THE BANDWIDTH YIELD MANAGEMENT PROBLEM

Salomón Wollenstein

Boston University
Division of Systems Engineering and Center for Information and Systems Engineering
15 Saint Mary's Street
Boston, MA 02215
www.bu.edu/se

May. 6, 2018

Technical Report No. SE-2018-2

This is the final report of the course project on Advanced Optimization Dynamic Programming and Stochastic Control (SE 710) at Boston University taught by Prof. Michael Caramanis.

Contents

List of Figures

List of Tables

1 Introduction

Yield Management Systems (YMs) (also known as *Revenue Management Systems*) are broadly used in service industries that offer perishable goods such as airline tickets, hotel rooms or car rentals. The main objective of these systems is to sell all the available inventory by maximizing the company's profits. In practice, YMs are generally variable pricing strategies that influence customers behavior.

Motivated by the *yield management problem*, it is natural to ask: how should communication companies create pricing strategies for networks such as the internet or telecommunications. This problem is commonly referred to as the *Bandwidth Yield Management Problem* and it is generally complex and multi-faceted. To solve this problem, service providers have to consider engineering (compatibility with protocols), financial (pay infrastructure loan) and Quality of Service (QoS) constraints, as well as simplicity and social welfare [3]. In this context, a central question arises when determining how prices should evolve (e.g. prices could stay fixed or might depend on parameters capturing the operating conditions) [4].

The objective of the *Bandwidth Yield Management Problem* is to select the price that maximizes the profit of a network provider (e.g. telecommunication, TV, internet) when demand and service time of the bandwidth are sensitive to price and capacity is finite. The pricing policy of network providers is important because it increases their income, allowing to recover their investments, pay their operational costs, and invest in expansions and new technologies. Additionally, it provides a competitive market for consumers willing to pay more for such service.

One of the interesting dynamics of this problem, in contrast with the classical *yield management problem*, is that customers arrive and depart in real-time having service times associated to them. Moreover, since the nature of these service do not have expiration date, we will analyze the long-term average formulation as opposed to a finite-horizon formulation for classical yield management.

In this project we will formulate the bandwidth yield management problem and will characterize some insights of its solution. In particular we will look at the scenario where there are multiple classes of customers $i = \{1, ..., M\}$, arrivals follow a *non-homogeneous Poisson process* $\lambda_i(u_i)$ as a function of the *connection price* $u_i(t)$ and service times follow an *exponential distribution* $\mu_i(c_i)$ as function of the *service price* $c_i(t)$.

Additionally, we present the formulation and results for the single-class case. In this formulation we assume that arrival rates are linear in connection processes, and that departure processes (service times) are quadratic in the service prices. With these assumptions we show that the problem can be solved by using a simple recursion starting when the differential cost (as specified in dynamic programming (DP) infinite-horizon problems) equals zero $h(0) = 0$. We also show that the average period cost $\bar{J}$ is quadratic on $h(n = i)$.

Finally, we present numerical results implemented in Matlab for two cases. The single-class, arrival process linear on connection cost and departure process quadratic on service cost; and the single-class with constant parameter on service times .

The rest of the report is organized as follows: In sections 2.1 and 2.2 we formulate the problem for the multi-class and construct its analytic solution using DP. Then, aiming to find a solvable instance the problem we present a uniformized version in section 2.3. Sections 3 and 4 show analytic and numerical results for the single-class problem with dynamic and static service processes depending on the pricing policy. Finally, in section 5 we state brief conclusions of this project.

2 Problem Formulation

2.1 Multi-class with prices on connection and service times

In this section, we formulate a model to find a pricing policy that maximizes the profit of the bandwidth provider. Let R be a fixed capacity of the bandwidth. We assume there are M classes of customers and classes $i = 1, ..., M$ arrive according to a *Non-homogeneous Poisson Process* depending on the connection price $\lambda(\mathbf{u}) = \{\lambda_i(u_i); i = 1, ..., M\}$. Upon arrival, each of these customers pays a fee u_i for connecting to the network. Once the customer has arrived it stays connected for a time interval which is exponentially distributed with parameter $\mu(\mathbf{c_i}) = \{\mu_i(c_i); i = 1, ..., M\}$. At departure the customers are charged a fee equal to $c_i(\mu_i)$.

Let $n_i(t)$ denote the number of clients i connected at time t. Then, we can define the state of the system to be $\mathbf{N} = \{n_i(t); i = 1, ..., M\}$. Notice that since $n_i(t)$ is a counting process it is discontinuous at arrivals and at departures. Finally, let r_i be the units of bandwidth used per customer type i. Note that a new client will only be accepted if the capacity of the bandwidth demanded is available, this is

$$\mathbf{N(t)}'\mathbf{r} + r_i \leq R \tag{1}$$

where $\mathbf{N(t)}$ denote the state of the system, $\mathbf{r}$ is the vector of the bandwidth required by each class; and r_i is the new arrival of class i. To continue the analysis we need to establish two important assumptions.

Assumption 1: there exists a maximum connection price $u_{i,max}$ for each customer class i. Beyond this price the demand $\lambda_i(u_i)$ becomes zero. Let $\lambda_{\mathbf{0}} = \{\lambda_{1,0}, ..., \lambda_{M,0}\}$ denote the vector of arrival rates when the cost is infinite.

Assumption 2: there exists a maximum service price $c_{i,max}$ for each customer class i. Beyond this prices, all customers leave the service and $\mu_i(c_i)$ becomes zero. Let $\mu_{\mathbf{max}} = \{\mu_{1,max}, ..., \mu_{M,max}\}$ denote the vector of service rates when the cost is at its maximum.

In the same manner, we can characterize a vector $\lambda_{\mathbf{0}} = \{\lambda_{1,0}, ..., \lambda_{M,0}\}$ to be the arrival rate when the connecting price is zero. Equivalently, we set $\mu_{\mathbf{0}} = \{\mu_{1,0}, ..., \mu_{M,0}\}$ to be the arrival rate when the service price is infinite.
With these definitions being set we can formulate a revenue maximization problem. It is important to mention that the system evolves as a *Continuous Markov Chain* with state $\mathbf{N(t)}$ and time-dependent pricing policies $\mathbf{u}(t)$ and $\mathbf{c}(t)$. For a small time step Δ there is a $\lambda_i(u_i(t))\Delta$ probability that an arrival of class i occurs. Conversely, there is a $\mu_i(c_i(t))\Delta$ probability that a departure of an active customer i occurs. Hence, the expected revenue for that Δ interval is equal to $\sum_i^m \Delta\lambda_i(u_i(t))u_i(t) + \Delta n_i(t)c_i(\Delta)$. Thus, the expected long-term average revenue equals

$$\lim_{T\to\infty} \frac{1}{T}\mathbf{E}\Big[\int_0^T \sum_{i=0}^{M} \lambda_i(u_i(t))u_i(t) + n_i(t)c_i(t)\Big] \tag{2}$$

$$\lim_{T\to\infty} \frac{1}{T}\mathbf{E}\Big[\int_0^T \lambda(\mathbf{u}(t))\mathbf{u}(t) + n\mathbf{c}(\Delta))\Big]$$

It is easy to show that the limit exists because for any pricing policy $\mathbf{u}(t)$ and $\mathbf{c}(t)$ the state $\mathbf{N} = \mathbf{0}$, corresponding to an empty system, is a recurrent state in the Markov Chain.

In order to facilitate the analysis we will include the next assumption

Assumption 3: the arrival and service time rates $\lambda_i(u_i(t))$ and $\mu_i(c_i(t))$ are linear for all classes. This is: $\lambda_i(u(t)) = \alpha_i - \beta_i u_i(t)$ and $\mu_i(c(t)) = \gamma_i + \eta_i c_i(t)$ where $\alpha, \beta, \gamma, \eta$ are known coefficient of the linear functions.

2.2 Dynamic Programming Formulation

As stated in the previous section, the problem introduced is a finite-state, continuous-time, average reward, dynamic programming problem. The control set $\mathcal{U} = \big\{\{\mathbf{u} \mid 0 \leq u_i \leq u_{i,max} \quad \forall i\}, \{\mathbf{c} \mid 0 \leq c_i \leq c_{i,max} \quad \forall i\}\big\}$ is compact and all states in the Markov Chain communicate, in other words, there exists a policy under which we can reach a specific state $\mathbf{N}'$ starting from any feasible $\mathbf{N}$.

By assuming linear dependence of the arrival and service processes on the controls (prices) we know that demand $\lambda_i(u_i)$ and service functions $\mu_i(c_i)$ are continuous. Moreover, the transition rates and the reward function are also continuous on the decision variables. Hence, the reward rate and the expected time at each state $\mathbf{N}$ are bounded functions of u_i and c_i (since μ_i and c_i are bounded).

Following this argument, we want to find the expected-bound of the transition flow rate v (i.e., the expected value of the fastest transition from one state to another). In the context of the problem this happens when the bandwidth provider offers a connection price of zero (this will decrease the inter-arrival time of customers to its minimum) and the highest cost for service time (this will increase the probability of a customer leaving the bandwidth). Hence, we have

$$v = \sum_{i=1}^{M} (\lambda_{0,i} + \mu_{max,i} \lceil R/r_i \rceil) \tag{3}$$

And the fastest expected transition is equal to

$$\mathbf{E}[v] = \mathbf{E}\Big[\sum_{i=1}^{M} (\lambda_{0,i} + \mu_{max,i} \lceil R/r_i \rceil) \Big] \tag{4}$$

For ease of notation we say that $\mathbf{E}[v]$ is equivalent to $1/(\Delta\hat{t})$. Having defined this, we clarify that the cost c_i is equal to the incurred cost by customer c_i in $\Delta\hat{t}$ units of time $c_i(\Delta\hat{t})$. The reason why we are interested in the expected fastest transition is because this will be an upper-bound on the selection of a time-step to uniformize the continuous Markov Chain.

In summary, the system is as follows:

State space: $\mathbf{N} = [n_1, ..., n_m]$

Control space: $\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} \mathbf{u} \\ \mathbf{c} \end{bmatrix} \leq \begin{bmatrix} \mathbf{u_{max}} \\ \mathbf{c_{max}} \end{bmatrix}$

Transition probabilities: see figure 1.

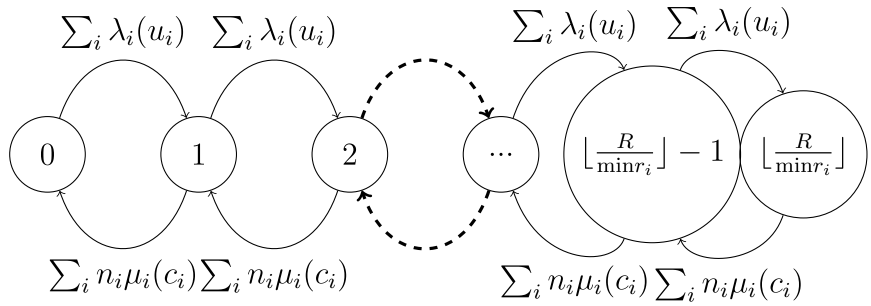

Figure 1: Markov Chain for the multi-class problem

2.3 Uniformization of Markov Chain

Recalling the uniformization of continuous Markov chains, where the Hamilton Jacobi Bellman equation (H-J-B) has the form [2]

$$h(n(t)) + \bar{J}\Delta\hat{t} = \max_{\mathbf{u}(t),\mathbf{c}(t)\in\mathcal{U}} \mathbf{E}_{\text{enabled events}} \Big[\int_t^{t+\Delta\hat{t}} g(n(t), u(t), w(\Delta t)\hat{)} dt + h(f(n(t), u(t), w(\Delta\hat{t}))\Big]$$

We can make uniform the Markov chain describing the bandwidth yield management problem. Then we get the next expression:

$$\begin{aligned} h(\mathbf{N}) + \bar{J}\Delta\hat{t} = \max_{u(n_T),c(n_T)} \Big[& \sum_{i\notin C(\mathbf{N})} \lambda_i(u_i)u_i\Delta\hat{t} + \sum_{i=1}^{M} c_i n_i + \sum_{i\notin C(\mathbf{N})} \lambda_i(u_i)h(\mathbf{N}+\mathbf{e_i})\Delta\hat{t} \\ & + \sum_{i=1}^{M} \mu_i(c_i)n_i h(\mathbf{N}-\mathbf{e_i})\Delta\hat{t} + \Big(1 - \Delta\hat{t}\Big(\sum_{i\notin C(\mathbf{N})} \lambda_i(u_i) - \sum_{i=1}^{M} \mu_i(c_i)\Big)\Big)h(\mathbf{N})\Big] \end{aligned} \tag{5}$$

where $n_T(t) = \sum_i n_i(t)$ and $C(\mathbf{N}) = \{i \mid (\mathbf{N}+\mathbf{e_i})'\mathbf{r} > R\}$.

In this formulation, we assume that the pricing of the bandwidth for both connection and service times depend on the total number of users without considering their specific customer class. This assumption reduces the complexity of the problem. It is natural to argue that from a bandwidth congestion point of view, this modification would not impact the solution of the problem dramatically.

It is worth noticing that the first two additions of equation (5) are summing the income (or costs) for the time period $\hat{\Delta t}$. Conversely, the last three terms refer to the expected transition probabilities of an arrival event, a departure event, and, none arrival or departure event.

Now, we can express equation (5) in terms of the functions describing the non-homogeneous Poisson processes, this is

$$\begin{aligned} h(\mathbf{N}) + \bar{J}\Delta\hat{t} = \max_{u(n_T),c(n_T)} \Big[& \sum_{i \notin C(\mathbf{N})} (\alpha_i - \beta_i u_i) u_i \Delta\hat{t} + \sum_{i=1}^{M} n_i c_i \\ & + \sum_{i \notin C(\mathbf{N})} (\alpha_i - \beta_i u_i) h(\mathbf{N} + \mathbf{e_i}) \Delta\hat{t} + \sum_{i=1}^{M} (\gamma_i + \eta_i c_i) n_i h(\mathbf{N} - \mathbf{e_i}) \Delta\hat{t} \\ & + \Big(1 - \Delta\hat{t}\Big(\sum_{i \notin C(\mathbf{N})} (\alpha_i - \beta_i u_i) - \sum_{i=1}^{M} (\gamma_i + \eta_i c_i) n_i \Big)\Big) h(\mathbf{N}) \Big] \end{aligned} \tag{6}$$

In this expression, we can see that the maximization operator is over quadratic functions of u_i and over linear c_i. Then we can take partial derivatives to obtain insight on the form of the solution.

$$\begin{bmatrix} \frac{\partial h(\mathbf{N}) + \bar{J}\Delta\hat{t}}{\partial u_i} \\ \frac{\partial h(\mathbf{N}) + \bar{J}\Delta\hat{t}}{\partial c_i} \end{bmatrix} = \begin{bmatrix} \alpha_i + 2\beta_i u_i^* + \beta_i h(\mathbf{N} + \mathbf{e_i}) - h(\mathbf{N})\beta_i \\ n_i + n_i \eta_i h(\mathbf{N} - \mathbf{e_i}) \Delta\hat{t} - h(\mathbf{N})\eta_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7}$$

which lead to:

$$\begin{bmatrix} u_i^* \\ c_i^* \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(h(\mathbf{N} + \mathbf{e_i}) - h(\mathbf{N})) - \frac{\alpha_i}{2\beta_i} \\ \text{N/A} \end{bmatrix} \tag{8}$$

From this derivation, we get two interesting results. First, the control u_i^* has a solution that depends on the parameters α_i , β_i and the differential functions $h()$. By plugging u_i^* onto the H-J-B equation, we can observe that the solution will be quadratic on the differential functions. Along the same lines, by taking advantage that in state $\mathbf{h}(\mathbf{0}) = 0$ [1] the only possible events are of the type $\mathbf{h}(\mathbf{0} + \mathbf{e_i})$. The

[1] clearly the cost is zero since there is nobody in the system

resulting equation for that state will have $\bar{J}$ and $\mathbf{h}(\mathbf{e_i})$ as unknowns, as follows

$$\bar{J}\Delta\hat{t} = \sum_{i=1}^{M}(\alpha_i - \beta_i u_i^*)u_i^*\Delta\hat{t} + n_i c_i^* + (\alpha_i - \beta_i u_i^*)h(\mathbf{e_i})\Delta\hat{t} \tag{9}$$

This result might not be very attractive for the multi-class problem, however if one thinks of the single-class problem [2], then the solution to the system of equations is easier by solving for $h(0)$ first, then for $h(1)$ and so on. To compute the solution for the multi-class problem, one can use classical dynamic programming algorithms (i.e. value iteration, policy iteration or linear programming). It is important to mention that the computational complexity increases as the state-space and the control-space increase. Also, the growth of the size of the problem is exponential in the number of classes required.

The second issue in this formulation, as the reader has probably figured out by now, is that we have no expression to find c_i^*. In order to overcome this problem, we need to impose the following assumptions.

Assumption 4: The *non-homogeneous Poisson process* describing the departure process is quadratic on the service costs with $\mu_i(c_i) = \gamma_i + \eta_i c_i^2$.

Assumption 5: There exists a bound $c_{i,\max}$, specified by the bandwidth provider, in which the new arrivals spend zero time on the service.

Now, lets analyze this case for the single-class problem when service time is quadratic on service prices.

3 Single-class, service process quadratic on price

In order to facilitate the analysis, we will now consider the case where there is a single-class of customers. Define the state $n \in \mathcal{S} = \{1, ..., \lfloor R/r \rfloor\}$ as the number of customers using a capacity of size r of the total bandwidth capacity R. Denote u as the price that a customer incurs when connecting to the bandwidth, and c the cost incurred for using r capacity of the bandwidth in $\Delta\hat{t}$ units of time. Finally, denote with $\lambda(u) = \alpha - \beta u$ and $\mu(c) = \gamma + \eta c^2$ as the parameters of the arrival and departure Poisson processes respectively.

[2]taking away: summation, i indices and $h(\mathbf{e_i}$ becomes $h(1)$

To make uniform the H-B-J equation we look for the maximum expected transition flow. This is $\Delta\hat{t} = \lambda(u_{\min}) + (\lfloor R/r \rfloor \mu(c_{\max}) = \alpha + (\lfloor R/r \rfloor(\gamma + \eta c_{\max}^2))$. Hence the H-B-J has the form

$$h(n) + \bar{J}\Delta\hat{t} = \max_{u,c}\{\lambda(u)u\Delta\hat{t} + cn + \lambda(u)h(n+1)\Delta\hat{t} + \mu(c)nh(n-1)\Delta\hat{t} \\ + (1 - \Delta\hat{t}(\lambda(u) - n\mu(c)))h(n)\} \tag{10}$$

which is the same as

$$h(n) + \bar{J}\Delta\hat{t} = \max_{u,c}\{(\alpha - \beta u)u\Delta\hat{t} + cn + (\alpha - \beta u)h(n+1)\Delta\hat{t} \\ + (\gamma + \eta c^2)nh(n-1)\Delta\hat{t} + (1 - \Delta\hat{t}((\alpha - \beta u) - n(\gamma + \eta c^2))))h(n)\} \tag{11}$$

then looking for u^* and c^* we get

$$\begin{bmatrix} u^* \\ c^* \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(h(n+1) - h(n)) - \frac{\alpha}{2\beta} \\ (2\Delta\hat{t}\eta(h(n-1) + h(n)))^{-1} \end{bmatrix} \tag{12}$$

Using the same analogy as in section 2.3, we can plug u^* and c^* into the H-J-B equation and obtain the differential functions $h(\cdot)$ which are quadratic and "inverse quadratic" in $\bar{J}$. Also note that when we apply $h(0) = 0$ we can use a similar intuition as in the previous section to solve the problem. It is important to remember that when $h(0) = 0$, then $c^* = 0$. Hence, in this case, the corresponding equation is:

$$\begin{aligned} \bar{J} &= (\alpha - \beta u^*)u^* + (\alpha - \beta u^*)h(1) \\ &= (\alpha - \beta(\frac{1}{2}h(1) - \frac{\alpha}{2\beta}))(\frac{1}{2}h(1) - \frac{\alpha}{2\beta})) + (\alpha - \beta(\frac{1}{2}h(1) - \frac{\alpha}{2\beta})))h(1) \end{aligned} \tag{13}$$

which is quadratic in $h(1)$ and has the same properties as when departure is not a function of price (a static parameter). This is similar to the case presented in Paschalidis et al. (2000) [4] and in the classnotes of SE710 at Boston University [1].

3.1 Numerical results and insights

In order to have a better understanding of the behavior of the solution, we implement the problem in Matlab [3]. In this example we set: $R = 50$; $r = 1$; $\alpha = 100$; $\beta = 2$; $\gamma = 1$; $\eta = 0.2$. With these parameters we are considering a high sensitivity of

[3] code available in the appendix of this report

attracting new customers by switching the connection price u. We also set the upper-bounds to be $u_{max} = 50$ and $c_{max} = 3$.

The solution shows two main results. First, it shows that the optimal control is achieved when we set $u = 22$ and $c = 2.4$. These prices try to maintain the state of the system (number of customers connected) in a range that maximizes profits. As expected when n is small, the suggested price is smaller in order to attract new customers. Conversely, as the number of customers gets closer to the full capacity of the bandwidth, the control policy suggests increasing prices in order to avoid losing customers as well as to operate as close as possible to full capacity (see figure 2).

The second important insight is that the service cost is more important than the connecting price. This follows from the fact that the optimal solution is closer to c_{max} than to u_{max}. This result is aligned with the pricing strategies of actual network services where it is common to charge for service but not for connecting costs.

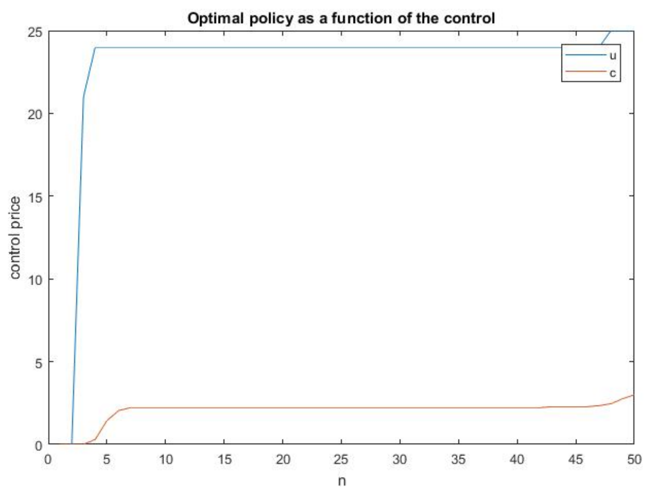

Figure 2: Optimal policy for the single-class, quadratic service time example.

In order to understand the Quality of Service (QoS) that the network company offers to its customers, as well as the usage of the bandwidth, we calculate the stationary distribution of the Markov Chain (see figure 3). For these specific parameters, the solution that maximizes revenues tries to maintain the bandwidth around half of its capacity. This is because the customer arrivals are very sensitive to prices and this will allow to have a good proportion of the customers paying the service fee.

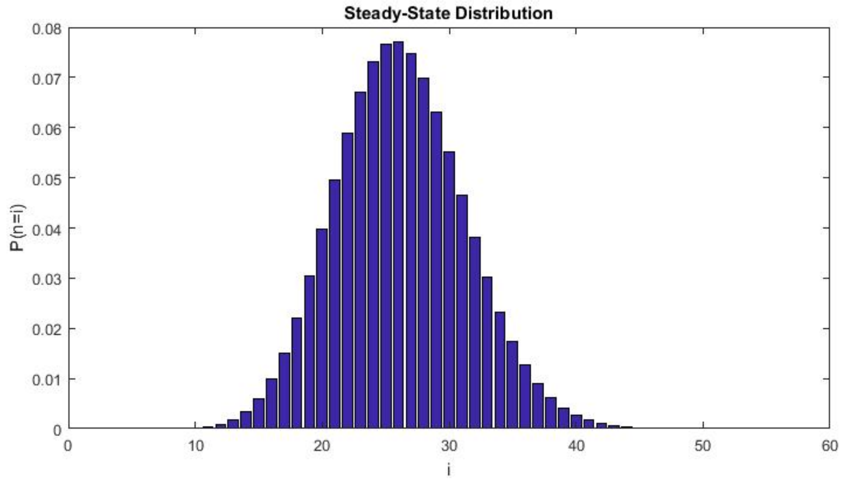

Figure 3: Stationary distribution solution for the single-class, quadratic service time example.

4 Single-class without pricing on service times

In this section we will focus on an easier instance of the problem. The objective is to compute results for the case where the service time (departure process) has a constant parameter μ and where there is no cost associated with the service times. There are two main reasons why we include this problem. First, this scenario happens in real systems when customers do not consider the service fee for usage (e.g. telephone companies), or when service times do not depend on the customer (e.g. processing on servers). Second, we present this example in order to validate our results with the ones presented in Paschalidis et al. (2000) [4].

Assume that there is a single class n_1 with arrival parameter λ_1 and service parameter μ_1. Further more, assume that the user bandwidth requirement is $r = 1$. Then, we have state-space $\mathcal{S} = \{1, ..., m\}$ and control space $\mathcal{U} = [u_{\min}, \alpha/\beta]$ with probability transition matrix as depicted in Figure 4.

Now, lets find the expected value of the fastest transition, this is $\mathbf{E}[v] = \mathbf{E}[\lambda(u_{min}) + (R/mr_1)]$. Hence, rewriting the uniform H-J-B we have

$$h(n) + \bar{J}\hat{\Delta}t = \min_u \big\{\lambda_1(u)u + \Delta\hat{t}\lambda_1(u)h(n+1) + \Delta\hat{t}n\mu_1 h(n-1) + (1 - \Delta\hat{t}(\lambda_1(u) + n\mu_1)h(n)\big\} \tag{14}$$

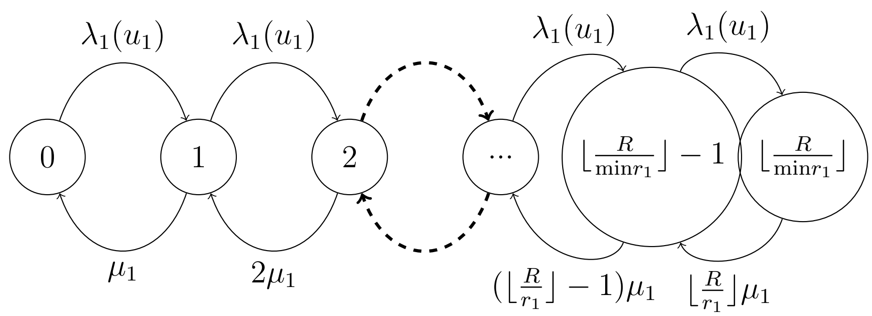

Figure 4: Markov Chain for the single-class problem

which is equal to

$$
\begin{aligned}
h(n) + \bar{J}\Delta\hat{t} = \min_u \big\{ &(\alpha + \beta u)u + (\alpha + \beta u)h(n+1)\Delta\hat{t} + \\
&n\mu_1 \Delta\hat{t} h(n-1) + (1 - \Delta\hat{t}(\alpha + \beta u) + n\mu_1))h(n) \big\}
\end{aligned}
\tag{15}
$$

where

$$
\frac{\partial h(n) + \bar{J}}{\partial u} = \alpha + 2\beta u + \beta h(n+1) - \beta h(n) \tag{16}
$$

$$
u^* = \frac{h(n) - h(n+1)}{2} - \frac{\alpha}{2\beta} \tag{17}
$$

4.1 Numerical results

Repeating the exercise section 3.1, we implement the problem in Matlab [4]. For this problem we use: $R = 100$; $r = 1$; $\alpha = 100$; $\beta = 10$; $\gamma = 1$; $\eta = 0.00001$. and uper-bound to be $u_{max} = 10$.

As expected from previous results, and considering that the only control in this case is the connecting price, it is optimal to increment the price as the system gets congested. In this way, users are discouraged from connecting when the system is overutilized.

We observe from our results (see figure 5) that the long-term optimal cost is archived when $u^* = 31$. This solution has the same shape of the solution in Paschalidis (2000) [4], therefore we have validated our model. Moreover, the stationary

[4]code available in the appendix of this report

distribution of the system in this case favors having a system that operates close to its maximum capacity (see figure 6).

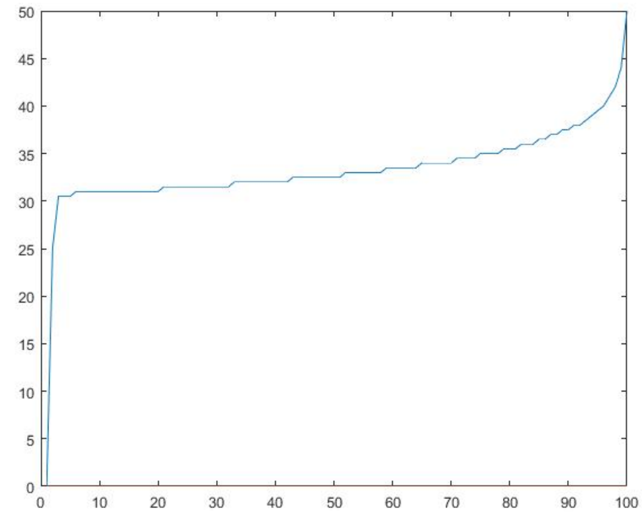

Figure 5: Optimal policy for the single-class, constant service time example. x-axis is the number of customers in the bandwidth, y-axis is the price u

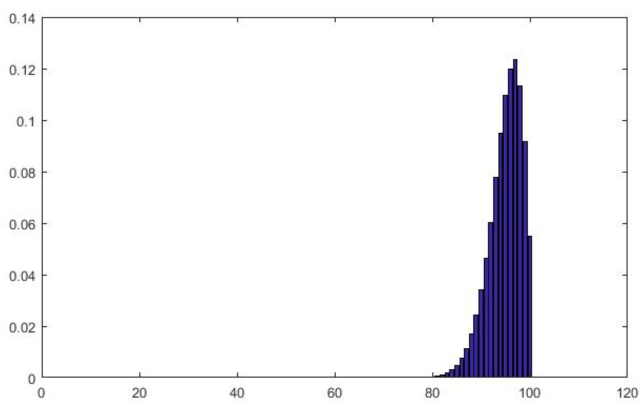

Figure 6: Stationary distribution solution for the single-class, constant service time example. x-axis is the number of customers in the bandwidth, y-axis is the probability of being in that state.

5 Conclusion

In this work we use the infinite horizon machinery of dynamic programming to solve the problem of dynamically pricing the connection costs and service fees for customers connecting to a bandwidth service with finite capacity. We extended the results shown in Paschalidis et al. (2000) [4] by considering dependence of the departure process on the service fee and we provided insight in the results.

As part of the report we included the derivation of the problem as well as interesting properties on how to solve the problem in an easier way.

References

[1] Michael Caramanis. VI. Continuous Time DP problems with Controllable Markov Jump Process Dynamics, 2017.

[2] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. 2005.

[3] Lee W. Mcknight and Joseph P. Bailey. Internet economics: When constituencies collide in cyberspace, 1997.

[4] Ioannis Ch Paschalidis and John N. Tsitsiklis. Congestion-dependent pricing of network services. *IEEE/ACM Transactions on Networking*, 8(2):171–184, 2000.

Appendix

Table 1: Optimal policy for single-class, quadratic service times

n	u^*	c^*
1	0	0
2	0	0
3	21	0
4	24	0.3
5	24	1.44
6	24	2.04
7-42	24	2.22
43	24	2.28
44	24	2.28
45	24	2.28
46	24	2.28
47	24	2.34
48	25	2.46
49	25	2.76
50	25	3

Single-class quadratic service costs

```
clear all
clc;

R = 100;
r1 = 1;

alfa = 500;
beta = 10;
gamma = 1;
eta =  0;

u_max = alfa/beta;
u_min = 0;
c_max = 0.000001;
c_min = 0;

S = [1:floor(R/r1)];
A = [u_min : ((u_max-u_min)/100) : u_max;
     c_min : ((c_max-c_min)/100) : c_max];

v =  alfa+(gamma+eta*c_max)*(R/r1);

P=[];
PR=[];
cntl = [];
cnt = 0;

for a1= 1:size(A,2)
    for a2 = 1:size(A,2)
        cnt = cnt+1;
        cntl = [cntl; cnt A(1,a1),A(2,a2)];
        for s1 = 1:R
            PR(s1,cnt) = (alfa-beta*A(1,a1))*A(1,a1) ...
                +(gamma+eta*A(2,a2))*A(2,a2);
            for s2 = 1:R
                if s1==s2-1
                    P(s1,s2,cnt) = (alfa-beta*A(1,a1))/v;
                elseif  s1==s2+1
                    P(s1,s2,cnt) = s1*(gamma+eta*A(2,a2))/v;
                elseif  s1==s2
                    P(s1,s2,cnt) = 1 - (alfa-beta*A(1,a1))/v - ...
                        (s1*(gamma+eta*A(2,a2))/v);
                else
                    P(s1,s2,cnt) = 0;
                end
            end
        end
    end
```

```
end

PR = PR; %since maximizing

% initial policy
policy=[];
for i = 1:R
   policy(i,1)= 1;
end

discount = 1-0.00001;
epsilon = 0.0001;
max_iter = 10000;

[V, policy, iter, cpu_time] = mdp_policy_iteration(P, PR, discount);
%[policy, iter, cpu_time] = mdp_value_iteration(P, PR, discount);

opt_policy = [];
for i=1:size(policy,1)
    opt_policy = [opt_policy; cntl(cntl(:,1)==policy(i),2:3)];
end

opt_policy
figure;
plot(opt_policy)

[Ppolicy, PRpolicy] = mdp_computePpolicyPRpolicy(P, PR, policy);
steady_dist= (Ppolicy)^1000;
figure;
bar(steady_dist(1,:));
```

Single-class no service costs

```
clear all
clc;

R = 100;
r1 = 1;

alfa = 500;
beta = 10;
gamma = 1;
eta =  0;

u_max = alfa/beta;
u_min = 0;
c_max = 0.000001;
c_min = 0;

S = [1:floor(R/r1)];
A = [u_min : ((u_max-u_min)/100) : u_max;
     c_min : ((c_max-c_min)/100) : c_max];

v =  alfa+(gamma+eta*c_max)*(R/r1);

P=[];
PR=[];
cntl = [];
cnt = 0;

for a1= 1:size(A,2)
    for a2 = 1:size(A,2)
        cnt = cnt+1;
        cntl = [cntl; cnt A(1,a1),A(2,a2)];
        for s1 = 1:R
            PR(s1,cnt) = (alfa-beta*A(1,a1))*A(1,a1) ...
                +(gamma+eta*A(2,a2))*A(2,a2);
            for s2 = 1:R
                if s1==s2-1
                    P(s1,s2,cnt) = (alfa-beta*A(1,a1))/v;
                elseif  s1==s2+1
                    P(s1,s2,cnt) = s1*(gamma+eta*A(2,a2))/v;
                elseif  s1==s2
                    P(s1,s2,cnt) = 1 - (alfa-beta*A(1,a1))/v - ...
                        (s1*(gamma+eta*A(2,a2))/v);
                else
                    P(s1,s2,cnt) = 0;
                end
            end
        end
    end
```

```
end

PR = PR; %since maximizing

% initial policy
policy=[];
for i = 1:R
   policy(i,1)= 1;
end

discount = 1-0.00001;
epsilon = 0.0001;
max_iter = 10000;

[V, policy, iter, cpu_time] = mdp_policy_iteration(P, PR, discount);
%[policy, iter, cpu_time] = mdp_value_iteration(P, PR, discount);

opt_policy = [];
for i=1:size(policy,1)
    opt_policy = [opt_policy; cntl(cntl(:,1)==policy(i),2:3)];
end

opt_policy
figure;
plot(opt_policy)

[Ppolicy, PRpolicy] = mdp_computePpolicyPRpolicy(P, PR, policy);
steady_dist= (Ppolicy)^1000;
figure;
bar(steady_dist(1,:));
```