CS711: Introduction to Game Theory and Mechanism Design **Nov-Dec 2020**

Project Report: Pricing Strategies for In-App Purchases

Group 14: Bholeshwar Khurana, Mahajan Dipak Anil, Mudreka Arif, Rishab Agarwal, Sudhanshu Bansal

Abstract

This paper discusses various mechanisms for the pricing of In-app purchases in games/apps. We discuss the mechanism presently used, where the supply is infinite. We propose new mechanisms where the supply in a time frame is kept limited to ensure a greater revenue to the company. We also formulate a mechanism when the sale of one item is affected by other items or items are to be renewed every season. We did various simulations for all the proposed mechanisms and results are presented for the same.

1 Introduction and Motivation

Most of the mobile apps today use freemium model, i.e. they provide the product/app for free but charge money for upgraded service or in-app purchases (like in-game money, costumes, skins etc.). However, to date, there don't exist good pricing strategies for in-app purchases. A mere 5% of app users typically make an in-app purchase, while the in-app purchases account for 48.2% of the total revenue [Saleh, 2018].

This paper provides a mathematical formulation for the problem and proposes different pricing mechanisms for the in-app purchases. Simulations are done for the proposed strategies and their results are presented in the subsequent sections. Once a developer knows the optimal pricing strategy, they can optimize their revenues; hence this work could be of great use.

1.1 Related work

This is a novel problem as there doesn't exist much research related to pricing strategies for IAPs. [Bright-Black, 2019] discusses about various pricing systems like Static Pricing, Segmented Pricing and Dynamic Pricing. It mentions various points that a company should keep in mind while choosing their pricing strategies, but doesn't mention any specific strategies that a company should employ. [Kramer et al.] examines the implications of digitization on strategic and operational pricing decisions and shows examples from various industries and enterprises. [Huu et al.] proposes a model for computing the optimal dynamic pricing strategies for various cloud providers, we take help from their work to apply to our problem.

There have been psychological studies like [Dinsmore et al., Seok et al.], which examine the relationship between personality traits and mobile app purchasing tendencies. Quantitative Studies on InApp Purchases have been done in [Qiu et al.].

1.2 Brief overview of the report

We introduce the formal model in Section 2, prove our main result in Section 3, report our experimental findings in Section 4 and finally conclude in Section 5.

2 Formal model of the problem

We try to construct a basic model to represent scenario for a given IAP. We assume that there are n in-game items (numbered 1 to n) and the price of one item does not affect the price/demand of another item. Let $I = \{I_j; j \in \mathbb{N}, j \leq n\}$ be the set of items on the store. We assume that the price of the item I_j is set to a particular value P_j^t during the time interval $[t\delta, (t+1)\delta)$ where δ is a time frame. We can think of it as a seasonal price where the developers could change the price of the item per season of the game.

Let there are N users (numbered 1 to N) of the app, which is constant with respect to time and i^{th} user has a threshold price of $\mathcal{T}_{i,j}^t$ for item I_j in the time interval $[t\delta, (t+1)\delta)$, i.e. the price the user is willing to pay for the item, which is unknown to the developer. We can, therefore, define the utility for user i in the time interval $[t\delta, (t+1)\delta)$ as $U_i^t = \sum_{j=1}^{n} \alpha_{i,j}^t (\mathcal{T}_{i,j} - P_j^t)$ where $\alpha_{i,j}^t = 1$ if player i purchases item I_j during the time interval, otherwise 0. We assume that a user buys the item if his utility for the item is greater than or equal to 0.

We define the maximum number of copies of item I_j that can be bought (supply) in time interval $[t\delta, (t+1)\delta)$ as S_j^t. The reason behind this assumption is that a developer can keep the supply of the items limited in order to control the number of people purchasing the item. Also, define the motivation of the user i to purchase the item I_j in the time interval as $(\mathcal{T}_{i,j}^t - P_j^t)$. If the number of players demanding for a particular item is greater than the supply, then the item will be given to the players based on either (i) First come, First serve or (ii) to the players having the maximum value of motivation. The reason behind this is that the players having larger motivation to buy the item would be willing to purchase it as early as possible.

3 Main results/findings

We propose different mechanisms and run simulations for the same. Section 3.1 discusses the pricing strategy when the supply is infinite, which is true since the in-app purchases' supply can be made infinite. This is the mechanism followed by most of the apps today. In Section 3.2, we propose new mechanisms where the supply is made limited during a time interval $[t\delta, (t+1)\delta)$ and there is a fixed price during this interval. In section 3.3, we look at the case when one item's sale is affected by other items and the item is to be renewed every time frame, like Royale Pass in PUBG or Spotify Premium.

3.1 Unlimited supply

Theorem 1 [1] *If the supply is unlimited, then the developer has to put a constant price for each item.*

Using the above theorem, we see that the price of the item has to be constant when the supply is infinite. Note that we are not considering any seasonal discounts/sales. Most of the present day apps use this strategy, where there is a fixed price for an in-app item. Moreover, the optimal constant price (P_c) would be one, which maximizes $\mathcal{N}(\mathcal{T}, P_c) * P_c$ where $\mathcal{N}(\mathcal{T}, P_c)$ are the total number of users (with thresholds $\mathcal{T}$) who are willing to buy the item priced at P_c, i.e. the number of users whose threshold is greater than P_c. Hence, $P_c = argmax_{P_c}(\mathcal{N}(\mathcal{T}, P_c) * P_c)$. For a company, it is impossible to know the exact value of thresholds of the users, however they can make estimates by looking at past trends.

3.2 Limited supply

Since we saw that the developers have to keep the price constant in the case of unlimited supply, so they can't exploit the fact that some users can pay higher price for the items they like. For this we propose the strategy where the developers provide a constant supply for the items in each time interval (here the time interval could be a day). If they start with a high price, users with high thresholds would buy the item early and the developers can keep fluctuating the price and supply in the subsequent time intervals based on the demand of the items. Here we assume that the users would buy the item in the initial days even if the developers keep on decreasing the price in every subsequent time frame, since the supply is constant. We propose and simulate the following strategies. Note that we assume that each item is independent of the other items and hence, we propose the strategies for a single item which can easily be replicated for n items.

3.2.1 Each day's supply is constant while the price keeps on decreasing

Let P_0 be the initial price. Let S_0 be the supply which is same on every day. Let P_{min} be the minimum price at which the company can sell the item. Let ΔP be the decrement in price which the company employs on each day. Therefore, $P_t = max(P_{min}, P_{t-1} - \Delta P)$. However, in this strategy comes a psychological factor where users might not be willing to buy an item early knowing that the price always decreases on subsequent days.

3.2.2 Next day's supply and price depends on the previous day's demand

Let P_0 be the initial price and S_0 be the initial supply. Let P_{min} be the minimum price at which the company can sell the item. Let S_{max} and S_{min} be the maximum and minimum supplies respectively which the company can provide on any day. Let ΔP be the increment/decrement in price which the company would employ on each day and ΔS be that for the supply. Let $Sold_t$ be the number of users who bought the item on t^{th} day. We employ the following strategy:

[1]Proof in Appendix

$$S_t = \begin{cases} S_{t-1} + \Delta S & Sold_{t-1} \geq \gamma.S_{t-1} \\ S_{t-1} - \Delta S & Sold_{t-1} < \gamma'.S_{t-1} \\ S_{t-1} & otherwise \end{cases} \text{ and } P_t = \begin{cases} P_{t-1} + \Delta P & Sold_{t-1} \geq \gamma.S_{t-1} \\ P_{t-1} - \Delta P & Sold_{t-1} < \gamma'.S_{t-1} \\ P_{t-1} & otherwise \end{cases}$$

Note that S_t on any day t doesn't exceed S_{max} and is not less than S_{min}. Similarly, P_t is never less than P_{min} on any day. Here γ and γ' are the parameters which the company could choose to have a correlation between demand and supply.

3.3 Multiple Items and The Game Theory Problem

In the previous sections we assumed that we could extend the model from single item to multiple items if the sales of one item do not affect the sales of other items. Sometimes this might not be the case, we can have multiple items at the same time and depending on various factors, users could make choices of buying the items. Another assumption that we took earlier was that purchasing an item would give lifetime subscription to that item, but designers might come up with items that have limited period of subscription, so users have to re-buy an item. We release these assumptions now to formulate a new game in which there are N users (Numbered 1 to N) and n items (numbered 1 to n). We will follow formulation similar to [Huu et al.] to construct a game modelled as Markov Decision Process (MDP) whose solution would be Markov Perfect Equilibrium which would give us an optimal pricing strategy [Filar et al.].

To model the game as MDP, we need to define the states of the game. We define a state of the MDP as the state of the market - $\beta = (\beta_0, \beta_1, ..., \beta_n)$, where vectors $\beta_i = (\beta_{i,1}, ..., \beta_{i,N})$ and $\beta_{i,j} = 1$ if player j buys item I_i at that moment. Also the actions are price changes of items.

We introduced the following modifications - instead of maximizing for each item individually, we maximize the sum of revenues produced by each of them. Also, we do not need the constraint that everyone has to buy at least one item in a given interval. For this, we add a phantom item I_0 such that the price of this item is 0. So choosing this item is equivalent to not purchasing anything. So our assumption here is that in a time interval the user can buy at max 1 item. For computation feasibility, we assume that our prices of each item takes integer values between 0 and MAX_VALUE.

As the developer is the designer of the game, we can state the rewards that the developer gets equal to the revenue collected in that time interval. Particularly, revenue for item I_i can be written as $R_i^t = \sum_{k=1}^{N} \beta_{i,k} P_k^t$. So for discount factor γ we have to maximise value functions for the states,

$$\hat{V}(\beta, P) = \sum_{i=1}^{n} \hat{V}_i(\beta, P) = \sum_{i=1}^{n} \sum_{t=0}^{\infty} \gamma^t E[R_i^t | \beta^0 = \beta, P]$$

Now, from the perspective of the user (having a budget constraint c_k), we define utility $U_{k,i}$ as the utility of player k when she buys item I_i where $U_{k,i} = b_i - P_i + \eta_{k,i} = v_{k,i} + \eta_{k,i}$ where we say that b_i is the absolute benefit that a player gets after buying that item and $\eta_{k,i}$ is the preference of the player k on the item i. Comparing this to the older model that we used these two components combined defined the threshold price a player would set on the item to decide whether or not to buy that item. Here we have assumed that the threshold price is due to two parts one is the benefit of that item on absolute level and the other is the preference of the player for that particular item. Following [Huu et al.] we assumed that the η values are random variables from Gumbel distribution. Also, following the budget constraint we assume that the budgets B_i of user i are coming from exponential distribution with parameter λ. To calculate the probabilities that the player k chooses item I_i. over other items is given by

$$P_{k,i} = Prob(U_{k,i} \geq U_{k,i'}, \forall i' \neq i \, and \, P_i < B_i) = Prob(\eta_{k,i'} \leq \eta_{k,i} + v_{k,i} - v_{k,i'}) \times (1 - Pr(B_i \leq P_i)) = \frac{e^{v_{k,i}}}{\sum_{i'} e^{v_{k,i'}}} \times e^{-\lambda P_i}$$

We used [Train et al.] for manipulating the values to derive the final expression. Also, similar to [Huu et al.], we can define the transition probabilities $P(\beta'|\beta) = q(\beta'|\beta) / \sum_{\beta''} q(\beta''|\beta)$, where

$$q(\beta'|\beta) = \prod_{i=1}^{n} [\exp{(-\sum_{k=1}^{N} (\beta_{i,k} - \beta'_{i,k})^2)} \prod_{k=1}^{N} (\beta'_{i,k} P_{k,i} + (1 - \beta'_{i,k})(1 - P_{k,i}))]$$

Now our goal remains to maximize $V(\beta, P)$, for this, we use the Bellman equation [Bellman 1954]:

$$\hat{V}(\beta) = \max_{P \in \mathbb{R}_+^n} [\sum_{i=1}^{n} R_i^t + \gamma \mathbb{E}_{\beta'}[\hat{V}(\beta', P)]] \quad \hat{P} = \arg\max_{P \in \mathbb{R}_+^n} [\sum_{i=1}^{n} R_i^t + \gamma \mathbb{E}_{\beta'}[\hat{V}(\beta', P)]] \quad \mathbb{E}_{\beta'}[\hat{V}(\beta', P)] = \sum_{\beta'} P(\beta'|\beta)\hat{V}(\beta', P)$$

where $\hat{P}$ gives the optimal pricing strategy at the state β. The expected value can be calculated using transition probabilities.

Now, releasing further assumptions, if we allow for purchase of multiple item at a given time, then we just need some minor changes in the above setup. Let us define a constant ϵ such that a user purchases all the items such that $U_{i,k} \geq \epsilon$ given the budget constraints c_k, i.e. $\sum_{i=1}^{n} \beta_{i,k} P_i \leq c_k$. In case of budget violation, choose the ones which maximizes the utility function $\tilde{U_k} = \sum_{i=1}^{n} \beta_{i,k} U_{i,k}$ for a given user. The probabilities $P_{k,i}$ can be redefined as

$$P_{k,i} = Prob(U_{k,i} \geq \epsilon \, and \, P_i < B_i) = Prob(\epsilon \leq \eta_{k,i} + v_{k,i}) \times (1 - Pr(B_i \leq P_i)) = (1 - Prob(\eta_{k,i} \leq \epsilon - v_{k,i})) \times e^{-\lambda P_i}$$

$$P_{k,i} = (1 - F(\epsilon - v_{k,i})) \times e^{-\lambda P_i}$$

4 Experiments/Simulations

We ran various simulations of the mechanisms proposed above. We chose 10000 users and their thresholds $\mathcal{T}$. We sampled the thresholds (Figure 1) from an exponential distribution with $\lambda = 8$, as the Gartner's survey [Gartner, 2016] shows that the distribution looks like exponential distribution with 45% users spending 5$ on in app purchases. The lowest threshold is $5.8 * 10^{-5}$ and the maximum threshold is 71.67. We consider one item in each of the simulation, since we assume that the items are independent of each other. So, the utility of each user becomes $U_i = \mathcal{T}_i^t - P^t, i \in [1, 10000]$, where $\mathcal{T}_i^t$ is the threshold of the i^{th} user and P^t is the price of item during the time interval $[t, t+1)$. We chose the parameters: $S_0 = 300, S_{max} = 600, S_{min} = 60, \Delta S = 20$ and $P_0 = 25, P_{min} = 5$. For the "Fixed supply strategy" (Section 3.2.1), we chose $\Delta P = 1$. For "Supply dependent on demand strategy" (Section 3.2.2), we chose $\Delta P = 2, \gamma = 3/4, \gamma' = 1/4$. We used "First come, first server" method if the number of users interested to buy the item exceeded the supply. Figure 2 shows the prices, supplies and revenues plots for "Fixed supply strategy". Figure 3 shows the plots for "Supply dependent on demand strategy".

For Figures 2,3, we assumed that the thresholds remain constant with time. However, this might not be true since users may lose interest in items with time. For example Thor's skin in Fortnite would be trending in the initial days, but users might lose interest in it with time. To relax this assumption, we decrease thresholds with time. We use $\mathcal{T}^t = \mathcal{T} - 0.05t$, where $\mathcal{T}$ is the constant thresholds we used in the previous simulations. Figure 4 shows the plots for "Supply dependent on demand strategy" with these new thresholds.

For infinite supply, we found the optimal revenue using the strategy discussed in Section 3.1 and we found that the company receives maximum revenue at the price $P_c = 9$. Table 1 summarizes the results for the simulations and we see that the revenues obtained by our mechanisms are significantly higher than those obtained with the presently-used infinite supply model.

For multiple items case, as the size of problem increases, the time complexity increases exponentially. Due to lack of high computing resources we couldn't run the simulation for considerable size of the game. For smaller setups (like 3 items and 4 players), the results were not accurate, as the sample was not able to reflect the distribution.

5 Summary and Discussions

We looked at various pricing strategies and showed that our proposed strategies are better than the presently-used infinite-supply model. We looked at formulating the problem as Markov Decision Process if the sales of items are affecting each other. For future work, we could try optimizing the simulation of MDP and use continuous price model. Another scope for future work is the case when the number of users in the game are not constant with respect to time.

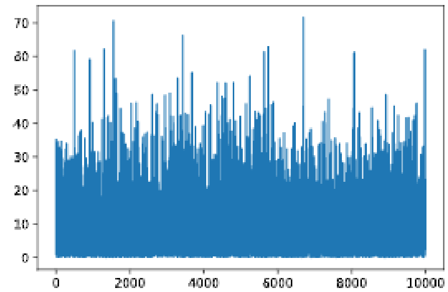

Figure 1: Thresholds of all the 10000 users

Strategy	Total Revenue
Infinite Supply	30618
Constant supply each day, Threshold: $\mathcal{T}$	64357
Supply dependent on demand, Threshold: $\mathcal{T}$	64156
Supply dependent on demand, Threshold: $\mathcal{T}(t)$	46249

Table 1: Total revenue obtained for each strategy

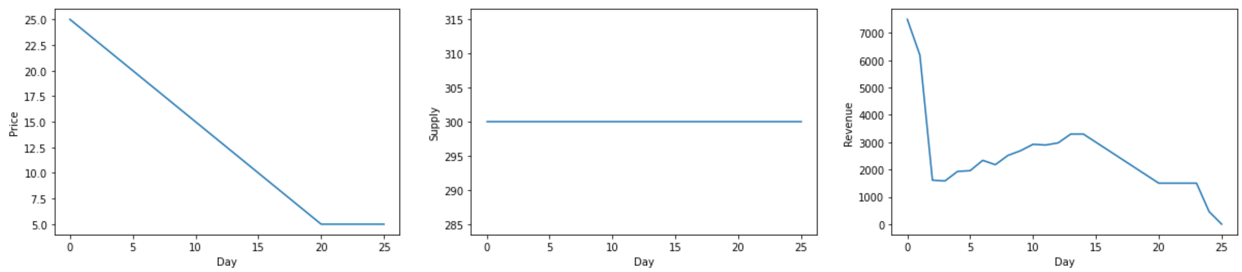

Figure 2: Plots for the "constant supply each day" strategy. Threshold: $\mathcal{T}$

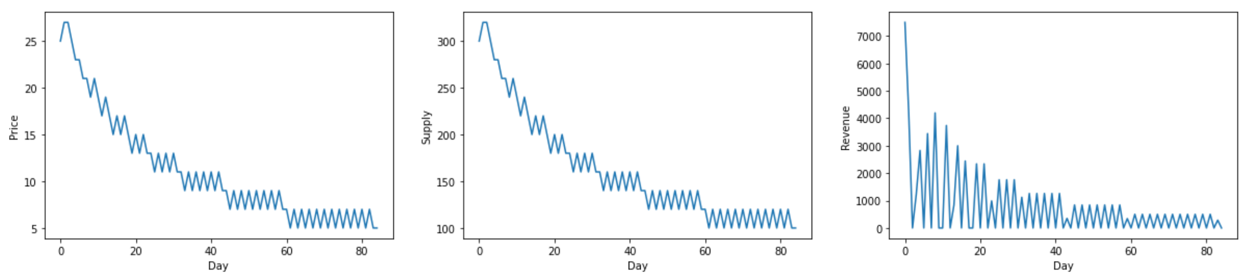

Figure 3: Plots for the "supply dependent on demand" strategy. Threshold: $\mathcal{T}$

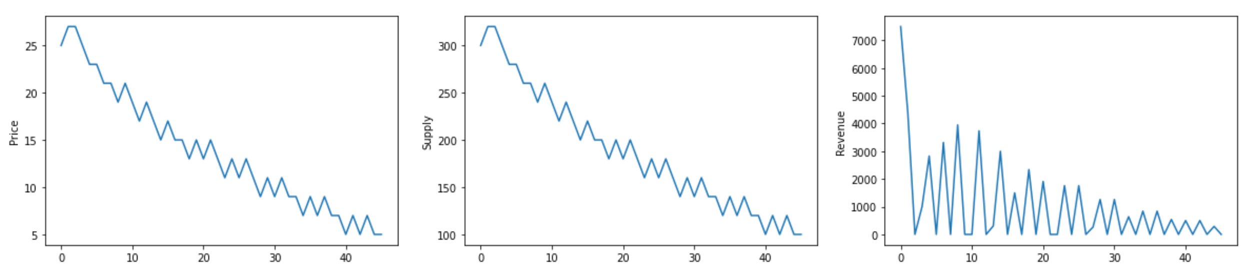

Figure 4: Plots for the "supply dependent on demand" strategy. Threshold: $\mathcal{T}^t = \mathcal{T} - 0.05t$

References

[Saleh, 2018] K. Saleh, "Global In-app Purchase Revenue Statistics and Trends". Retrieved from https://www.invespcro.com/blog/in-app-purchase-revenue.

[BrightBlack, 2019] , "How to price your In-app Purchases". Retrieved from http://brightblack.co/blog/how-to-price-your-in-app-purchase-items.

[Kramer et al.] Krämer A., Kalka R. (2017), "How Digital Disruption Changes Pricing Strategies and Price Models". In: Khare A., Stewart B., Schatz R. (eds) Phantom Ex Machina. Springer, Cham. https://doi.org/10.1007/978-3-319-44468-0_6.

[Huu et al.] Truong-Huu, Tram, and Chen-Khong Tham, "A game-theoretic model for dynamic pricing and competition among cloud providers." *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing. IEEE, 2013.*

[Dinsmore et al.] Dinsmore, J. B., K. Swani and R. G. Dugan. "To "free" or not to "free": Trait predictors of mobile app purchasing tendencies.". Psychology & Marketing 34.2 (2017): 227-244.

[Seok et al.] Seok S. and DaCosta B. (2015), "Predicting video game behavior: An investigation of the relationship between personality and mobile game play." Games and Culture, 10, 481-501.

[Filar et al.] J. Filar and K. Vrieze, "Competitive Markov Decision Processes". New York, USA: Springer-Verlag New York, Inc., 1996

[Qiu et al.] Qiu John X. , "Alternative Revenues: A Quantitative Study on InApp Purchases" (2014)

[Train et al.] K. E. Train, "Discrete Choice Methods with Simulation," Identity,vol. 18, no. 3, pp. 273–383, 200

[Bellman 1954] Bellman, Richard. "The theory of dynamic programming". Bull. Amer. Math. Soc. 60 (1954), no. 6, 503–515. https://projecteuclid.org/euclid.bams/1183519147

[Gartner, 2016] "Gartner Mobile App Survey Reveals 24 Percent More Spending on In-App Transactions than on Upfront App Payments". Retrieved from https://www.gartner.com/en/newsroom/press-releases/2016-05-26-gartner-mobile-app-survey-reveals-24-percent-more-spending-on-in-app-transactions-than-on-upfront-app-payments.

A Proof of Theorem 1

Proof: Suppose the price isn't constant.
Claim 2 *If the price isn't constant, then the developers have to decrease the price of the item in every subsequent time frame.*
Proof: Suppose the developers start with an initial price of P_0. All the users who have their thresholds greater than P_0 would have already bought the item on day 1. Now if the developers increase the price of item, i.e. $P_1 > P_0$ in the next time interval, then no user would buy the item since their threshold is already less than P_0, hence less than P_1. So, the developers have to decrease the price of the item in the next time interval. ■

Using Claim-2, we see that the developers decrease the price of the item in every subsequent time frame. We assume that all of the users are rational. We also assume that the users know this trend of prices decreasing in every time frame, which they could know by seeing the trend of other items or by seeing the trend in initial days. Hence, a rational user would not buy the item on day-1, knowing that the prices would keep on decreasing in every time interval. Hence, the developers have to keep a constant price. ■

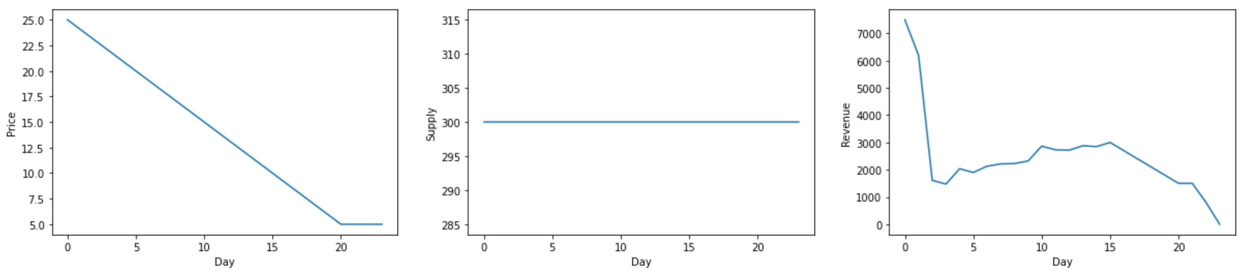

Figure 5: Plots for the "fixed supply" strategy. Threshold: $\mathcal{T}^t = \mathcal{T} - 0.05t$

B Additional Experiments

Figure 5 shows the plots for "Fixed supply strategy" when the thresholds vary with time. The total revenue obtained for this strategy is 59448.