Dynamic pricing of durable products with heterogeneous customers and demand interactions over time

Onur Kaya *

Department of Industrial Engineering, Koc University, Eng 206, Sariyer, Istanbul 34450, Turkey

A R T I C L E   I N F O

Article history:
Received 8 September 2012
Received in revised form 12 February 2013
Accepted 20 May 2013
Available online 28 May 2013

Keywords:
Pricing
Stochastic dynamic program
Heterogeneous customers
Interdependent demand

A B S T R A C T

In this study, we analyze a dynamic pricing problem in which the demand is interdependent over time and the customers are heterogeneous in their purchasing decisions. The customers are grouped into different classes depending on their purchase probabilities and the customer classes evolve over time depending on the demand realizations at every period, which are a function of the prices set by the company. To decide on the optimal prices at every period, we model this problem using a stochastic dynamic program (SDP) and we develop several approximation algorithms to solve this SDP since the size of the state space of the SDP makes the optimal solution almost impossible to find. We present the efficiencies of the heuristics and provide managerial insights through a computational study in which we compare the revenues obtained with each heuristic with an upper bound value that we find on the optimal revenues.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Revenue management and dynamic pricing is a widely used concept in various industries to improve revenues and also widely studied in literature (e.g. Bitran & Mondschein, 1995; Gallego & van Ryzin, 1994; Hall, Kopalle, & Pyke, 2009, etc.). Bitran and Caldentey (2003), Elmaghraby and Keskinocak (2003) and McGill and Van Ryzin (1999) present surveys on dynamic pricing and revenue management. In most of the dynamic pricing literature, it is often assumed that demand is independent across periods and that price levels in the previous periods do not affect the demand in the future. However, in industries like the durable goods industry, in which the products have a fairly long life, the demand structure can be a little different. In such industries, customers can postpone their purchase decisions and when they make a purchase, they want to use it for some time before renewing it. Thus, the demand in a period affects the demand in the future periods and is affected by the demand in the previous periods. In a population, if the demand is high in a certain period, these customers will not renew their products for a certain amount of time and during this time period, the demand will be relatively low. However, after a certain time period, as these products get older, these customers want to renew them, leading to an increase in demand. In such a market, customers will have different purchasing structures depending on their past purchases and thus a heterogeneous group of customers exist in the market.

In this study, we consider different customer classes depending on their purchase probabilities and we aim to make the pricing decisions based on the state of the market. We divide the customers in the market into different classes and we define the state of the market as the number of customers in each class. We aim to improve the company revenues by dividing the customers into different classes and considering the state of the market explicitly in setting the prices. We use the previous purchases of the customers and the demand values in the previous periods to determine the different classes. Note that a customer who already has a product might have a different willingness to purchase a new product than a customer who does not have that product. In addition, a customer with a relatively old product at hand might have a higher willingness to make a purchase than a customer with a relatively new one at hand. It is observed in reality that a very low portion of the customers renew their products when their product is relatively new. However, more and more customers are seen to renew their products as they get older. For example, when a customer buys a new laptop, she wants to use it for some time and will not renew it for a certain amount of time, however, as time passes, this customer’s willingness to renew his laptop will increase. In addition to laptops, pricing of items like durable goods, cell phones, luxury items, several types of vehicles, etc. can be considered in the focus of our study.

In this paper, we model the dynamic pricing problem considering the heterogeneity of the customers and the interdependency of demand over time using a stochastic dynamic program. There is a wide area of research on dynamic pricing in the literature (e.g. Chatwin, 2000; Gallego & van Ryzin, 1994, 1997; Feng & Xiao, 2000; Lin, 2004; Lin & Sibdari, 2009; Zhao & Zheng, 2000). In these
In this study, we analyze the pricing strategy of a company selling a single product in a market with heterogeneous customers. We consider a discrete time model and try to determine the optimal sales price levels depending on the state of the market at every period. We assume that the company makes pricing decisions at discrete time epochs $t = t_1, t_2, \ldots, T$. Even though, we consider a finite horizon model, we note that an infinite horizon model can be approximated by a choice of large enough $T$. We let $p_i$ denote the price at period $t$ and $c$ denotes the unit production cost of an item. We consider $n$ different classes of customers in the system and let $M_{it}$ denote the number of individuals in class $i \in \{1, 2, \ldots, n\}$ at period $t$ such that $\sum_{i=1}^{n} M_{it} = P$ where $P$ denotes the total population size which is assumed fixed. In our model, we form the customer classes according to their willingness to make a purchase. We assume that a customer's willingness to make a purchase increases with the time from her last purchase and thus we use the age of the products that the customers have as an estimator for the willingness of the customers to make a purchase at the current period. We use the first $n-1$ classes for the customers with an existing product at hand where $i = 1$ denotes the customers with the newest products and $i = n-1$ denotes the customers with the oldest products. We use class $i = n$ for the customers who does not own any product. For example, when we consider the cell phone market, the people who made a purchase in the last 3 months can be considered as the first class, people who made their last purchase 3–6 months ago can be considered as the second class and the customers who have not made any purchase in the last 5 years (assuming that a cell phone has a maximum life of 5 years) can be considered as class $n$ customers. Depending on their last purchase time, each class has a different willingness to make a new purchase; the customers with relatively newer products have a lower willingness to renew their products and the customers with relatively older products have a higher willingness to make a new purchase.

Observe that in our model, demand is interdependent over time such that the prices in a certain period do not only affect the demand in that period but also affect the demand in the future periods by changing the market structure through class sizes. Therefore, the demand interdependency is modeled by tracking the number of customers $M_{it}$ in each class $i$ at time $t$. Similar to the model in Sibdari and Pyke (2010), we assume a fixed size population even though we allow new arrivals to the system (first time customers) and departures from the system (due to various reasons such as death, and termination of the need for the product). We assume that a person in class $i$ leaves the system with probability $\rho_i$, and the random variable $E_i$ denotes the number of people in class $i$ who leave the system in period $t$. Note that $E_i$ has a binomial distribution with parameters $M_{it}$ and $\rho_i$. We let $A_t$ denote the number of new arrivals to the system in period $t$ for the stability of the system we assume that $A_t = \sum_{i=1}^{n} E_i$ such that the total market size is fixed.

We let $\gamma(p)$ and $\gamma(p)$ denote the probabilities that a customer in class $i$ buys a new product from our company, and from one of our competitors, respectively, when we set the price to $p$. In this study, we assume that the prices of the competitors are fixed since our focus is not on competition between the companies and we consider competitive model as a future extension of this study.
since it requires a different analysis. Then, $1 - \alpha_i(p) - \gamma_i(p)$ denotes the probability that a customer in class $i$ does not make any purchase in period $t$. We let $Q_t$ denote our companies' demand from the people in class $i$, and $Q_t$ denotes the total market demand from class $i$. Note that $Q_t$ is a random variable with a binomial distribution with parameters $M_t$ and $\alpha_i(p)$ and $D_t$ is a random variable with a binomial distribution with parameters $M_t$ and $\alpha_i(p) + \gamma_i(p)$. We also note that, in some cases, when a customer renews her product, she can salvage the old one at a certain price $s_p$. Such a case can easily be absorbed into our model by adjusting the purchase probabilities appropriately for each class $i$, to reflect the utility that a customer would get when she renews his product by salvaging the old one.

In our model, when a person makes a new purchase at period $t$, she moves to class $1$ in the next period, however, when she does not make a purchase or does not leave the system in period $t$, she may move to the next class or stay in her current class in the next period depending on the time length between successive decision epochs $\Delta t$ and the structure of the groups. For example, in the cell phone example above, when the groups are formed with 3 month differences, if the pricing decisions are done every month, in the cell phone example above, when the groups are formed with 3 month differences, if the pricing decisions are done every month, then from 1 month to the next, a random customer in class $i$ moves to class $i+1$ with an approximate probability of $1/3$ (if she is at the last month of the 3-month period) and stays in her current class with an approximate probability of $2/3$ (if she is at the first 2 months of the 3-month period). We note that these transition probabilities are not exact since the ages of the customers in a group that covers a 3-month period are not known exactly, however, it is reasonable to assume that the ages of the customers are approximately equally spread throughout the group. We use $\beta_i$ to denote the approximate probability that a person in class $i \in \{1, 2, \ldots, n-1\}$ moves to the next class when she does not make a purchase assuming that she does not leave the system in period $t$. Then the random variable $B_{it}$ denotes the number of customers moving from class $i$ to $i+1$ at the end of period $t$. A list of the symbols used in the paper is presented in Table 1.

In this study, we aim to maximize the total discounted profit of the company over a planning horizon of $T$ periods by deciding on the prices $p_t$ that the company should set at the beginning of every period $t = 1, 2, \ldots, T$ in a dynamic manner such that the price $p_t$ is decided at the beginning of period $t$ based on the state of the system at that time. In this system, the price $p_t$ set at time $t$ not only affects the sales in the current period but also affects the future states of the system and thus the future demand. Thus, we need to consider both the immediate revenue and the future revenue in the system when deciding on the prices $p_t$ that should be set at every period $t$. The objective function in our problem can be written as below where $\delta$ denotes the discount factor and $Q_t$ depends on the price $p_t$ and the state of the system at that time.

$$\max_{p_t} \sum_{t=1}^{T} \left( p_t - c \right) \sum_{i=1}^{n} E\left[ Q_t^i \right] + \delta E[V_{t+1}(M_{t+1})]$$

(2.1)

In this study, we use a discrete time stochastic dynamic programming (SDP) model to determine the optimal sales price $p_t$ depending on the state of the system at period $t$. We denote the state of the system in period $t$, using the number of customers in class $i$, $M_t$. The state transition from one period to the next will be as follows:

$$M_{t+1} = M_t + \sum_{i=1}^{n} D_t - E_t - B_{it}$$

$$M_{t+1} = M_t - D_t - E_t - B_{it} + A_{it}$$

Note that, from one period to the next, the number of customers in class $i \in \{2, 3, \ldots, n-1\}$ will decrease by the number of customers who buy a new product, $D_t$, who leave the system, $E_t$, and who move to the next class, $B_{it}$, and it will increase by the number of customers who move from the previous class, $A_{it}$. Similarly, the number of customers in the first class increases by the amount of demand from other classes and decreases by the number of people who leave the system or move to the next class. The number of customers in the last class decreases by the number of people who make a purchase in that period or leaves the system and increases by the new arrivals to the system and the arrivals from the previous class.

Then, we aim to solve the following SDP algorithm as $t \to \infty$:

$$V_t(M_t) = \max_{p_t} \left[ \left( p_t - c \right) \sum_{i=1}^{n} E\left[ Q_t^i \right] + \delta E[V_{t+1}(M_{t+1})] \right]$$

(2.2)

Table 1

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>Indexes different customer classes</td>
</tr>
<tr>
<td>$t$</td>
<td>Indexes discrete time epochs</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of different customer classes</td>
</tr>
<tr>
<td>$p$</td>
<td>An array denoting all the prices set by all the different companies for period $t$</td>
</tr>
<tr>
<td>$c$</td>
<td>Unit production cost of an item</td>
</tr>
<tr>
<td>$\alpha_i(p)$</td>
<td>Probability that a customer in class $i$ makes a purchase from our company at time $t$ for a given price array $p$</td>
</tr>
<tr>
<td>$\gamma_i(p)$</td>
<td>Probability that a customer in class $i$ makes a purchase from one of the competitor companies at time $t$ for a given price array $p$</td>
</tr>
<tr>
<td>$s_p$</td>
<td>Salvage price of a product in class $i$ at period $t$</td>
</tr>
<tr>
<td>$\beta_i$</td>
<td>Probability that a customer in class $i$ leaves the system at period $t$</td>
</tr>
<tr>
<td>$E_i$</td>
<td>Random variable denoting the number of customers in class $i$ leaving the system in period $t$</td>
</tr>
<tr>
<td>$D_i$</td>
<td>Random variable denoting the number of customers in class $i$ who buys a new product in period $t$</td>
</tr>
<tr>
<td>$A_i$</td>
<td>Random variable denoting the number of customers in class $i$ who moves to the next class at the end of period $t$</td>
</tr>
<tr>
<td>$B_i$</td>
<td>Random variable denoting the number of customers moving from class $i$ to $i+1$ at the end of period $t$</td>
</tr>
<tr>
<td>$Q_i$</td>
<td>Random variable denoting our company's demand at period $t$ from customers in class $i$</td>
</tr>
<tr>
<td>$u_i$</td>
<td>A value denoting the difference between the utility levels of having a new product and having a product of class $i$</td>
</tr>
<tr>
<td>$v_i$</td>
<td>Weighted average of the $u_i$ values such that $v_i = \sum_{i=1}^{n} \frac{u_i}{M_i}$</td>
</tr>
<tr>
<td>$S_t$</td>
<td>Sales amount $k$ periods ago</td>
</tr>
<tr>
<td>$w_m$</td>
<td>Probability of a customer using a product for a duration of $m$ periods before renewing it</td>
</tr>
<tr>
<td>$g_k$</td>
<td>Probability of a customer who bought her product $k$ years ago has a product of age $i$ now</td>
</tr>
<tr>
<td>$p_2$</td>
<td>Price level of the competitor company</td>
</tr>
<tr>
<td>$\alpha_i$</td>
<td>A parameter specifying the heterogeneity of the purchase probabilities among the customer classes through their utility functions such that $a_i = di$</td>
</tr>
<tr>
<td>$b_i$</td>
<td>Price sensitivity of the utility function for the customers in class $i$</td>
</tr>
</tbody>
</table>
where \( M_t = [M_{1t}, M_{2t}, \ldots, M_{nt}] \) is an array with elements denoting the number of customers in each class \( i = 1, 2, \ldots, n \).

Even though we can use the SDP Eq. (2.2) to find the optimal prices when \( n \) is small, it becomes almost impossible to use the Eq. (2.2) in its current form to determine the optimal prices as \( n \) increases since the size of the state space in this SDP equation becomes extremely large as \( n \) gets larger. In addition, the randomness in the problem requires the consideration of all possible demand realizations at any state which makes the problem even harder for large \( n \). Because of these reasons, we need efficient heuristics in order to solve this problem and to approximate the optimal prices. In the next section, we explain these heuristics and analyze their efficiency in the numerical results section.

3. Approximation algorithms and bounds

In the SDP Eq. (2.2), since there are \( n \) customer classes and each of them takes a value between 0 and \( P \) such that their sum makes the population size \( P \), there are \( \text{comb}(P + n - 1, P) \) possible states which make the problem intractable even for moderate values of \( n \). In addition, the randomness of demand in the problem makes the solution even harder since we need to consider all possible combinations of demand realizations from different classes in order to evaluate the right hand side of the SDP Eq. (2.2). In this section, we develop several approximation algorithms to overcome these difficulties and to approximate the optimal prices efficiently, depending on the state of the market and we also find an upper bound on the optimal objective value of this problem.

3.1. Decomposition heuristic

The main difficulty in solving the optimality equation in (2.2) arises from the fact that we have to set the same price \( p_t \) for all the customers in different classes. However, if different prices could be set for different classes, then the problem can be decomposed into \( n \) individual problems which is much easier to solve. In this section, we develop a solution method, that we call decomposition heuristic (DH), that is based on the idea of relaxing the requirement that the same price should be charged to all classes and setting different prices for different classes. With this relaxation, we can analyze each class separately from each other and the state space for each problem decreases substantially. Then using the value functions obtained from this relaxation, we develop an approximate dynamic programming model to approximately determine the optimal prices for the original problem. We let \( p_t \) denote the price for class \( i \) at period \( t \). Then, we can rewrite the optimality Eq. (2.2) as follows:

\[
V_t(M_t) = \max \sum_{i=1}^{n} (p_{t,i} - c)E[Q_i^t] + \delta E[V_{t+1}(M_{t+1})] \quad \text{s.t.} \quad p_{t,i} = p_t, \quad p_t \geq 0
\]  

(3.1)

Observe that in the feasible set of solutions (3.1)–(3.3), constraints (3.2) links the prices for different classes. This suggests relaxing these constraints and solving the relaxed dynamic program

\[
V_t^p(M_t) = \max \sum_{i=1}^{n} (p_{t,i} - c)E[Q_i^t] + \delta E[V_{t+1}^p(M_{t+1})] \quad \text{s.t.} \quad p_{t,i} \geq 0
\]  

(3.3)

Then, the following proposition shows that the optimality Eqs. (3.4) and (3.5) decomposes by the customers.

**Proposition 1.** For all \( i = 1, 2, \ldots, n \), if \( v_t(e_i) \), where \( e_i \) is the \( n \)-dimensional unit vector with a 1 in the element corresponding to \( i \), is defined as

\[
v_t(e_i) = \max \{ (p_{t,i} - c)x_{it} + \delta [p_{t,i}v_{t+1}(e_i) + (1 - p_{t,i})(x_{it} + \gamma_{it})v_{t+1}(e_i) + (1 - \gamma_{it})v_{t+1}(e_{t+1})] \}
\]  

(3.6)

then, we have

\[
V_t^p(M_t) = \sum_{i=1}^{n} M_{it} v_t(e_i)
\]  

(3.7)

**Proof.** We show the result by induction over the time periods. It is easy to show the result for \( t = 0 \). Assuming that the result holds for time period \( t - 1 \) for \( t \geq 1 \):

\[
V_t^p(M_t) = \max_{p_{t,i} \geq 0} \left\{ \sum_{i=1}^{n} (p_{t,i} - c)E[Q_i^t] + \delta E[V_{t+1}^p(M_{t+1})] \right\}
\]  

(3.8)

Using the separable structure of the classes, we can solve the above problem more efficiently by focusing on one class at a time. Note that there are \( \text{comb}(P + n - 1, P) \) possible states in the dynamic program (2.2), however we can solve the Problem (3.4) and (3.5) with only \( n \) possible states using the equations in (3.6). Eq. (3.6) determines the optimal price when there is a single class of customers and Eq. (3.7) determines the relaxed value function \( V_t^p(M_t) \) by taking a weighted sum of individual optimal value functions \( v_t(e_i) \), thus \( V_t^p(M_t) \) provides an upper bound for \( V_t(M_t) \) as stated below.

**Corollary 1.** \( \sum_{i=1}^{n} M_{it} v_t(e_i) \) provides an upper bound for \( V_t(M_t) \).

**Proof.** Since \( V_t^p(M_t) \) is a relaxation of \( V_t(M_t) \), \( \sum_{i=1}^{n} M_{it} v_t(e_i) = V_t^p(M_t) \geq V_t(M_t) \)

When we implement the decision rule given in (3.6), we can obtain \( V_t^p(M_t) \) which gives us an upper bound on the optimal value of \( V_t(M_t) \). Note that in this solution, we allow different prices to be set
for different classes, which is infeasible for the original Problem (2.2). However, using the approximate value functions $V^*_t(M_t) = \sum_{t=1}^{n} M_t v_t(e_t)$ in the right hand side of the SDP Algorithm 3.9, we approximate the optimal $p_i$ values that we need to set at any state $M_t$ and we also obtain feasible approximate values $V^*_t(M_t)$ for $V_t(M_t)$. Thus we use the following equation, which is easy to solve, to approximate the optimal values as $t \to \infty$.

$$V^*_t(M_t) = \max_{p_i} \left( p_i - C_0 \sum_{i=1}^{n} E[Q_i] + \delta E[V^*_{t+1}(M_{t+1})] \right)$$

$$= \max_{p_i} \left( p_i - C_0 \sum_{i=1}^{n} E[Q_i] + \delta E[M_{t+1} \mu_i(e_t)] \right) \quad (3.9)$$

In the numerical results section, we present the efficiency of this approximation procedure using a simulation of the system. We note that using a Lagrangian relaxation approach by assigning Lagrange multipliers for each relaxed constraint and adding it to the objective function might give better results than the total relaxation of the constraints. However, such an approach does not help the solution of this problem since Proposition 1 is not valid when Lagrange multipliers are used in the SDP equation, except the case when all Lagrange multipliers are equal to 0 which leads to the same result as above. In order for Proposition 1 to be valid with Lagrange multipliers, there needs to be a Lagrange multiplier for each possible state in the original problem which makes the problem intractable again and does not help the solution of the problem.

3.2. Weighted average pricing heuristic

In this section, we propose a novel approximation algorithm for the optimal prices, that we call the weighted average pricing heuristic (WAP), that is based on taking the weighted averages of the optimal prices for each customer class that are found in the previous subsection through the DP Eq. (3.6). Let $p_i$ denote the optimal price for a customer in class $i$ that is found through the Eq. (3.6) as $t \to \infty$. Then, using these individual optimal prices, at any state $M = (M_1, M_2, \ldots, M_n)$, we set the price $p$ as the weighted average of the individual optimal prices such that $p = \sum_{i=1}^{n} M_i \mu_i$. We present the efficiency of this algorithm in the numerical results section.

3.3. Representative customer heuristic

In this section, we propose another approximation algorithm to solve the SDP Algorithm 2.2 and we call this algorithm as the representative customer heuristic (RCH). To approximate the optimal prices, instead of keeping all the customer classes, we consider keeping and updating only the position of a representative customer in the whole population. In this heuristic, instead of keeping all the $M_i$ values as the state variables, we propose to keep only a single state variable, $\mu_i$, that reflects the average position of the customers in the market. In our model, $\mu_i$ gives an information about the distribution of the customers to different classes at period $t$. In this heuristic, we use a parameter $u_i$ for each class $i$ that is related to the different purchase probabilities of the people in different classes and $\mu_i$ denotes the weighted average of the $u_i$ values such that $\mu_i = \frac{\sum_{i=1}^{n} M_i u_i}{\sum_{i=1}^{n} M_i}$. For this model, we think of $u_i$ as the difference between the utility levels of having a new product and having a product of class $i$, however, some other definitions for $u_i$ can also be used depending on how the purchase probabilities differ among classes. Note that the customers in the first class will have the lowest $u$ value since they have the newest products and the customers in the last class will have the highest $u$ value. Then, we update the state variables from one period to the next, using the $u_i$ values, demand values, exits from the system and new arrivals to the system as follows:

$$V_t(M_t) = \max_{p_i} \left( p_i - C_0 \sum_{i=1}^{n} E[Q_i] + \delta E[M_{t+1} \mu_i(e_t)] \right) \quad (3.10)$$

where $\mu(e_t)$, which is a function of both $p$ and $\mu_i$, is the purchase probability of the representative customer in the market. In the above model note that $\mu_i$ takes continuous values, however for computational purposes in our numerical results section, we discretize the continuous range of $\mu_i$ and use discrete values with step sizes equal to 0.01 in the state space of the above problem and round the $\mu_i$ values to their nearest discrete values. Keeping $\mu_i$ instead of all $M_i$ values gives an approximate idea about the real state of the system. Using the above DP algorithm, we obtain approximate values for the optimal prices and we present the efficiency of this algorithm in the numerical results section.

3.4. Myopic pricing policy

As a benchmark policy, we also consider the simple myopic pricing policy (MP) and according to this policy we set the prices considering only the profits that can be obtained in the current period without considering its future effects. We use this heuristic to analyze the efficiencies of the other heuristics that we propose above compared to this simple policy and to analyze the improvements that can be obtained by considering the future periods and the interdependency of demand across periods in addition to the current period profits. In this heuristic, at any state $M_t$, we solve the following problem to decide on the approximate prices $p_i$ at time $t$.

$$\max_{p_i} \left( p_i - C_0 \sum_{i=1}^{n} E[Q_i] \right)$$

4. Estimation of the initial state

Note that, the initial state of the market needs to be known in order to use the above DP formulation and the heuristics. Thus we need to estimate the number of people in each class at the initial state of the market. Observe that the number of people in each class at the initial state is related to the number of purchases in the previous periods. For example the number of people who bought a product in the last period will belong to the first class in the next period and the customers who leave the system in period $t$ (which is equal to the number of new arrivals to the system, $A_t$) will belong to class $n$ in the next period. All the other customers will pass a period with their current products and their products will become a period older in period $t+1$. Thus, we multiply $D_t$ with $u_i$, $A_t$ with $u_n$ and $P - D_t - A_t$ with $(\mu_i + \Delta u)$ where $\Delta u$ denotes the decrease in the utility level of a customer when she moves from one period to the next without renewing his product. Then the DP equation will be as follows:

$$\mu_{t+1} = \frac{D_t u_i + A_n u_n + (P - D_t - A_t) (\mu_i + \Delta u)}{P}$$

Note that the customers who buy a new product at period $t$, $D_t$, will belong to the first class in the next period and the customers who leave the system in period $t$ (which is equal to the number of new arrivals to the system, $A_t$) will belong to class $n$ in the next period. All the other customers will pass a period with their current products and their products will become a period older in period $t+1$. Thus, we multiply $D_t$ with $u_i$, $A_t$ with $u_n$ and $P - D_t - A_t$ with $(\mu_i + \Delta u)$ where $\Delta u$ denotes the decrease in the utility level of a customer when she moves from one period to the next without renewing his product. Then the DP equation will be as follows:

$$\mu_{t+1} = \frac{D_t u_i + A_n u_n + (P - D_t - A_t) (\mu_i + \Delta u)}{P}$$
periods before renewing it, which can be easily obtained through the company data or a customer survey. Then, assuming that she has not left the system, we let $g_i$ denote the probability that a customer, who bought her product $k$ years ago, has a product of age $i$ now and by the help of $w_0$ values, $g_i$ can be calculated through the solution of the following recursive relations

$$g_k = \begin{cases} \sum_{j=1}^K w_j g_{k-j} / P(m \geq i) & \text{for } k = i + 1, i + 2, \ldots, K \\ 0 & \text{for } k = i \end{cases}$$

(4.1)

As a result, the number of products of age $i$ today can be estimated through the relation $M_i = \sum_{j=1}^K w_j (1 - \rho^j) g_j$ where $\rho$ is the probability that a customer leaves the system at any period.

5. Numerical results

In this section, we aim to analyze the efficiency of the heuristics for this problem, analyze the effect of different parameters and extract some managerial insights on the optimal pricing policies through numerical analysis. We consider a stationary system, thus we drop the subscript $t$ from the parameters from this point on for clarity.

5.1. Input parameters for the computational study

In our computations, similar to Sibdari and Pyke (2010), we use the multinomial logit (MNL) model to describe the customer choice behaviors and to find the probabilities of the customers making a purchase. In our problem, the customers in different classes have different choices and purchasing probabilities depending on their utilities from the purchase and the MNL model is commonly used in the marketing literature to model customer choice in settings similar to our problem, especially when the customers face a discrete choice of actions (see Franses & Paap, 2001; Lilien, Kotler, & Moorthy, 1992). In this model, it is assumed that customers act independently from each other to maximize their own utility. We assume a linear utility function such that if a customer in class $i$ decides to buy product $j$ at price $p$, then his or her utility is equal to $U_{ij} = a_i - b_j p_i + Z_{ij}$ where $a_i$ denotes the increase in the utility of a customer in class $i$ by renewing his product, $b_j$ denotes the price sensitivity of the utility function and $Z_{ij}$ is an independent and identically distributed random variable with Gumbel distribution, used to reflect the individual preferences of the customers (see Ben-Akiva & Lerman (1985) for detailed information on the MNL models). We assume without loss of generality that the expected utility of a customer who does not buy any product is $E[U_{i0}] = 0$ and we let all companies have identical products, thus we can use $a_y = a_x$, $b_y = b_x$ for all $j$ and $Z_0$ follows the same distribution for all $j$, thus we drop the subscript $j$ and use $Z_i$ from now on. In the MNL model, the probability of a customer in class $i$ choosing product $j$ can be calculated from the following closed form equation:

$$q_{ij}(p) = P(U_{ij} = \max_{k=0 \ldots K} U_{ik}) = \frac{e^{a_i - b_j p_i}}{1 + \sum_{k=1}^K e^{a_k - b_j p_i}}$$

Using the above equation, we determine the values of $a_i(p)$ and $\gamma_i(p)$ that denote the purchase probabilities of the customers from our company and from the competitors depending on our price $p$, which are exactly equal to $q_{ij}(p)$ values with appropriate $j$ indices. We also note that, in some cases, when a customer in class $i$ renews his product, she can salvage the old one at a certain price $s_i$. Such a case can easily be absorbed into our model by adjusting the utility function of class $i$ appropriately to include the salvage value as $U_{ij} = a_i - b_j(p_i - s_i) + Z_i = a_i + b_j s_i - b_j p_i + Z_i$. Thus adjusting the value of the parameter $a_i$ as $a_i = a_i + b_s$ and using $q_i$ values in the analysis will be sufficient to cover the case with salvage values of the used products.

In this study, we use the parameters $c = 5$, $b_i = 1$ for all $i \leq n - 1$, $s_i = 0$, $\rho_1 = 0.01$, $b_i = 0.5$, $a_i = \theta i$ for all $i$ and $\theta = 0.5$ specifies the heterogeneity of the purchase probabilities among the customer classes. Note that increasing $i$ values denote the customers with increasingly old products and thus the utility of a customer in class $i$ is higher than the utility of a customer in class $j < i$ when they both renew their products at the same price. As the value of $\theta$ increases, the customer classes become increasingly distinct from each other. We note that all the heuristics in our computational study are coded in Matlab R2011b and the simulation runs are performed on a workstation with an Intel (R) Core (TM)2 Duo processor, 2.53 GHz speed, and 2 GB of RAM.

5.2. Performances of the heuristics

In our base case analysis, we consider a single company without any competition and use the parameters stated as above. For this base case, we design a simulation of the system using 1000 periods and a discount factor $\delta = 0.99$. In this simulation, the prices at each period are set according to the heuristics that are defined in Section 3 in order to approximate the optimal pricing policy. Note that, in order to solve the maximization problem in each heuristic, we use a one dimensional search over $p$ to find the prices at every period of the simulation run depending on the state of the market. We note that, even though theoretically $p$, can take any continuous value, in our computational study, we discretize the price values in the possible price range with a step size of 0.1 and pick the best prices among these values. In Table 2, using a population size of $P = 1000$ and $P = 5000$, we present the ratios of the profits obtained with each heuristic over an upper bound of the profits for each simulation run, where the upper bound value for the profits is obtained through the formulation as explained in Section 3.1, in which we allow different prices to be set for different classes. We compare the efficiencies of the heuristics for different numbers of classes $n$, where each value in the table shows the average of 100 simulation runs. We note that, for experimental purposes, we also increased the number of simulation runs up to 500, however, we observed that the average results did not change much and thus we present all our results based on the average of 100 simulation runs. We see that all the heuristics can solve pretty big instances in a very short time (less than a minute in all cases) while we were not able to solve those instances with the direct SDP formulation (2.2).

Even though the run times for all the heuristics are very short and close to each other, we observe that the heuristic $DH$ performs significantly better than all the other heuristics, especially as $n$ increases, and can provide more than 25% better results than the myopic pricing strategy. We see that the population size does not have a significant effect on the performances of the heuristics, however the number of classes can affect the performances significantly. When $n$ is low, all the heuristics perform very close to the optimal solution, but as the number of different classes increase, all heuristics’ performances get worse. We see that the heuristic $DH$ has an optimality gap of less than 7% when $n$ is less than or equal to 17, but when $n$ is 49, the optimality gap increases to about 15%. However, the performance of $DH$ is still much better than the other heuristics and performs about 10% better than the second best heuristic. When $n = 49$, the optimality gaps of $RCH$, $WAP$ and $MP$ are seen to increase to around 25%, 32% and 40% respectively. However note that the optimality gaps of the heuristics do not only originate from the performance of the heuristics but it also includes the gap between the optimal solution and the upper bound that we use in this study. Thus, the actual optimality gaps could be lower than the stated values in Table 2.
Note that the efficiencies of the heuristics depend on the system parameters. After further analysis, we observe that the values of \(\theta\) and \(p_2\) effect the system the most, while the other parameters do not have significant effects. Thus, in Table 3, using \(n = 17\) classes and \(P = 1000\), we change the values of \(\theta\) and \(p_2\), one by one, keeping all other parameter values as in the base case. Recall that as \(\theta\) gets larger, the customer classes become increasingly different from each other and the purchase probabilities of customers in different classes become more distinct. We observe in Table 3 that when \(\theta\) is small, all the heuristics perform better which is logical since the customer classes are not much different from each other and the optimal price does not depend much on the state of the system. As \(\theta\) increases, in general, the heuristics' performances worsen since the customer classes become increasingly different from each other and approximations can lead to high losses in profits. However, we also observe that the performance of the heuristic \(DH\) improves when \(\theta\) is increased from 1 to 5 which seems counterintuitive. This is due to the fact that when \(\theta\) is very high, the higher classes dominate the lower classes in terms of their revenue potential and just focusing on the last classes gives very good results. Insignificance of the lower classes leads to better performance of the heuristic \(DH\) for very high \(\theta\) values. In all cases in Table 3, we observe that the heuristic \(DH\) performs reasonably well even for high \(\theta\) values with an optimality gap of 7.5% in the worst case.

In Table 3, we also look at the effect of the competitor’s price \(p_2\) on the efficiencies of the heuristics by introducing a competitor into the system with price \(p_2\) while every other parameter is kept the same as in the base case. We observe that when the competitor's price is low, the heuristics have a better performance and this is due to the fact that when \(p_2\) is low, the optimal price for our company also tends to be low and there is less flexibility in the pricing decision which leads to similar prices for different states of the system. However, when \(p_2\) is high, the company has more flexibility in the pricing decision and the state of the system closely effects the pricing decision.

5.3. Analysis of prices set by the heuristics

In this section we analyze the prices set by the heuristics using the same parameters as stated in the previous section by simulating the system for 500 time periods with \(n = 17\) and \(P = 1000\). When we look at the prices set by the heuristics as seen in Fig. 1, we observe that the prices set by the heuristic \(DH\) are higher than the myopic prices, since myopic prices only consider the state of the soughe market in the current period and any customer that does not buy a product in the current period is seen to be lost and it has no effect to the future periods. However, when we also consider the future periods as in the heuristic \(DH\), if a customer does not buy a product in the current period, she will still be a potential customer in the next period with a higher willingness to pay, which leads to the prices higher in the current period since the risk of not making a sale is smaller than the myopic model. The prices set by the other two heuristics seem to be in between the \(DH\) prices and the myopic prices in general.

We also observe that the prices do not change much after the initial periods and fluctuate slightly around a certain value, when the population size \(P\) is high, which is in line with the results in Fleischmann et al. (2006). As stated intuitively, spreading demand equally across periods and smoothing the demand is the best one can do and in the long run, prices try to converge to a constant value to keep the market structure in the best possible state for all periods. This observation also suggests that a static pricing policy performs close to a dynamic pricing one. The reason for the somewhat constant pricing scheme to provide better results than making promotions with certain intervals and high price fluctuations between periods is that we consider only a single market segment with a single product in our model, we do not consider inventory and operational issues and in our examples we do not include any highly asymmetric market response to prices. In case of these considerations, promotions might be more useful than constant pricing, however we consider these cases as subjects of a future study.

In our simulations, when the population size \(P\) is high, small fluctuations in prices are observed due to the randomness in demand since the prices are a function of the market state. However, when \(P\) decreases, the price fluctuations become more significant as seen in Fig. 2 in which the population size \(P = 100\). The reason for this is that when \(P\) is high, due to law of large numbers, the difference between the actual demand realizations and the expected values are less significant, however when \(P\) is small, the randomness in demand has a higher effect on the market structure and thus the prices. As a result, dynamic pricing policy becomes much more beneficial than static pricing when the population size is small. After periods in which demand realizations are higher than expected, prices are seen to decrease since the market potential is

<table>
<thead>
<tr>
<th>(n)</th>
<th>(P = 1000)</th>
<th>(P = 5000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta)</td>
<td>MP/UB</td>
<td>RCH/UB</td>
</tr>
<tr>
<td>2</td>
<td>0.987</td>
<td>0.992</td>
</tr>
<tr>
<td>3</td>
<td>0.981</td>
<td>0.989</td>
</tr>
<tr>
<td>5</td>
<td>0.844</td>
<td>0.896</td>
</tr>
<tr>
<td>9</td>
<td>0.786</td>
<td>0.857</td>
</tr>
<tr>
<td>17</td>
<td>0.655</td>
<td>0.784</td>
</tr>
<tr>
<td>49</td>
<td>0.584</td>
<td>0.758</td>
</tr>
</tbody>
</table>
expected to be low after high demand periods. Similarly, after low demand periods, prices are seen to increase since the potential demand is expected to be higher in the next one. However, in reality, contrary to this situation, it is observed in some cases that when the demand for a product is high, the prices tend to increase in the next period since higher demands are also expected in the future periods. However this might be misleading if the increase in demand is due to periodical randomness rather than structural and permanent. Since we assume a fixed size population and do not consider structural changes in demand between periods and assume that the fluctuations in demand is due to pure periodical randomness, the prices in our model happen to be somewhat inversely proportional with the demand in the previous periods. Note that, when a higher (or lower) demand than expected is observed and if it is believed that the demand structure is changed at that period, we can update the purchase probabilities or the market size to reflect this structural change in demand and this might lead to an increase (or decrease) in prices similar to the observations in practice.

5.4. Effect of number of classes used in the heuristics

Note that in making the pricing decision, different number of classes can be used for grouping the customers. Thus, grouping the customers into different number of classes is also a decision that the marketing managers can focus on. Ideally, to match the real system exactly, all customers should be considered as a separate class, however such a grouping might not be feasible or possible to implement. In addition, even though we can mimic the real system better by using a high number of classes, as seen above, using high number of classes lead to higher optimality gaps for the heuristics. In addition, using more classes would lead to low numbers of customers in each class and estimating the system parameters for each class, such as the number of people in each class or the utility parameters, becomes much more difficult. Thus, for a given population, we analyze the effect of using different numbers of classes in Table 4.

In this example, we consider the pricing of laptop computers which is assumed to have a useful life of 4 years. First, we group
the customers, who made a purchase previously, with 1 month differences and thus we divide the customers into 49 classes according to the age of their products. Class number 1 denotes the customers who bought their laptops in the last month, class 2 denotes the customers who bought their laptops 1–2 months ago and so on. We consider the people who have not made any purchase in the last 4 years in class 49. Then, instead of using \( n = 49 \) classes, we use less number of classes to group the customers. Instead of using 1 month differences between classes, when we group the customers with 3 month periods, we will have \( n = 17 \) classes in which class 1 denotes the customers with products of age less than 3 months, class 2 denotes the customers with products of age between 3 and 6 months and so on. Class 17 again denotes the set of people who have not purchased any product in the last 4 years. Observe that when \( n = 17 \), each class from 1 to 16 will be a combination of 3 classes in the case \( n = 49 \). Note that, when we move from \( n = 49 \) classes to \( n = 17 \) classes, we also need to update the parameters of the system to reflect the effect of combining 3 classes into one. For example, we use the parameters \( a_i = -\beta_i \) with \( \beta_i = 0.5 \) for all \( i \), and \( \beta_i = 1 \) for all \( i < n - 1 \) when \( n = 49 \). However, when \( n = 17 \), we use the adjusted values \( \beta_i = 1/3 \) since the pricing periods are still taken as 1 month while each class covers 3 months and thus only one third of the customers in each class will move to the next one on average at each month. In addition, we take \( a_i = 2\theta + 30(i - 1) \) for all \( i \leq 16 \) and \( a_{17} = 49\theta \) since the last classes are the same for both \( n = 17 \) and \( n = 49 \) cases while three classes are combined into one for the other classes. All the other parameters will be the same for \( n = 17 \) and \( n = 49 \) cases. We also consider the grouping of the customers into \( n = 9 \), \( n = 5 \), \( n = 3 \) and \( n = 2 \) classes in addition to \( n = 17 \) and \( n = 49 \) classes. When \( n = 9 \), the customers will be grouped with 6 month differences and the 9th class again denotes the customers who have not made any purchase in the last 4 years. All the other classes are obtained from the \( n = 17 \) case by combining the customers in two consecutive classes into one. We obtain the \( n = 5 \), \( n = 3 \) and \( n = 2 \) cases similarly and update the parameters accordingly.

In Table 4, we present the ratios of the profits obtained with the heuristic DH using different numbers of classes \( n \) over a common upper bound. In this example, we use the upper bound for \( n = 49 \) case, \( UB_{49} \), as the common upper bound for all \( n \) values, since this upper bound has the highest value among all \( n \) cases and it reflects the highest profits that can be obtained in the market. We observe that using too few classes such as 2 or 3 classes for grouping the customers can lead to revenues that are significantly lower than what could be obtained and using more classes help to improve the revenues. However, after a certain value, increasing the number of classes even further does not increase the revenues significantly and can even lead to lower profits due to the decrease in the performance of the heuristic for large \( n \) values. We observe that when \( \theta \) is small (i.e. if the customers in different classes are similar to each other), using 9 classes gives the best result and dividing the customers into more classes does not improve the revenues. However as \( n \) increases, more classes are needed and using 17 classes give the best result when \( \theta = 5 \). When we look at the effect of the competitor’s price \( p_2 \), we see that as \( p_2 \) gets lower (i.e. if the market is more competitive), less number of classes will be enough since the company does not have much choice in its pricing decision and using different number of classes does not effect the solution very much. However, if \( p_2 \) is high (i.e. the market is not very competitive), the number of classes have a more significant effect on the revenues and using a higher number of classes are required.

### 6. Conclusion

In this study we analyze the revenue management problem for the products with fairly long lives by explicitly consider different classes of customers with different willingness to make a purchase. Considering the demand interactions over time and the heterogeneity among the customers, we construct a stochastic dynamic programming model to determine the optimal prices. However, since the state space of the dynamic program can get very large as the number of classes increases, we present approximation algorithms to solve this problem and present the efficiencies of our heuristics through a computational study.

We observe that significant improvements can be obtained by considering the future effects of the prices and the interdependence of demand over time and it becomes especially important as more customer classes are considered in the system. We observe that the heuristic DH gives the best results and the dynamic pricing strategy considering the state of the market can provide more than 25% better results than the myopic pricing method which does not consider the effect of the pricing decisions to future periods. The performances of the heuristics seem to improve when the number of classes are small, the market is more competitive or the customer classes are similar to each other. In addition, we observe that as the customers are differentiated into more classes, higher profits can be obtained until a certain number of classes are used. However, using too many classes above a certain level does not improve the revenues significantly and can even lead to decreased revenues due to the decrease in the performances of the heuristics for high number of classes.

Of course, real world situations have many more and complex characteristics that are not captured by the model in this paper. For example, we assume that the prices of the competitors are fixed, however, in real life, the competitors can also adjust their prices and a competitive dynamic pricing model can be studied in the future as an extension of our study. Consideration of inventory (including operations and manufacturing costs) and different market segments in the pricing model can be other subjects to focus on in the future.

### References


