studied in this paper.

Envy-free pricing captures the notion of fairness of equilibrium pricing in economics (for related works, see, e.g., [10], [11] and the references within), and has recently received much attention in computer science [3]–[9], [12].

In fact algorithmic pricing is the computational optimization problem that sellers (e.g., in supermarkets) face when trying to set prices for their items in order to maximize their profit while they are aware of demands. Guruswami et al. [12] was the first who proposed this problem and give logarithmic approximations (in the number of consumers) for the unit-demand and single-parameter versions where there is a specific set of consumers and their valuations for items are known precisely. Subsequently several versions of the problem were discussed like tollbooth pricing, item pricing for single-minded bidders and pricing with restricted valuation ( [14]–[16]).

In [12] an $O(\log n)$-approximation algorithm was provided. Briest [5] recently showed that given appropriate complexity assumptions (the hardness of the balanced bipartite independent set problem in constant degree graphs or refuting random 3CNF formulas), the envy-free pricing problem can not be approximated within $O(\log n)$ for some $\epsilon > 0$. For the multi-unit demand setting, Briest [5] showed that the problem is hard to approximate within a ratio of $O(n^\epsilon)$ for some $\epsilon > 0$, unless $NP \subseteq BPTIME(2^{n^{\epsilon}})$.

When consumers desire a fixed subset of items (i.e., consumers are single-minded), a logarithmic approximation algorithm was derived in [12] and an almost tight lower bound was provided by Demaine et al. [9]. A few special cases of single-minded demand, such as the tollbooth problem where consumers desire paths in a graph, were studied in [6], [12]. Balcan and Blum [4] studied the graph vertex pricing problem where each consumer requests the two endpoints of an edge in a given graph and the goal is to set prices on vertices to maximize the total revenue. The graph vertex pricing problem has a similar flavor to our model, for which a 4-approximation algorithm was given in [4]. The best result which has been ever proposed for one the specific version of this problem was [17]. In [17] they assume that the valuation of the $i$’th consumer for $j$’th item - $v_{i,j}$ - forms a metric space, they solved their proposed problem exactly in polynomial time by reducing it to an instance of weighted independent set on a perfect graph.

Other pricing schemes (min-buying, max-buying, or rank-buying, where consumers buy an item with the smallest price, highest price, or highest ranking according to their preference) were studied in [3], [7], [13], where different algorithmic and lower bounds results were given.

In this paper, we consider establishment cost into account, that means that we can not have stores in all graph’s nodes. We want to find the stores locations and the items constant price that maximize the seller’s profit. In our model each consumer has a location and valuation for different items - $v_{i,j}$ - which is the maximum amount she would be willing to pay for the $j$’th item. In this case if she buys this item at price $p_j$, her profit will be $v_{i,j} - p_j$ (obviously she will buy the item only if its profit is positive for her) we want to find an envy free pricing scheme for the items in chosen locations that maximize seller’s profit, which is a natural assumption. We study this problem in two specific cases: 1.“Equal Prices-Equal Costs” 2.“Equal prices-Different Costs”. In case 1, the input establishment costs for all locations are equal and we are forced to give equal prices for all chosen locations. Case 2 is similar to case 1 except

---

Envy Free Chain Store Pricing

MohammadAmin Fazli  
Department of  
Computer Engineering  
Sharif University of Technology  
Tehran, Iran  
Email: fazli@ce.sharif.edu

Nastaran Nikparto  
Department of  
Computer Engineering  
Sharif University of Technology  
Tehran, Iran  
Email: nikparto@ce.sharif.edu

Morteza Saghafian  
Department of  
Mathematics and Computer Science  
Sharif University of Technology  
Tehran, Iran  
Email: saghafian@mehr.sharif.edu
that establishment costs can vary in each location. We give
\((1 - \frac{1}{e})\)-approximation algorithm for each of these cases.

II. OUR MODEL

Suppose that we have \(n\) locations and we want to sell an
item over these locations. There are \(k_i\) customers at \(i\)th
location. And also suppose that establishing a chain at \(i\)th
location costs \(c_i\). Customers of \(i\)th location have valuation
\(v_{i,j}\) over buying the item from \(j\)th location. For example their
valuations may be associated with their distance from that city.

We want to find a subset of cities \(S\) and establish our stores
there, then propose a price \(\pi(j)\) for our item in \(j\)th store such
that maximizes our profit. Customer \(i\) buy the item from one
the cities of \(S\) like \(k\), that maximizes \(v_{i,k} - \pi(k)\). If this
maximum value is negative then he will discard buying.

More formally we define \(u_i\) as the maximum possible utility
of the customers at location \(i\) that is \(\max_{j \in S}(v_{i,j} - \pi(j))\).
And also suppose that this maximum occurs at index \(t_i\). The
profit that we can get from each customer of \(i\)th location is defined as :

\[
p_i = \begin{cases} 
0 & \text{if } u_i < 0 \\
\pi(t_i) & \text{if } u_i \geq 0 
\end{cases}
\]

So we want to find \(S \subseteq \{1, \cdots, n\}\) and \(\pi : S \to \mathbb{R}\) that
maximizes:

\[
\sum_{i=1}^{n} p_i - \sum_{i \in S} c_i 
\]

In other sections of this paper, We consider various types of
this problem from Algorithmic and complexity point of view.

III. EQUAL PRICES-EQUAL COSTS

In this section, we take the version of this problem into
consideration in which establishment of all the chains cost
equal and we are supposed to propose equal prices for our
item in each of these stores (i.e. \(c_1 = c_2 = \cdots = c_n = C\)
and \(\pi\) must be a constant function ).

So our goal is to determine a subset \(S\) of the cities and a
price \(\Pi\) that results in following profit:

\[
\max_{S,\Pi} \sum_{\pi(S) \subseteq \Pi} k_i,\Pi - C,|S| 
\]
in which \(v_i(S) = \max_{j \in S} v_{i,j}\).

At first we prove that this problem is NP-Hard by reducing
to MDS\(^1\) to our problem.

**Theorem 3.1:** Suppose that we are given a constant
price \(\Pi\). The problem of choosing the best set \(S\) in “Equal
prices-Equal costs” chain store pricing is NP-Hard.

**Proof:** Assume that we are given a graph \(G\) with vertex
set \(V\) and we are supposed to find a MDS in it. Consider
an instance of the “Equal prices-Equal costs” chain store pricing
in which each \(k_i = 1\), \(\Pi = C = |V| - 1\) and \(v_{i,j} = |V| - d_{i,j}\)
where \(d_{i,j}\) is the distances between vertices \(i\) and \(j\) in \(G\).
Suppose that the maximum profit set chosen for this instance
of the problem is \(S_{OPT}\). We would establish a chain on all
the cities in \(S_{OPT}\). Consider the vertices of \(G\) whose distance
to \(S_{OPT}\) is exactly 1. Name these vertices \(B\). The vertices
which are in \(B \cup S_{OPT}\) would buy the item. Because their
distance to \(S_{OPT}\’s\) stores is within 1, we know that \(v_{i,j}\) for
\(i \in S_{OPT} \cup B\) and \(j \in S_{OPT}\) is greater than \(|V| - 1\) and then
\(v_{i,j} \geq 0\). So the total profit will be:

\[
|S_{OPT} \cup B|,\Pi - |S_{OPT}|,C = (|V| - 1)|B|
\]

If we add each member of \(V - (S_{OPT} \cup B)\) to \(S_{OPT}\) this
profit wouldn’t change because adding that member to \(S_{OPT}\)
costs \(C\) which is equal to \(\Pi\), the profit we can gain by selling
the item to it. So without loss of generality we can assume that
\(V - (S_{OPT} \cup B) = \emptyset\). Now we can assume that \(S_{OPT}\) is
a dominating set (See Figure 1).

With these preliminaries we claim that \(S_{OPT}\) is a MDS.
For a case of contradiction suppose that it is not and there is a
dominating set \(D\) in \(G\) with less than \(|S_{OPT}\) vertices. Since
\(|D| < |S_{OPT}|\), We can establish our chains on \(D\) vertices
and then we can earn the profit of \((|V| - 1),(|V| - |D|) >
(|V| - 1),(|V| - |S_{OPT}|) = (|V| - 1)|B\) which contradicts
the optimality of choosing \(S_{OPT}\).

In the next step, we propose an \(O(1)\)-approximation algo-
rithm for this simplified version of the problem.

**Lemma 3.2:** Suppose that we have \(n = |V|\) sets
\(A_1,\cdots,A_n\). The function \(f : 2^{1,\cdots,n} \to N\) defined by
\(f(A_1,A_2,\cdots,A_k) = \sum_{i \in \bigcup_{j=1}^{k} A_j} k_p\) is submodular.

**Proof:** Assume that \(S \subseteq T \subset \{1, \cdots, n\}\) and \(x \notin T\).
\(f(S \cup \{x\}) - f(S) = \sum_{p \in (A_x - \bigcup_{i \in S} A_i)} k_p\)

And also we have

\[
f(T \cup \{x\}) - f(T) = \sum_{p \in (A_x - \bigcup_{i \in T} A_i)} k_p
\]

Since \(S \subseteq T\) we have \(\bigcup_{i \in S} A_i \subseteq \bigcup_{i \in T} A_i\). So we have
\(f(T \cup \{x\}) - f(T) \leq f(S \cup \{x\}) - f(S)\).

**Suppose** that we have fixed a price \(\Pi\) and a set \(S\). Define
\(A_i(\Pi)\) to be the set \(\{j | v_{j,i} - \Pi \geq 0\}\). These vertices will
buy the item, but not necessarily from \(i\). Then our final profit

\footnote{Minimum Dominating Set}

![Figure 1. \(S_{OPT}\) is a MDS](image-url)
will be $\Pi_1|\bigcup_{i\in S} A_i(\Pi)| - C_i|S| = \Pi_1f(S) - C_1|S|$. From the previous lemma we know that $f$ is submodular. More than that this function is trivially monotone. We have the following famous theorem about optimizing the monotone and submodular functions:

**Theorem 3.3:** ([11], [2]) For a non-negative, monotone submodular function $f$, let $S$ be a set of size $k$ obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let $S^*$ be a set that maximizes the value of $f$ over all $k$-element sets. Then $f(S) \geq (1 - \frac{1}{e})f(S^*)$; in other words, $S$ provides a $(1 - \frac{1}{e})$-approximation.

The algorithm proposed in [2] to find the best $S$ with $k$ members is a kind of greedy hill climbing algorithms. This algorithm gets the desired target set size ($|S| = k$) and desired price(II) as inputs and gives one of the most profitable target set with a good approximated profit. We call the output of this algorithm $S^*(k, \Pi)$ but first we should limit the set of possible price values.

**Lemma 3.4:** The best constant price ($\Pi$) is in set $W = \{v_{i,j}|1 \leq i,j \leq |V|\}$

**Proof:** Suppose that $a = v_{i,1} \leq v_{i,2} \leq \cdots \leq v_{i,|V|^2}$ is a sorted list of the $W$'s members. If $v_{i}^{'} < \Pi < v_{i+1}^{'}$ we can increase $\Pi$ to $v_{i+1}^{'}$ and earn more money.

So we have enough tools to express the main result of this part.

**Theorem 3.5:** There is a $(1 - \frac{1}{e})$-approximation algorithm for solving “Equal prices-Equal costs” version of the chain store pricing.

**Proof:** The algorithm 1 is a desired algorithm. Its correctness is simply deduced from 3.4 and 3.2 and theorem 3.3.

---

**Algorithm 1** “Equal Prices-Equal Costs” chain store pricing

1. for each $\pi \in W$ (Lemma 3.4) and each $1 \leq k \leq |V|$ do
   2. Calculate $S^*(k, \pi) - C_1k$ and save the most profitable one as $k^*$ and $\pi^*$.
3. end for
4. return $\pi^*$ and $S^*(k^*, \pi^*)$.

---

### IV. Equal Prices-Different Costs

In this section we verify the case in which all the prices must be the same (ie. $\pi$ is a constant function) but establishment costs can be unequal. In this case we want to find the following:

$$\max_{S, \Pi} \left\{ \sum_{i \in S} k_i \Pi - \sum_{i \in S} c_i \right\}$$

The hardness of this new problem can be easily resulted from theorem 3.1 just by setting $c_i = |V| - 1$ for all $1 \leq i \leq |V|$. The main result of this section is that adding this degree of freedom to problem does not violate the submodularity of the profit function for a fixed price $\Pi$.

**Lemma 4.1:** Consider the function $f$ defined in lemma 3.2. The function $g(S) = f(S) - \sum_{i \in S} c_i$ is also submodular.

**Proof:** Suppose that $c(S) = \sum_{i \in S} c_i$. Then for $S \subseteq T$ we have $c(S \cup \{x\}) - c(S) = c_x = c(T \cup \{x\}) - c(T)$. So $g(S \cup \{x\}) - g(S) = f(S \cup \{x\}) - f(S) - c_x \leq f(T \cup \{x\}) - f(T) - c_x = g(T \cup \{x\}) - g(T)$. That completes the proof.

So then again we can find one of the most profitable greedy hill climbing algorithm with the same sense of the previous section.

**Theorem 4.2:** There is a $(1 - \frac{1}{e})$-approximation algorithm for solving “Equal prices-Different costs” version of the chain store pricing.

---

### V. Conclusion

In this paper, we presented a general model for chain store pricing and solved some specific versions of this problem by giving hardness results and approximation algorithms for them. Specifically, we prove that in a case of constant price our problem is NP-Hard and we present a $(1 - \frac{1}{e})$-approximation algorithm for solving “Equal prices-Equal costs” and “Equal prices-Difference costs” version of this problem. The results of this paper can be extended in future works by considering more parameters such as different prices, marketing phases and other related factors.

### Acknowledgment

The authors would like to thank Amir Rahmati for his useful comments during the preparation of this paper.

### References
