UNIVERSITY OF CALIFORNIA
Los Angeles

Information Chains and Content Management

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Management

by

Boyoun Choi

2015

ABSTRACT OF THE DISSERTATION

Information Chains and Content Management

by

Boyoun Choi
Doctor of Philosophy in Management
University of California, Los Angeles, 2015
Professor Uday S. Karmarkar, Chair

The dissertation consists of three chapters that study topics in information chains. Information chains acquire, process and distribute content (information or data) in a way analogous to physical goods in supply chains. However information chains are different in that demand does not deplete inventories. Rather, information can lose value over time, become erroneous, or false, so that it needs to be purged. Furthermore, it is often not possible to plan or control the production of content, and the arrival of data elements or content can be stochastic. In effect flow in the chain is driven by content arrival, rather than inventory depletion. In the first chapter, we describe the flow of information in the content processing and storage portion of the chain using different models, and formulate decision problems related to capacity planning. The objectives include the size of the content data base (as a proxy for value) and the time required for content to be available to users.

In the second chapter, we consider a dynamic capacity planning problem for information content management in supply chains, where information is processed

to be entered into a data base, which is then available to customers and users. Information processing is done by workers, and the capacity decision requires the determination of the number of workers required to process a time varying work load. The average time taken for processing is an important performance parameter, and this drives capacity decisions. In the short term planning problem it is possible to vary capacity by using overtime and part-time work. The problem is formulated as a multi-period, nonlinear, mixed integer program. Clearing functions are used to capture processing delays. Small problems can be solved optimally, but large cases can be challenging. We develop Lagrangean relaxation methods to decompose the problem and generate lower bounds, and propose heuristics for solving the problem efficiently.

The third chapter models and studies a decision problem faced by supplier of information content. With customers that are sensitive to both price and release time of content, we study profit maximization problem for a monopolist provider. Then we look at the case of supplier with two downstream distribution channels where the channels have to compete on setting their price and release time for the identical content to attract more customers. When supplier fixes the release time for the content and charges fixed prices to both channels, we find that there is an equilibrium pair of prices for the channels to charge the customers.

The dissertation of Boyoun Choi is approved.

Felipe Caro

Charles Corbett

Ichiro Obara

Uday S. Karmarkar, Committee Chair

University of California, Los Angeles

2015

To my family

Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor and chair of dissertation committee, Professor Uday Karmarkar. It was truly a privilege for me to work under his supervision while he provided tremendous guidance and invaluable advice. His encouraging comments and exceptional mentorship helped me move forward and accomplish my educational goals. I learned how to translate an idea into formulations and lay those out as models. I am also truly thankful for his patience and generosity during my Ph.D. program.

I am also extremely grateful to my dissertation committee members: Professor Charles Corbett, Professor Felipe Caro, and Professor Ichiro Obara for their valuable time and advice on my dissertation. I was privileged to learn in-depth topics in dynamic programming from Felipe Caro which helped broaden my analytical skills. I also had a great opportunity to work as a teaching assistant for Professor Corbett's Operations Management course, where I learned about valuable managerial insights and real world problems in operations. I am thankful to Professor Obara for teaching me fundamental concepts of microeconomics and game theory which were essential in completing many parts of my dissertation.

I also wish to acknowledge other UCLA faculty members from whom I had opportunities to learn and build on many important analytical skills and acquire invaluable knowledge in various disciplines of Operations Management: Professor Reza Ahmadi, Professor Christiane Barz, Professor Kumar Rajaram, and Professor Rakesh Sarin. I am sincerely grateful for advice and guidance from Professor Barz during my teaching assistantship for Data and Decisions courses.

In addition, I would like to thank UCLA Anderson Graduate School of Management for generous funding over the past five years. I would also like to acknowledge Easton Technology Leadership Program at Anderson School for their financial support that provided me resource to conduct research and to present my work at conferences. I also want to thank Lydia Heyman for all her support during my Ph.D.

program.

I would like to thank DOTM Ph.D. students that graduated before me who provided precious guidance during my studies: Dimitrios Andritsos, Foad Iravani, Morvarid Rahmani, George Georgiadis, Aparupa Das Gupta, and Jaehyung An. Finally, I would like to thank all my colleagues, Wei, Paul, Sandeep, and Christian with whom I had great time at Anderson together.

Last but not least, I would like to thank my family for endless love and support that they gave me throughout my life and especially during my Ph.D. program. In particular, I am extremely grateful to my husband, Jong Woung Kim for his encouragement, understanding, prayers, and patience that helped me move forward throughout my studies.

VITA

2007	Sc.B., Applied Mathematics-Computer Science Brown University
2007-2010	Software Engineer Oracle Corporation
2010-2015	Graduate Student Researcher UCLA Anderson School of Management

Chapter 1

Content Management and Capacity Decisions in Information Chains

1.1 Introduction

The rapid growth of information chains is a major part of the transition of the US from a material-based to an information-based economy (Apte et al. 2012). Just as physical goods are moved from stage to stage in supply chains, the flows in information chains are of data or information. One form of this data is experiential digital content such as songs, video streams, and images. Other examples include financial data, catalogs of parts and products, weather data, content portals, news, magazines and books. Transaction based services such as retail banking or online retailing, also generate content and the transactions themselves could be considered a micro form of content.

As an example of content management, consider the news service industry. A news distribution company like a TV broadcaster or newspaper, receives news items from their own and from external sources whenever a newsworthy event occurs. They have to decide which stories to acquire and process. The selected items are then edited, formatted and processed to the appropriate distribution format, and made available to users through their distribution mechanism, whether broadcast, print or online. This sequence of stages from the arrival of data to processing to distribution accessible to users is an example of an information chain. Figure 1.1 shows a schematic example of these stages.

Taking the viewpoint of an industry sector, section 1.5 in Appendix shows the information chain for weather services (Connor 1998). The stages in this chain include data acquisition, processing, storage, and finally distribution in various forms at a retail level. The content itself is converted from raw data acquired from a variety of sensors, to formatted data, to weather forecasts including graphics, and finally to information at accessible to end consumers. The chain includes both B2B and B2C transactions and has multiple market segments, ranging from commuters to agriculture to aviation.

A distinct characteristic of such information chains that differentiates them from supply chains is that when the final product of information is consumed, there is no depletion of inventory. It is not demand that eventually reduces inventory, but obsolescence and active purging. Content becomes obsolete either when it becomes false, or when the demand for it, and its value, declines. Examples of false content could be a listing in a guidebook for a restaurant that has since closed down, the temperature at a particular location, or a price listing for an item for which the price has changed. Content types with declining value and declining demand over time, include music, videos, news and books (fiction or non-fiction).

Another important characteristic of information chains is that often one cannot control or predict when a new item will arrive. There are some situations like weather services case where raw weather data is captured fairly regularly and predictably at monitoring points. A more typical case is the legal database industry, where a publisher has to process all the legal cases that arrive on a random basis to have them available in a database within a required time (Karmarkar 2014). As another example, an electronic parts catalog publisher (Bashyam and Karmarkar 2000) must find, select and process electronic component information from parts manufacturers as and when it becomes available, and then update the database to be accessible to users. In publishing articles or research papers, the content arrivals are random, and must be processed (reviewed, possibly rejected, edited) to be available for publication. In the insurance industry, applications for insurance policies can arrive at any time, and must be processed rapidly and correctly (Harrison 1997). While this is the back-room of a service process, the process characteristics, performance requirements, costs and management decision issues are very similar. For companies in such information chains, operational decision-making is required at many stages of the chain, from the acquisition and selection of input items, and processing of content and format, to detecting and then purging the database of obsolete items. This gives rise to processing capacity decisions which will then determine the quality of the data base

in terms of the time of availability, quality of content, and correctness.

In this chapter, we focus on the processing stage of the chain. Before introducing decision models, we first describe different approaches to modeling the information flow in the chain, from arrival to obsolescence. Mainly using Poisson arrival assumptions, we consider the number of processors available to process the arriving items, and the characteristics of obsolescence. The simplest case assumes that data items may become obsolete only after they are in the database. In a second case, data are subject to obsolescence as soon as they are acquired, meaning that they may leave the system without being processed while waiting for processing or even while being processed. The flow models are used to formulate decision problems regarding the choice of optimal processing capacity to ensure timely flow into the database. When we are in settings in which items may become obsolete at any time, we consider the loss probability, i.e. the probability of losing the item due to obsolescence before it enters the database. In determining the optimal processing capacity, the cost of additional processors versus the cost of time is a key tradeoff.

The chapter is organized as follows. In section 1.2 we provide a review of existing literature. In section 1.3, we describe five different flow models and derive key operating characteristics of each model. Then in section 1.4, we formulate decision models in three different settings, and analytically or numerically derive the optimal solution. Section 1.5 provides conclusions and possible extensions to the problem, and suggests directions for future research.

1.2 Literature Review

Information chains are a topic that has not been extensively covered in the research literature. Karmarkar and Apte (2007) review the evolution from material-based economy to an information-based economy, and discuss the similarities and differences between operations in information industries and traditional manufactur-

ing industries. Bashyam and Karmarkar (2007) and Bashyam (2000) examine methods of delivering business information, and models the information services market where each provider has either physical or online delivery technology. Papadimitriou (2004) discusses the efficient management and use of information; he treats data like a perishable good and develop optimal update policies to ensure the "freshness" of the database. The frequency of updates is determined considering the cost of update versus the cost of having erroneous data in the database. De Vleeschauwer and Laevens (2009) study a caching algorithm that tracks popularity of objects to make intelligent decisions in TV on-demand services. In computer science literature, Ipeirotis et al. (2005) models database changes over time to predict when each content summary should be updated.

We will see that queuing models are relevant to modeling content management, not unlike models of services. The extensive queuing literature tends to emphasize process models rather than decision models. Models closely related to our case in which obsolescence occurs throughout the flow, are those which address queuing with impatient customers, or queuing with reneging. Haight (1959) considers a queue with a single server with a limit on the holding capacity in which a person may decide to leave and give up service if time exceeds some maximum threshold. He first discusses the point of view of those who join the queue, and later summarizes the behavior of the queue. Barrer (1957) also considers customers who will wait in the queue for at most a fixed time, and derives expressions for the ratio of the average loss rate to the average arrival rate of customers. Ancker and Gafarian (1963a) obtain the probabilities of balking, waiting, reneging, and acquiring service in a single-server facility where customers may balk or renege. As an extension, Ancker and Gafarian (1963b) consider a facility with multiple heterogeneous servers with balking and reneging, and obtain the probability that an arrival reaches service and the mean rate of loss due to balking and reneging.

There is some existing work which includes the optimization of queuing system

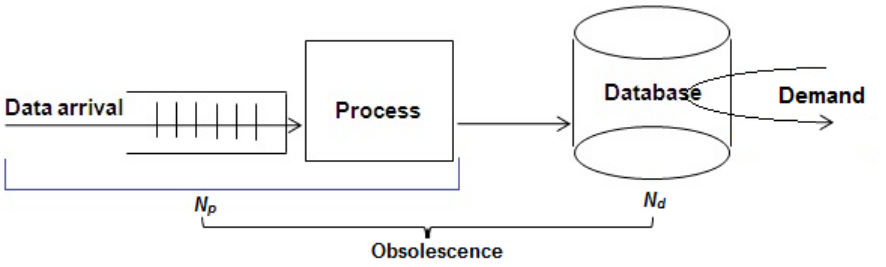


Figure 1.1: General illustration of information flow

parameters. Crabill (1972) considers the determination of optimal control rules for a service facility which has a finite number of possible service rates and a constant input rate, where the objective is to minimize the long-run average expected cost rate of the queue by finding a policy which determines the service rates to be employed at any point in time. George and Harrison (2001) consider a similar problem and obtain the optimal service rates as a function of the number of jobs in system. Both works show that the optimal service rates are non-decreasing as a function of queue length. Since the optimal rate depends on queue length, it is difficult to implement the optimal policy in a real application unless the system size changes infrequently.

Work on applied settings includes the literature on call center staffing problems. Borst et al. (2004) look at the staffing problem of large call centers and determine the asymptotically optimal staffing level as arrival rate increases. In Garnett et al. (2002), optimal staffing rules are analyzed with the existence of impatient customers who might decide to leave before the service begins. They compare the three staffing rules under different magnitudes of load conditions, and perform an asymptotic analysis as load increases indefinitely.

1.3 Flow Models and System Behavior

The basic information flow pattern for content management is shown in 1.1. Data items arrive and are accepted or rejected. The accepted items are processed by the available servers, and are then loaded into the database where they become available to users. The data may become obsolete at any time during the process. Certain different variations of the model are listed in table 1.1 below. Each of the arrival, processing, and obsolescence processes can be either deterministic or stochastic with a given rate. In the first and the third model, we assume that the items may become obsolete only after they enter the database, while in other models, items may become obsolete at any point in time throughout the flow at any time after arrival, as in Figure 1.1.

# of Servers	Arrival	Process	Obsolescence	Model
One	Deterministic	Deterministic	Determinstic/Always	I
	Poisson	Poisson	Poisson/ Only in DB	II
	Poisson	Poisson	Poisson/ Always	III
Many	Poisson	Poisson	Poisson/ Only in DB	IV
	Poisson	Poisson	Poisson/ Always	V

Table 1.1: Model characteristics

We adopt the following notation. Let N_p denote the total number of items waiting to be processed plus those being processed. N_d denotes the number of items in the database. λ and μ denote the rate of arrival and rate of processing, respectively, and $\sigma$is the rate of obsolescence for each item. We introduce alternative modeling approaches, starting with a simple form and then adding complexity in terms of the modeling assumptions.

1.3.1 A Deterministic Flow Model (I)

We assume that items arrive at a known deterministic rate and it takes a server fixed amount of time, T_p, to process an item. By Little's Law, $T_p = N_p/\lambda$, where N_p is the number of items waiting in queue and in process. We define L_1 as the rate of obsolescence from the first part of the flow before items enter the database, and L_2 as the rate of obsolescence from the database. Then $L_1 = N_p\sigma$ and $L_2 = N_d\sigma$, and $\lambda = L_1 + L_2$ for system at steady state. From the given information, we can obtain the database size at steady state, $N_d = \frac{L_2}{\sigma} = \frac{\lambda - L_1}{\sigma} = \frac{\lambda}{\sigma} - N_p = \lambda(\frac{1}{\sigma} - T_p)$.

These expressions give us basic relationships between flows and quantities but are not adequate to capture capacity planning tradeoffs. We will take a look at more realistic scenarios in the following models, where we assume stochastic arrivals.

1.3.2 Single-Server Process with Obsolescence in the Database Only (Model II)

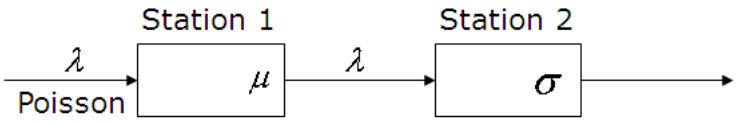


Figure 1.2: Queues in tandem model

We assume that data items arrive according to Poisson process with rate λ. Each item is then processed at a single-server facility according to a Poisson process with rate μ. After the item is processed, it is loaded into the database and is available to users until it becomes obsolete. We assume that items do not become obsolete in the processing stage. This can be a reasonable assumption if the time spent in processing is very small and the rate of obsolescence is also very low. The rate of obsolescence

for each item in the database is σ, and we assume items leave the database (become obsolete) according to a Poisson process. Assuming $\lambda/\mu < 1$, the rate of arrival of items in the database in steady state is also λ, which is the total throughput or output rate from the processing stage (Station 1 in Figure 1.2). Hence, we can view the entire process as two queues in tandem. The first system is a $M/M/1$ queue with arrival and service rates λ and μ, and the second is a $M/M/\infty$ queue with arrival and service rates λ and σ. The number in the first system is N_p in Figure 1.1, and the number in the second system is the number in the database, N_d. The steady-state distribution of the first system is then $P(i) = (1 - \frac{\lambda}{\mu})(\frac{\lambda}{\mu})^i$. The distribution of items in the second system $M/M/\infty$ is $P(j) = \frac{(\lambda/\sigma)^j}{j!}e^{-\lambda/\sigma}$. By Little's Law, we know that the average time it takes for a data item to become available in the database after arrival (the mean delay time of the first queue) is $\frac{1/\mu}{1-\lambda/\mu} = \frac{1}{\mu-\lambda}$. The average number of items in the database at steady state is λ/σ, and the average life time for a data item in the database is $1/\sigma$. Since at steady state, the number of items in queue 1 is independent from the number of items in queue 2, we can obtain the joint stationary distribution as:

$$\begin{aligned} P(i,j) &= P(\text{size of Station 1} = i \ \& \ \text{size of Station 2} = j) \qquad (1.1) \\ &= P(i) \cdot P(j) \\ &= (1 - \frac{\lambda}{\mu})(\frac{\lambda}{\mu})^i \cdot \frac{(\lambda/\sigma)^j}{j!} e^{-\lambda/\sigma} \end{aligned}$$

1.3.3 Multi-Server Process with Obsolescence in the Database Only (Model IV)

We model the multi-server processing facility modeled as a $M/M/m$ queue. Other parameters remaining the same, station 1 in Figure 1.2 now has m servers processing incoming data items. Again, assuming $\lambda < m\mu$ (i.e., $\rho = \frac{\lambda}{m\mu} < 1$), we have the

steady-state distribution of the processing station as

$$\begin{aligned} P(0) &= \left(\sum_{k=0}^{m-1} \frac{(\lambda/\mu)^k}{k!} + \frac{(\lambda/\mu)^m}{m!(1-\rho)} \right)^{-1}, \\ P(n) &= \begin{cases} \frac{(\lambda/\mu)^n}{n!} P(0), & \text{for } 0 < n \leq m-1 \\ \frac{(\lambda/\mu)^n}{m! m^{n-m}} P(0), & \text{for } m \leq n, \end{cases} \end{aligned} \tag{1.2}$$

and the average waiting time in queue is given by

$$\begin{aligned} T_q &= \frac{\rho}{\lambda(1-\rho)} P(m^+) \\ &= \frac{\rho}{\lambda(1-\rho)} \cdot \frac{(\lambda/\mu)^m}{m!(1-\rho)} P(0), \end{aligned} \tag{1.3}$$

where $P(m^+)$ is the probability that all servers are occupied. With λ and μ fixed, the average waiting time is monotonically decreasing in m. The average time for an item to enter the database is $T_q + \frac{1}{\mu}$.

1.3.4 Single-Server Process with Obsolescence in the Entire System (Model III)

In the two preceding sections, data items could only become obsolete after they entered the database. However, for time-sensitive materials such as news, or financial data, this assumption is not realistic. We now assume that the items are subject to obsolescence at any time from the moment that they arrive in the queue to be processed. Obsolescence of data is again assumed to follow a Poisson process with rate σ for each item, so that the total rate is proportional to the number of items present at any stage in the system.

We model the relationship between the number of waiting and in-process items and the number of items in the database as a two-dimensional Markov chain, as in Figure 1.3. Each state is described by the pair (N_p, N_d), and the state space is semi-infinite on each dimension. The Markov chain model corresponds to a level-dependent

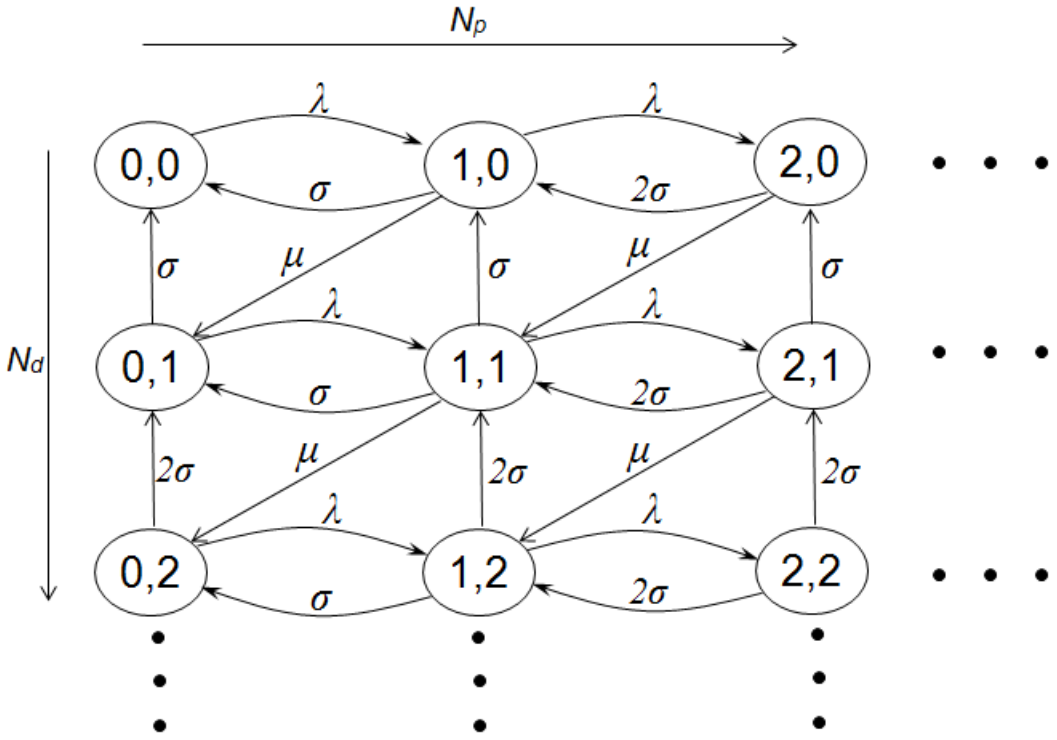


Figure 1.3: 2-dimensional model for N_p and N_d

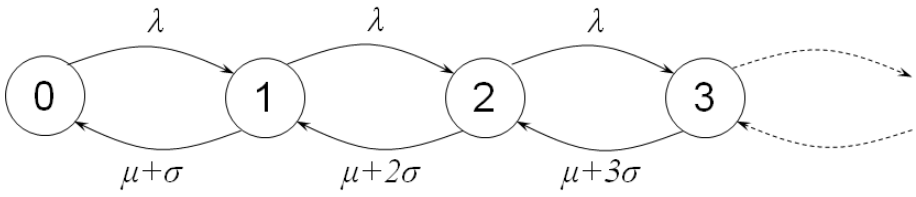


Figure 1.4: Birth-and-death model for the number of waiting and in-process items

quasi-birth-and-death process (Kharoufeh 2011), which is a bivariate Markov process with state space $S = \{(i, j) : i \geq 0,\ j = 1, 2, ..., m\}$ where i is called the level of the process, j is called the phase of the process, and m is an integer that can be finite or infinite. Algorithms for truncating the infinite space and calculating the steady state distribution exist, and numerical examples are included in Appendix 1.5. Given the steady state probabilities of the truncated Markov chain, we can obtain the average size of the database or the average number of waiting and in-process items by straightforward calculations.

A Reduced Model for the Number in Waiting and Process (N_p)

Focusing on the first part of the flow from Figure 1.1, or looking only at horizontal dimension in Figure 1.3, we can represent the number of items waiting or being processed, N_p, as a one-dimensional birth-and-death process (Figure 1.4), with rate of arrival being λ and rate of departure being $\mu + i\sigma$, since data items can either leave the system by obsolescence or by completion of processing. We use i in place for N_p for notational convenience here. This model is similar to the $M/M/1$ queue with reneging (Haight 1959 and Ancker and Gafarian 1963a), except that in our scenario, even the item in the processor may become obsolete, at the same rate σ. Note that we do not need the assumption $\lambda < \mu$ here, since the obsolescence process for any non-zero obsolescence rate ensures that the number in the system does not

grow infinitely. We can derive the steady-state distribution for i as:

$$\begin{aligned} P(0) &= \frac{1}{1+\sum_{n=1}^{\infty} a_n}, \\ P(n) &= P(0)a_n, \\ a_n &= \prod_{i=1}^{n} \frac{\lambda}{\mu + i\sigma}, \\ &= \frac{\lambda^n \Gamma(\frac{\mu}{\sigma}+1)}{\sigma^n \Gamma(\frac{\mu}{\sigma}+n+1)}, \end{aligned} \tag{1.4}$$

where $\Gamma(\cdot)$ is the gamma function.

Deriving the mean time spent in this system is not straightforward, because of the two different components that control the departure. Some of the items depart the system by being processed (at a rate μ), and some items leave at the rate of obsolescence, $i\sigma$, where i includes the one currently being processed. We can determine the proportion of data items that reach the processor before becoming obsolete, denoted P_μ as:

$$\begin{aligned} P_\mu &= \sum_{n=0}^{\infty} P(n) \cdot P(\text{reach the processor} | n \text{ items in system}) \\ &= \sum_{n=0}^{\infty} P(n) \frac{\mu}{\mu + n\sigma} \\ &= \mu P(0) \sum_{n=0}^{\infty} \frac{\lambda^n \Gamma(\frac{\mu}{\sigma}+1)}{\sigma^n (\mu + n\sigma) \Gamma(\frac{\mu}{\sigma}+n+1)}. \end{aligned} \tag{1.5}$$

In fact, equation (1.5) is similar to $P(A)$ in equation (29) from Ancker and Gafarian (1963a), the probability that a new arrival will reach service. Given that an item started being processed, it will either enter the database after completion or become obsolete during the process. The probability that the item will enter the database upon reaching the processor, or the probability of completing the process is given as

follows:

$$\begin{aligned} P_c &= P(\text{server busy})P(\text{processing completed}|\text{item is being processed}) \\ &= (1 - P(0))\frac{\mu}{\mu + \sigma}. \end{aligned} \tag{1.6}$$

Proposition 1. Items enter the database according to Poisson Process with rate $\lambda P_\mu P_c$.

This is straightforward. The initial arrival to the processor is a Poisson process with rate λ, and the proportion of arriving items that eventually enter the database is $P_\mu P_c$, so the rate at which data items enter the database is $\lambda P_\mu P_c$. The loss probability is $1 - P_\mu P_c$. The items that complete processing experience the same environment as $M/M/1$ queue with rate of processing μ, while the items that leave from obsolescence essentially go through $M/M/\infty$ queue where there is no waiting. Using these two facts, we can calculate the average time until the item either becomes obsolete or reaches the processor (i.e. mean time in queue),

$$E(\text{time in queue}) = P_\mu P_c \frac{\rho}{\mu - \lambda} + (1 - P_\mu P_c)\frac{1}{\sigma}, \tag{1.7}$$

using the known expressions for the mean waiting and delay times of two types of queues.

From P1, we also obtain the average number of items in the database, since the flow into the database is equivalent to the flow into station 2 in Figure 1.2, with arrival rate of $\lambda P_\mu P_c$ instead of λ. Thus the average number of items in the database is $\lambda P_\mu P_c/\sigma$.

1.3.5 Multi-Server Processing with Obsolescence in the Entire System (Model V)

The assumptions about obsolescence remains the same as in section 1.3.4, and now we assume a multi-server facility. We omit the analysis for the 2-dimensional model

equivalent to Figure 1.3 and just focus on the birth-and-death model representing N_p, the number of items waiting in queue and in processors. The only difference from the previous multi-server case is that obsolescence of data may occur at any point during the flow. The steady-state distribution can be derived in the same manner as we obtained equation (1.4), since the model is similar to Figure 1.4 with rate of service being $m\mu + i\sigma$. As before, denoting $P(N_p = n)$ as $P(n)$,

$$\begin{aligned} P(0) &= \left(\sum_{k=0}^{m} \frac{(\frac{\lambda}{\mu+\sigma})^k}{k!} + \sum_{k=m+1}^{\infty} \frac{(\frac{\lambda}{\mu+\sigma})^k}{k! \prod_{i=m+1}^{k} (m\mu + i\sigma)} \right)^{-1}, \\ P(n) &= \begin{cases} \frac{(\frac{\lambda}{\mu+\sigma})^n}{n!} P(0), & \text{for } 0 < n \leq m \\ \frac{(\frac{\lambda}{\mu+\sigma})^n}{n! \prod_{i=m+1}^{n} (m\mu+i\sigma)} P(0), & \text{for } n \geq m+1. \end{cases} \end{aligned} \tag{1.8}$$

Expressions equivalent to equations (1.5) and (1.6) can be obtained as follows:

$$\begin{aligned} P_\mu &= \sum_{n=0}^{m} P(n) \frac{\mu}{n\mu + n\sigma} + \sum_{n=m+1}^{\infty} P(n) \frac{\mu}{m\mu + n\sigma} \\ &= P(0) \left(\sum_{n=0}^{m} \frac{\mu (\frac{\lambda}{\mu+\sigma})^n}{(n\mu + n\sigma) n!} + \sum_{n=m+1}^{\infty} \frac{\mu (\frac{\lambda}{\mu+\sigma})^n}{(m\mu + n\sigma) n! \prod_{i=m+1}^{n} (m\mu + i\sigma)} \right), \end{aligned} \tag{1.9}$$

$$P_c = (1 - P(0)) \frac{\mu}{\mu + \sigma}. \tag{1.10}$$

1.4 Decision Models

We have looked at key characteristics of the chain under each flow model in Table 1.1, allowing us to compute quantities like the average time to enter the database, the average size of the database, and the probability that an item becomes obsolete before entering the database (loss probability). We now examine the formulation of decision models.

As with supply chains, capacity determination is an important problem in information chains as well. However, unlike supply chains, inventories cannot be used to

improve the matching of supply to demand. Rather, as with service settings, capacity is driven by the requirements of processing arriving data. As a result capacity is driven more by the spread of the arrival distribution rather than the average.

At an operational level, supply chains require production decisions which depend on the state of inventory and the need for replenishing depleted stocks. However, in information chains, processing is simply initiated by the arrival of data. Now in many cases of content processing, it is possible to choose what data elements to process, and to decline or discard some part of the data. For example, a daily news broadcast must fill a certain amount of time. On a busy news day, many potentially usable news items may be discarded if they are thought to be less significant. On the other hand such items may well be "accepted" and processed on a slow day. Such "accept or decline" decisions are often made at a pre-processing stage in many other content management settings like journal publishing. Note that in service systems, customers often make these decisions (by balking or reneging) when queue times are long.

In some settings, like legal publishing (Karmarkar 2014), since it is necessary to process and include all arriving information, there is no accept-decline option.

The objective function for both planning and operational decisions can be quite specific to particular settings for content management. For example, the costs of holding information are typically so small as to be negligible. Data base size in some cases is a positive feature, since in competition, customers may choose the larger data base over a smaller. For example, in the Aspect case (Bashyam and Karmarkar 2000), "completeness" of the parts catalog is one of their four C's of performance measurement. However, data base size can have diminishing returns with the inclusion of items that are less valuable to the customer. This is often the case with catalogs. In some cases, data base size may also increase search and access costs, as well as the costs of reviewing, updating and purging obsolete items from the catalog.

In a broadcast, or publishing situation, the size of the database or content package might be limited or even fixed, and it may not be of value to simply maximize the size beyond a threshold level. In legal publishing (Karmarkar 2014), the size of the data base is not a decision or design choice, since it is necessary to include all arriving information without exception.

In terms of overall performance, a universally relevant parameter is "currency" or the time from the arrival and potential inclusion of content, to the time when it is available to customers. This is the case for settings like news publishing. The time from arrival to availability can also affect the measure of "loss probability" which the probability that an item never makes it into the data base, because it becomes obsolete while it is still waiting or being processed. The total time to process and loss probability are equivalent measures in the expected sense, but not necessarily in terms of distribution.

In some cases, it may be a reasonable approximation to consider processing rate to be a decision variable. If processing is being performed by scalable capacity, and if the processing task can be correspondingly scaled or divided (as with certain kinds of data processing and file transformation tasks), this may be a reasonable model. However, in certain cases it may be necessary to make capacity decisions in terms of the number of processors. This is often the case where the information processing is being done by human effort or by specialized capacity that cannot easily be varied or scaled.

1.4.1 Optimal Processing Rate

We consider a single-server system with obsolescence (model III). In section 1.3.4, we have developed expressions for the probability of reaching the processor, the probability of entering the database, and the mean time until obsolescence or reaching the database.

In Crabill (1972) and George and Harrison (2001), the objective is to minimize average cost per time unit over infinite horizon, where cost includes holding cost for each item in system and cost of operating the server at a given rate. In examples such as customers waiting in line at a restaurant or a hospital, having a line is often costly; however, in our setting where digital content and electronic files are in place of customers, holding cost is almost zero, or negligible. Instead of minimizing the cost of operation, we want to find a processing rate that minimizes the *loss probability*, or maximizes the proportion of arriving items that enter the database before becoming obsolete, or $P_\mu P_c$. This is especially important for items that lose value quickly.

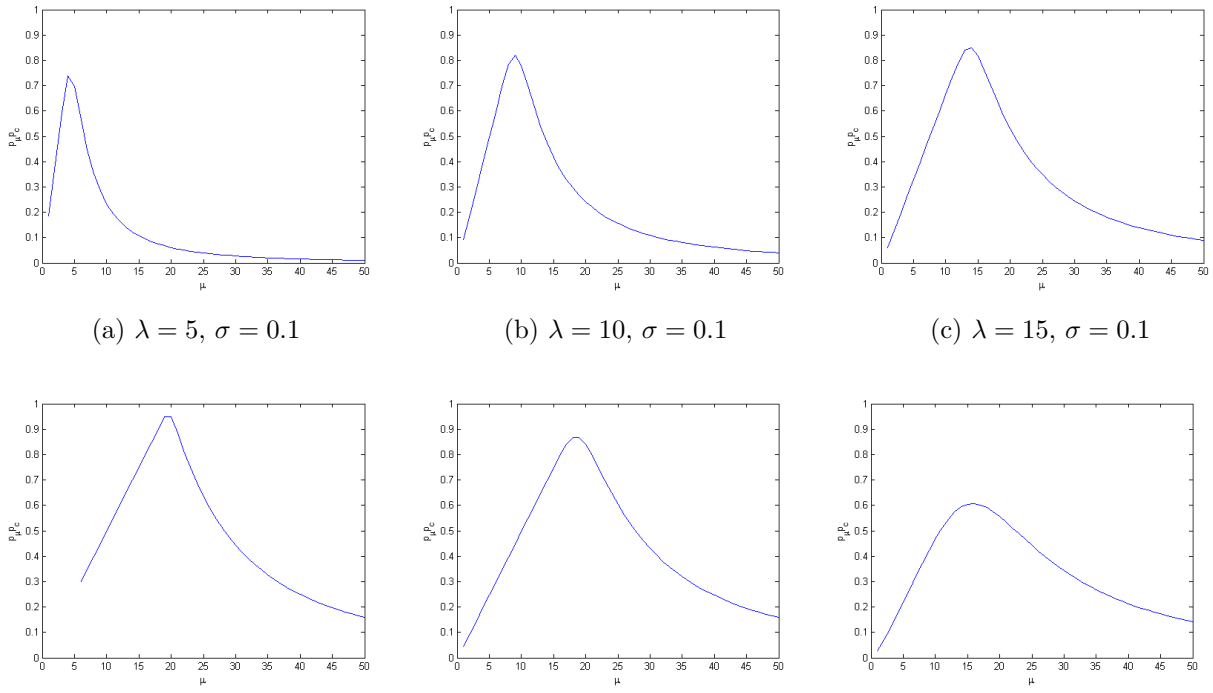
Given an exponential arrival rate λ, the unconstrained problem then becomes $\max_\mu\{P_\mu P_c\}$, and the maximizing processing rate then results in the minimum loss probaility and maximum average size of the database with σ also fixed (recall that loss probability was $1 - P_\mu P_c$ and the average size of the database was $\lambda P_\mu P_c/\sigma$). Using P_μ and P_c derived in equations (1.5) and (1.6), we want to check the relationship between μ and $P_\mu P_c$.

Conjecture 2. *P2. $P_\mu P_c$ is quasiconcave in μ.*

Numerical tests show the unimodal shape of $P_\mu P_c$ as a function of μ. In sum, the processing rate that minimizes the loss probability is close to the value of arrival rate λ given a small fixed value of σ, and as expected, minimum loss probability decreases as arrival rate goes up (Figure 1.5 (a) to (c)). When λ is fixed, the optimal processing rate *decreases* as rate of obsolescence increases, which is unexpected.

1.4.2 Optimal Processing Capacity

In the cases of Aspect Development and Autocite, or many other information providers, timely processing of newly arriving data items is a critical component in the success of business, since customers demand up-to-date items in the database.



(a) $\lambda = 5$, $\sigma = 0.1$ (b) $\lambda = 10$, $\sigma = 0.1$ (c) $\lambda = 15$, $\sigma = 0.1$

(d) $\lambda = 20$, $\sigma = 0.01$ (e) $\lambda = 20$, $\sigma = 0.1$ (f) $\lambda = 20$, $\sigma = 1$

Figure 1.5: Graph of $P_\mu P_c$ as a function of μ (top: with fixed σ; bottom: with fixed λ)

The bigger the processing capacity, the quicker the item enters database (as in section 1.3.3), but bringing additional processors (i.e. hiring more workers or setting up machines) incurs addtional cost. We assume the settings of model IV, and shift our focus to the *number* of processors instead of the processing rate, assuming homogeneous processors (μ is same for all servers). Firms try to minimize the total cost, which includes the processing cost and cost of time.

The minimum cost problem solved by a provider is

$$\min_{m>\lambda/\mu} \{C_p(m) + E[C_t(T_m)]\}, \tag{1.11}$$

where m is the number of processors, $C_p(m)$ is cost of processing per unit time using m servers, which is an increasing function of m, and $C_t(T_m)$ is an increasing cost function of time with T_m denoting the average time it takes an item to enter the database when there are m processors (see equation 1.3). We assume exponential arrival rate λ and exponential service rate μ for each processor added. Also, we impose a lower bound on m so that the processing rate will exceed the arrival rate. Assuming for simplicity that each item has a constant demand once it is in the database, unit cost of time is taken to be equivalent to the price of demand for an item per time period, since a delay in getting an item into the database results in lost revenue. Since m is a positive integer, this is a discrete optimization problem. We show that the expected cost is convex in the number of processors under certain assumptions.

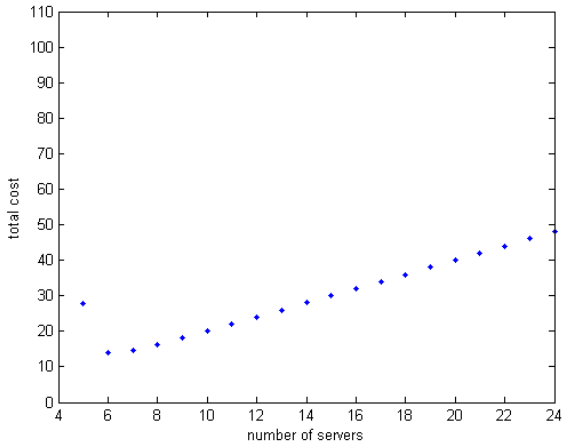
Proposition 3. If $C_p(\cdot)$ is a convex function of m, and $C_t(\cdot)$ is a convex function of time, the expected cost is convex in m, the number of processors.

Proof. T_m is equivalent to $T_q + \frac{1}{\mu}$, where T_q is the mean waiting time in equation (1.3). Dyer and Proll (1977) proved the convexity of the expected queueing time formula for an $M/M/m$ queue, so T_q and thus T_m is convex in m. Since $C_t(\cdot)$ is strictly increasing and convex in time in the assumption, $C_t(T_m)$, a composition function,

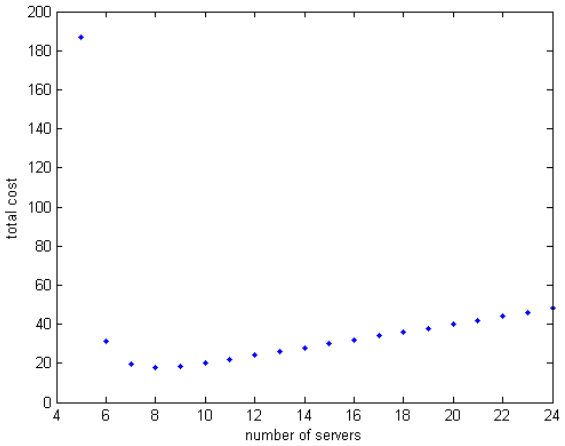
is convex in m (chapter 3.2.4 in Boyd and Vandenberghe 2004). Since expectation preserves convexity, and sum of two convex functions is convex, the expected cost in equation (1.11) is convex in m. □

As a result, the cost-minimizing number of processors can be found by simple marginal analysis. Below is an examples with both C_p and C_t being linear functions.

Example. $\lambda = 1$, $\mu = 0.21$, $C_p(m) = k_1 m$, $C_t(T) = k_2 T$ for some constants k_1 and k_2. k_1 is a cost per processor per time unit, and k_2 is a delay cost per item per time unit.



(a) $k_1 = 2$, $k_2 = 1$: minimum cost at $m = 6$.



(b) $k_1 = 2$, $k_2 = 10$: minimum cost at $m = 8$.

As expected, it is optimal to employ more servers when cost of time increases. Furthermore, the results in this example confirms the square-root safety staffing rule widely used in determining call center capacity (Borst et al. 2004), which gives $N^* = \frac{\lambda}{\mu} + y^*(\frac{k_2}{k_1})\sqrt{\frac{\lambda}{\mu}}$ for the optimal number of staff where $y^*(\cdot)$ is an increasing function for which an approximation was given.

1.5 Conclusion and Future Research

In this chapter we discussed the structure of the information chain highlighting the difference from the traditional supply chains, and formulated a few different models that fit certain scenarios. We categorized the flow models first by number of processors available, and then by the obsolescence characteristic. We considered two scenarios; one is where only the items in the database become obsolete, and the other is where items may become obsolete and leave the system anytime after they arrive. We assumed exponential arrival rate in all models except the deterministic case, and in each model derived the steady state distribution of the number of items in system, average size of the database, and average time to enter the database and loss probability, where applicable.

We then formulated optimization problems based on some of the flow models we discussed. Assuming a single-server facility with obsolescence occurring anytime during the flow, we search for the optimal processing rate which minimizes the loss probability, or maximizes the probability of entering the database and the average size of the database. We provided a conjecture that $P_\mu P_c$, the probability of an item entering the database before becoming obsolete, is quasiconcave in μ, illustrated with numerical examples.

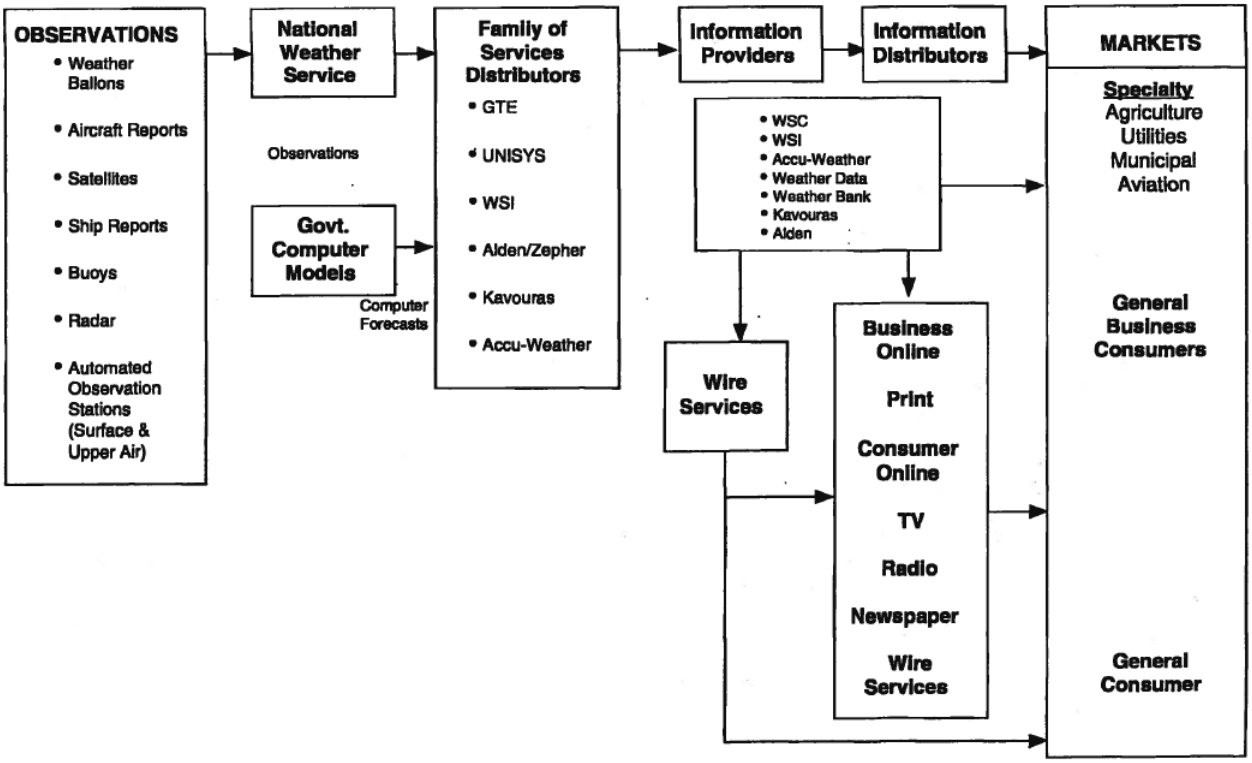
We also discussed the problem of finding optimal number of processors, assuming a multi-server processing facility where obsolescence only occurs in the database. The objective function, which is a sum of the cost of processing and the cost of time, were shown to be convex in number of processors, and numerical examples verified that the optimal value equals the value obtained by the square root formula widely used for call center design.

The immediate extension we can add is a capacity decision problem when the obsolescence occurs throughout the flow. Some asymptotic analysis using different staffing rules were presented in Garnett et al. (2002). Also, the decision problems

in this chapter were centered on processing stage of the information flow. However, there are other areas where planning is needed, from acquisition to purging, which incur cost and time. We can base the decision to acquire each item on the forecast of demand for that type of item, cost of having the item in the database, and so on. Purging the database can use some ideas similar to that of Papadimitriou (2004). It is certain that information chain and the content management is a topic with a room for extended research and improvement.

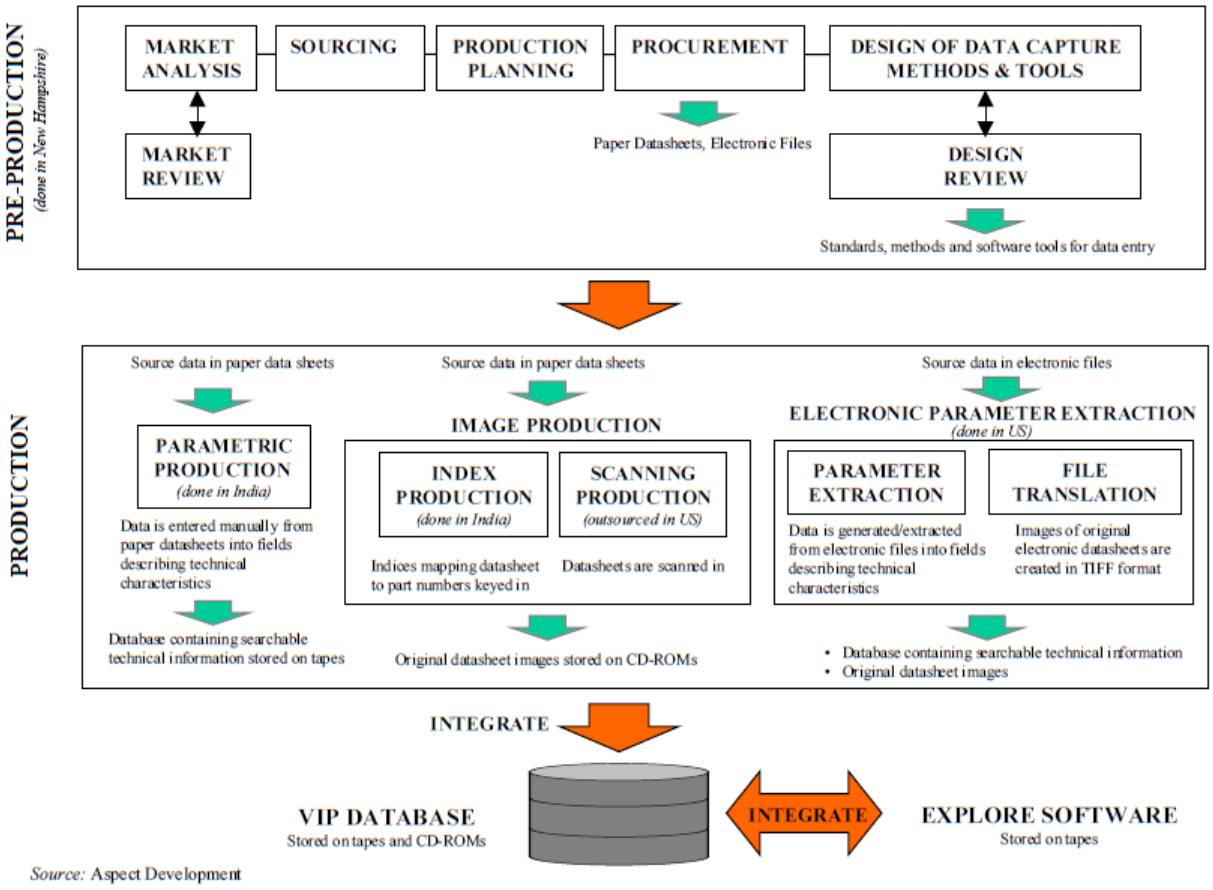
Appendix 1.1: Examples of Information Chains

A. Weather Information Industry [1]



[1] Connor (1998)

B. Aspect Development Production Process[2]



Source: Aspect Development

Appendix 1.2: Steady State Probabilities of 2D Markov Chain

Quasi-birth-and-death-process, as briefly described in section 1.3.4, is a Markov chain, in which transitions are allowed only to the neighboring levels or within the same level (see Osogami 2005). In Figure 1.3, N_p and N_d correspond to level and

[2]Bashyam and Karmarkar (2000)

phase, respectively. A generator matrix of a QBD process is of the form

$$Q = \begin{bmatrix} L^0 & F^0 & & & \\ B^1 & L^1 & F^1 & & \\ & B^2 & L^2 & F^2 & \\ & & \ddots & \ddots & \ddots \end{bmatrix},$$

where each entry is a submatrix. For levels $l \geq 0$, L^l contains transitions within level l, F^l contains transitions from level l to $l+1$, and B^l contains transitions from level l to $l-1$ (for $l \geq 1$). So from state (l, j), the process transitions to (l, k) at rate $[L^l]_{j,k}$, to $(l+1, k)$ at rate $[F^l]_{j,k}$, and to $(l-1, k)$ at rate $[B^l]_{j,k}$.

In our example in Figure 1.3, the submatrices will look like

$$L^l = \begin{bmatrix} -(\lambda+\mu+l\sigma) & & & \\ \sigma & -(\lambda+\mu+(l+1)\sigma) & & \\ & 2\sigma & -(\lambda+\mu+(l+2)\sigma) & \\ & & 3\sigma & \ddots \end{bmatrix},$$

$$F^l = \begin{bmatrix} \lambda & & & \\ & \lambda & & \\ & & \lambda & \\ & & & \ddots \end{bmatrix}, \qquad B^l = \begin{bmatrix} l\sigma & \mu & & \\ & l\sigma & \mu & \\ & & l\sigma & \mu \\ & & & \ddots \end{bmatrix},$$

where we treat the entry $(1, 1)$ as $(0, 0)$ in each matrix since in Figure 1.3, level and phase (N_p and N_d) start from 0.

From Kharoufeh (2011), assuming we have an irreducible generator matrix Q, the process is positive recurrent if and only if there exists a strictly positive solution to the system of eqautions

$$\vec{\pi_0}(L^0 + R^1 B^1) = \vec{\mathbf{0}}, \tag{1.12}$$

where $\vec{\pi_l}$ is a vector containing stationary probabilities whose i^{th} element denotes the stationary probability that the process is in state (l, i), and R^l is given recursively

Input: $\mathbf{F}$, $\mathbf{L}$, $\mathbf{B}$, ϵ
Output: $\mathbf{R}$, $\mathbf{G}$, $\mathbf{U}$

$\mathbf{H} = (-\mathbf{L})^{-1}\mathbf{F}$;
$\mathbf{K} = (-\mathbf{L})^{-1}\mathbf{B}$;
$\mathbf{G} = \mathbf{K}$;
$\mathbf{T} = \mathbf{H}$;
repeat
  $\mathbf{U} = \mathbf{HK} + \mathbf{KH}$;
  $\mathbf{M} = (\mathbf{H})^2$;
  $\mathbf{H} = (\mathbf{I} - \mathbf{U})^{-1}\mathbf{M}$;
  $\mathbf{M} = (\mathbf{K})^2$;
  $\mathbf{K} = (\mathbf{I} - \mathbf{U})^{-1}\mathbf{M}$;
  $\mathbf{G} = \mathbf{G} + \mathbf{TK}$;
  $\mathbf{T} = \mathbf{TH}$;
until $||\vec{1} - \mathbf{G}\vec{1}||_\infty \leq \epsilon$
$\mathbf{U} = \mathbf{L} + \mathbf{FG}$;
$\mathbf{R} = \mathbf{F}(-\mathbf{U})^{-1}$;

(a) repeating part

Input: $\mathbf{R}$, $\mathbf{G}$, $\mathbf{U}$
Output: $\mathbf{R}^{(\ell)}$'s, $\mathbf{G}^{(\ell)}$'s, $\mathbf{U}^{(\ell)}$'s

$\mathbf{U}^{(\hat{\ell}+1)} = \mathbf{U}$
$\mathbf{G}^{(\hat{\ell}+1)} = \mathbf{G}$
$\mathbf{R}^{(\hat{\ell}+1)} = \mathbf{R}$
for $\ell = \hat{\ell}$ to 1
  $\mathbf{U}^{(\ell)} = \mathbf{L}^{(\ell)} + \mathbf{F}^{(\ell)}\mathbf{G}^{(\ell+1)}$
  $\mathbf{G}^{(\ell)} = (-\mathbf{U}^{(\ell)})^{-1}\mathbf{B}^{(\ell)}$
  $\mathbf{R}^{(\ell)} = \mathbf{F}^{(\ell-1)}(-\mathbf{U}^{(\ell)})^{-1}$
end

(b) nonrepeating part

Figure 1.6: Algorithm for calculating R and R^l's

computed via

$$F^{l-1} + R^l L^l + R^l R^{l+1} B^{l+1} = 0.$$

The stationary probability vectors are given recursively by $\vec{\pi_l} = \vec{\pi_{l-1}} R^l$, so once R^l's are obtained, $\vec{\pi_l}$'s are calculated from $\vec{\pi_0}$, which must satisfy equation (1.12) and the normalization condition

$$\vec{\pi_0} \sum_{l=0}^{\infty} \prod_{i=1}^{l} R^i \vec{1} = 1.$$

For QBD processes that have infinite number of levels, the state space needs to be truncated so that R^l matrices can be calculated from a certain large enough integer L (see Bright and Taylor 1995). In Figure 1.6, the algorithm for obtaining R and R^l's are given. We can use F^L, L^L, B^L for some large L as the input matrices.

Chapter 2

Dynamic Capacity Planning for Content Management in Information Chains

2.1 Introduction

The rapid growth of information chains is part of the transition of the US from a material-based to an information-based economy (Apte et al. 2012). Just as physical goods are moved from stage to stage in supply chains, the flows in information chains are of data or information. One form of this data is experiential digital content such as songs, video streams, and images. Other examples include financial data, catalogs of parts and products, weather data, content portals, news, magazines and books.

As an example of content management, consider a publication like a magazine or newspaper. Articles from their own and from external sources "arrive" periodically, perhaps whenever some major event occurs. Some or all the arriving items may be selected, others declined. The selected items are then edited, amended, enhanced, formatted and processed to the appropriate distribution format, stored as a master copy, and made available to users through their distribution mechanism, whether broadcast, print or online. In the online case some or all of the data base may be accessible to end consumers. This sequence of stages from the arrival of data to processing to distribution in a form accessible to users is an information chain. Figure 2.1 shows a schematic example of these stages.

A distinct characteristic of such information chains that differentiates them from supply chains is that when the final product – information – is demanded and consumed, there is no depletion of inventory. It is not demand that reduces inventory, but eventual obsolescence and purging or updating of the data base. Content becomes obsolete either when it becomes false, or when its value to the consumer and hence the demand for it, declines. Examples of false content could be a listing in a guidebook for a restaurant that has since closed down, the current temperature at a particular location, or a price listing for an item for which the price has changed. Content types with declining value and declining demand over time, include music, videos, news, articles, blogs and books (fiction or non-fiction). False or erroneous

content is generally purged as soon as possible. For content with declining value, the decision to purge depends on a cost benefit comparison. In some cases, there may be no removal of items, but older content with lower value and demand, might be moved to a separate data category or data store, to reduce the size of the active data base and simplify search and access.

Another important characteristic of information chain is that often one cannot control or predict when a new item will arrive. There are some situations like weather services case where raw weather data is captured fairly regularly and predictably at monitoring points. A different situation holds for legal publishing, where a publisher has to process all the legal cases that arrive to have them available in a database within a required time (Karmarkar 2014). In yet another setting, a parts catalog publisher (Bashyam and Karmarkar 2000) must find, select and process component information from all relevant parts manufacturers as and when it becomes available, and then update the catalog database to have current information rapidly accessible to users. A case familiar to academics is that of a journal where papers are submitted by authors at any time. Here the accept/decline decision is of major importance, with emphasis on correctness and contribution, with less weight given to speed.

Information chains and content management in such chains, have characteristics that are in some ways like manufacturing and in other ways like services. On the one hand, processing is a back room operation like manufacturing. It is often possible to standardize some aspects of the process, the steps are not directly visible to the customer, and efficiency is an important factor especially when content volumes are high. On the other hand, processing is triggered by the arrival of content, rather than by orders or the need to replenish inventories. Furthermore the outputs are not standard pre-defined products. So production cannot be planned ahead based on forecasts, as in manufacturing. There is some similarity with agricultural product processing or water management, where arrival rates can be stochastic.

Since content is not pre-defined and standardized, advance production is not

possible, and information inventories do not provide a buffering function that allows for smoothing production levels in manufacturing. In that respect, this is more like a service situation, where buffering must be done through capacity. Consequently, capacity levels tend to be driven by the variations in content arrival and peak arrival rates rather than average demand rate. Finally as noted above, the size of the data base may be a positive factor for customers. This is a bit like retail inventories in certain supply chains where variety is a desirable property. In effect size is a proxy for variety or for "completeness" of the data base. (Bashyam and Karmarkar 2000)

Capacity planning is the process of determining production capacity to meet the changing demands. Just as lead time is a critical factor in manufacturing, timely processing of information is important for content management where information content is equivalent to physical goods in traditional supply chain. A typical though stylized flow of information in content management is shown in Figure 2.1. Items arrive and wait in a queue until processed by an available processor. Then they are loaded to the database which is available to customers. As noted above, demand does not deplete the inventory (database), nor can inventory act as a buffer. Holding costs are negligible, and data base size may actually be a desirable performance measure. However, there may be diminishing returns to size, and in some cases search and access costs may be relevant. Capacity of the processing stage determines the time it takes for each item from arrival till availability in database. Since data arrival is stochastic, processing capacity also affects the length of the queue and the size of the database. Content in the database become obsolete at some rate, either because it loses value over time or it becomes invalid or false. In general, content is periodically reviewed and purged to remove the obsolete content which is either erroneous, which erodes the perceived value of the data base, or which does not contribute to revenue generation.

We study how to make periodic decisions on capacity and processing by developing a discrete time multi-period model, in which capacity can be increased or

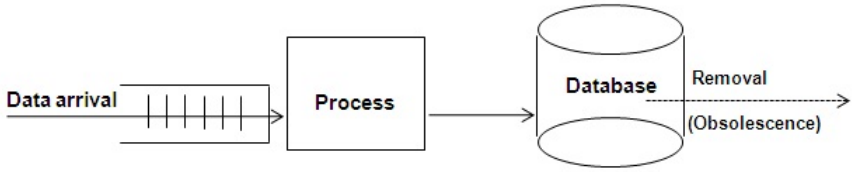


Figure 2.1: Flow of information content

decreased in each period by acquisition ("hiring") or reduction ("firing"). We assume that the performance characteristics of processors are known in terms of the output achievable as a function of the available capacity, and the current workload. The objective is to minimize the total cost of operations, including hiring, firing and processing costs, while maximizing the value of the database (measured in terms of size). The decision problem is formulated as a dynamic program with nonlinear constraints and mixed integer and continuous variables.

In the next section we review existing literature on capacity planning and processing. Then we formulate a general dynamic capacity planning model. We develop decomposition and relaxation approaches to provide bounds on the problem value, and propose heuristics. Next we describe a specific case study of content management for an online database, and illustrate the application of the methods to the example.

2.2 Literature Review

There are several streams of literature that relate to the present paper. The first is that of aggregate planning or seasonal planning in manufacturing setting. Traditional multi-period formulations of such models (c.f. Hax 1978, Bitran and Tirupati 1993) employ capacity constraints, and often include consideration of short term capacity planning in terms of workforce changes. These are typically LP models

which do not consider the effect of capacity and loading on processing lead times. Recently, clearing functions have been used to capture lead time and work-in-process consequences in the context of a deterministic model (Karmarkar 1989, Selcuk et al. 2008, Asmundsson et al. 2009, Armbruster and Uzsoy 2012). The dynamic factors in these planning models are due to demand variations over seasonal cycles, which are usually approximated as deterministic forecasts. Dobson and Karmarkar (2011) have included stochastic demand in a multi-period model with clearing functions, so as to be able to capture the interaction between capacity loading, lead times, and safety stocks (which depend on lead times).

A second stream of related literature is that addressing capacity planning in service settings. Many of these address the static stochastic case, including our own related work (Choi and Karmarkar 2014). We provide a brief review of relevant work in that paper.

The third stream of work is that addressing dynamic capacity decisions in service (non-manufacturing) settings. These papers include staffing decisions in call centers (Garnett et al. 2002, Borst et al. 2004) and in various service systems (Thompson 1997, Whitt 2007). One work that is related to ours in terms of methodology and some modeling aspects is that of Feldman et al. (2008) which is a deterministic multi-period model with capacity constraints, lead time effects due to congestion, budget limitations, resource allocation, and the choice of screening policies.

2.3 Model Formulation

Consider a content processing setting, where the decisions include determining capacity by determining the number of workers to hire or fire in each period. The available capacity constrains the number of items (volume) that are processed in each time period$t$, $t \in T = (1, ..., n)$. We assume that the number of data items arriving at the beginning of each period is known. In each period, seeing the newly

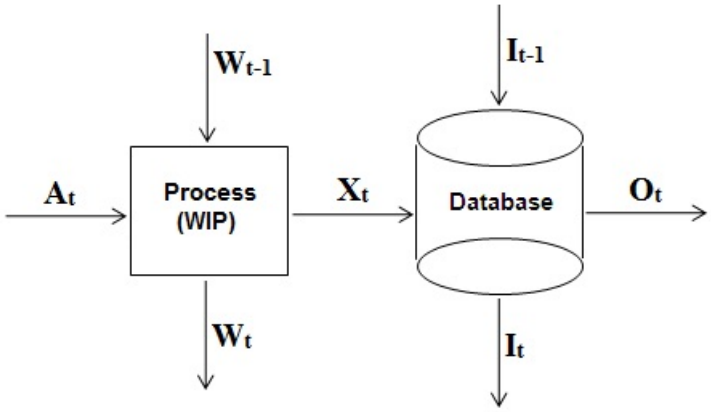


Figure 2.2: Material balance relationship

arriving items and the items from previous period waiting to be processed, we can hire more workers or fire some existing workers (i.e., add or remove processors) to satisfy certain requirements. We also decide how many items to process, which will then affect the size of the database. Processed items enter the database, and items left in the queue carry over into the next period. Figure 2.2 presents the material balance relationship for period t. We define the following decision variables:

M_t = number of workers used in period t.

H_t = number of workers hired in period t.

F_t = number of workers fired in period t.

W_t = work-in-process (WIP) at the end of period t.

X_t = number of items processed during period t.

We also have the following parameters that are given exogenously:

A_t = number of items that arrive at the beginning of period t.

L_t = current workload in period t.

I_t = size of the database at the end of period t.

O_t = number of items that become obsolete during period t.

$f(M_t, L_t)$ = the clearing function.

c_t = cost per worker in period t.

c_H = cost of hiring a worker.

c_F = cost of firing a worker.

p = processing cost per item.

$V(I_t)$ = value of the database of size I_t.

We use a clearing function (e.g. Karmarkar 1989, 2006) to determine the maximum possible throughput rate given the current workload and number of workers. This imposes an upper bound on the number of items processed. It is assumed that each worker is able to process work at the same rate (this assumption can be relaxed at the expense of a larger state space). The cost per worker can depend on the time period, since pay rates may be higher for overtime and weekend periods, and may also vary seasonally. The cost of hiring and firing includes administrative costs for those activities. We assume (Bashyam and Karmarkar 2000) that the value function $V(I)$ is concave increasing in I, the database size. We present the dynamic capacity planning problem as a dynamic program with continuous and integer variables. Of course, it is also possible to state the problem as a monolithic formulation. The objective is to minimize the sum of operating costs, hiring/firing costs and item processing cost, and to maximize the value of the database. We assume that the holding cost for content is negligible, or is captured in the function V. We also assume that there is a constraint on the average processing time for items to be processed in each period which captures a policy decision regarding the desired currency and availability of new content. We model this by setting an upper bound on the workload, divided by total processing capacity in that period (i.e. using Little's Law).

In the last period, we solve the following nonlinear mixed-integer program which we call CPP (capacity planning problem):

$$(CPP^n): \quad Z_n(M_{n-1}, W_{n-1}, I_{n-1}) \;=\; \min_{M_n, X_n} \Big\{ c_n M_n + c_H H_n + c_F F_n + p X_n - V(I_n) \Big\}$$

subject to

$$X_n \leq f(M_n, L_n) \tag{2.1}$$

$$L_n = W_{n-1} + A_n \tag{2.2}$$

$$W_n = W_{n-1} + A_n - X_n \tag{2.3}$$

$$W_n \leq W_{max} \tag{2.4}$$

$$I_n = I_{n-1} + X_n - O_n \tag{2.5}$$

$$M_n = M_{n-1} + H_n - F_n \tag{2.6}$$

$$W_n, X_n, I_n \geq 0 \tag{2.7}$$

$$M_n, H_n, F_n \in \mathbb{Z}^+. \tag{2.8}$$

The objective is to minimize the sum of operating cost, hiring/firing costs, and item processing cost, and maximize the value of the database. We assume that the holding cost is negligible, as storage cost for data is very minimal in general.

Constraint (2.1) sets an upper bound on the number of items processed in each period n. Constraint (2.2) defines the total workload in the period. Constraints (2.3) and (2.5) are the content balance equations corresponding to Figure 2.2. Observe that W_n follows directly from X_n, while H_n and F_n are determined once M_{n-1} and M_n are set, as given in constraint (2.6). There can be several different ways to include the quick processing requirement. Since in this discrete period model, we do not have the explicit expression for average lead time, or the time for an item to enter the database, we include constraint (2.4) to ensure that the queue does not build up beyond a predetermined threshold. Integrality and nonnegativity conditions are enforced by constraints (2.7) and (2.8). For period $t = 1, 2, ..., n-1$, we solve the following:

$$(CPP^t): \ Z_t(M_{t-1}, W_{t-1}, I_{t-1}) = \min_{M_t, X_t} \left\{ c_t M_t + c_H H_t + c_F F_t + pX_t - V(I_t) + Z_{t+1}(M_t, W_t, I_t) \right\}$$

subject to (2.1) - (2.8), with subscripts n replaced by t.

The optimal solution is obtained by solving problem (CPP^1). However, this dynamic program is difficult to solve due to the large state space since the number of items in the database can take on large set of values in many applications. It is unlikely for a realistic instance of the problem to be solved to optimality, as we describe in our computational result below. In some cases the number of workers can also be quite large. One possibility is to limit the size of the state space by only considering a finite number of data base states. Another is to treat the data base size as a continuous variable, and to use value function approximations. In the next section, we develop a Lagrangean decomposition of the problem that can be solved easily, and that provides lower bounds. We propose potential heuristic solution methods.

2.4 Problem Decomposition and Lower Bounds

In order to solve the problem by decomposition, we first look at problem (CPP^n). Observe that we are deciding *capacity* of the processing stage, and the output from the process. The two decision variables are linked together by the nonlinear constraint (2.1). We relax this constraint by introducing a Lagrange multiplier λ_n, and also relax (2.2) with a multiplier μ_n. Now the capacity decision problem (C^n) is separated from the output decision problem (T^n), and we get the following sub-problems for period n:

$$(C^n): \quad U_n(M_{n-1}, \lambda_n, \mu_n) \;=\; \min_{M_n, L_n} \left\{ c_n M_n + c_H H_n + c_F F_n - \lambda_n f(M_n, L_n) + \mu_n L_n - \mu_n A_n \right\}$$

subject to (2.6), (2.8), and $\lambda_n \geq 0$,

$$(T^n): \quad Y_n(W_{n-1}, I_{n-1}, \lambda_n, \mu_n) \;=\; \min_{X_n} \left\{ (\lambda_n + p) X_n - \mu_n W_{n-1} - V(I_n) \right\}$$

subject to (2.3), (2.4), (2.5), and (2.7). The relaxed problem (RP^n) is

$$(RP^n): \quad \tilde{Z}_n(M_{n-1}, W_{n-1}, I_{n-1}, \lambda_n, \mu_n) = U_n(M_{n-1}, \lambda_n, \mu_n) + Y_n(W_{n-1}, I_{n-1}, \lambda_n, \mu_n).$$

In (C^n), the Lagrange multiplier λ^n can be interpreted as the value gained per unit of increased output, whereas in (T^n) it can be interpreted as the extra cost for processing an additional item.

In period t, we solve:

$$(C^t): \ U_t(M_{t-1}, \lambda_t, \mu_t) = \min_{M_t, L_t} \left\{ c_t M_t + c_H H_t + c_F F_t - \lambda_t f(M_t, L_t) + \mu_t L_t - \mu_t A_t + U_{t+1}(M_t, \lambda_{t+1}, \mu_{t+1}) \right\}$$

subject to (2.6), (2.7), (2.8), and $\lambda_t \geq 0$,

$$(T^t): \ Y_t(W_{t-1}, I_{t-1}, \lambda_t, \mu_t) = \min_{X_t} \left\{ (\lambda_t + p) X_t - \mu_t W_{t-1} - V(I_t) + Y_{t+1}(W_t, I_t, \lambda_{t+1}, \mu_{t+1}) \right\}$$

subject to (1.11), (1.4), (2.5), and (2.7),

$$(RP^t): \quad \tilde{Z}_t(M_{t-1}, W_{t-1}, I_{t-1}, \lambda_t, \mu_t) = U_t(M_{t-1}, \lambda_t, \mu_t) + Y_t(W_{t-1}, I_{t-1}, \lambda_t, \mu_t).$$

To solve the capacity decision problem in the last period, we first define the clearing function using equation (2.23) from the Appendix. The following proposition characterizes problem (C^n).

Proposition 4. *The objective function of (C^n) is jointly convex in M_n and L_n.*

Proof. We have $f(M_n, L_n) = \frac{M_n r L_n}{\alpha + L_n}$, where r is the processing rate of a worker and $\alpha \geq 0$ is a constant.

$$\begin{aligned} \frac{\partial f(M_n, L_n)}{\partial M_n} &= \frac{r L_n}{\alpha + L_n}, \\ \frac{\partial f(M_n, L_n)}{\partial L_n} &= \frac{M_n r \alpha}{(\alpha + L_n)^2}, \\ \frac{\partial^2 f(M_n, L_n)}{\partial L_n^2} &= \frac{-2 M_n r \alpha}{(\alpha + L_n)^3} \\ &\leq 0. \end{aligned}$$

And we have $\frac{\partial^2 f(M_n, L_n)}{\partial M_n^2} = 0$, $\frac{\partial^2 f(M_n, L_n)}{\partial M_n \partial L_n} = \frac{r\alpha}{(\alpha + L_n)^2}$. Thus, Hessian is negative semidefinite, and we have that $f(M_n, L_n)$ is jointly concave in M_n and L_n. Then $-\lambda_n f(M_n, L_n)$ is a convex function and hence the objective function of (C^n), which is a sum of linear and convex functions, is convex in M_n and L_n. □

Starting in period n, we can solve (C^n) using convex optimization methods with specific choice of multipliers, and search the neighborhood of the obtained solution to find integer optimum M_n. Continuing recursively to period 1, we can obtain the solution with this dynamic programming algorithm of complexity $O(mn)$, where m is the maximum number of workers available in each period, which is not prohibitive for reasonable values of m and n. In problem (T^n), assuming that fixed fraction of items in the database becomes obsolete every period, we can replace constraint (2.5) with $I_n = \beta I_{n-1} + X_n$, with some β between 0 and 1. Then $V(I_n)$ is a concave function of X_n, and the objective function of (T^n) is convex in X_n. We solve (T^n) as a constrained convex optimization problem, and continue with $(T^{n-1}), ...(T^1)$. In each period, we need to determine the optimal solution for every possible pair of (W_t, I_t), which can involve a large number of computations compared to solving (C^t). To obtain tight lower bounds, we can use subgradient methods to solve the Lagrangian dual of (CPP^t),

$$LD^t = \max_{\lambda_t, \mu_t} \tilde{Z}_t(M_{t-1}, W_{t-1}, I_{t-1}, \lambda_t, \mu_t). \tag{2.9}$$

2.5 Heuristics and Upper Bounds

The complexity of solving the decomposed problem is less than that for solving the original formulation, but the solution might be infeasible, since the constraint relating throughput to the number of workers using a clearing function is relaxed in the decomposition. We introduce heuristic solution approaches below.

2.5.1 Myopic Heuristic

In the case of legal database compilation (Karmarkar 2014), managers attempt to bring down the queue daily, as it quickly builds up on days with large arrivals. In this heuristic, we simulate what the real data suggests about practice in the industry by processing as much content as possible in each period. In the beginning of period

t, given the current load in the queue (that is, $L_t = W_{t-1} + A_t$ in our notation), we determine the number of workers for the period as $M_t = \lceil L_t/r \rceil$, where r is the processing rate per worker. This is a heuristic approach for picking the appropriate number of workers when the only information available is the number of items in the queue. Once M_t is chosen, each worker processes as many items as possible, i.e., set $X_t = \lfloor f(M_t, L_t) \rfloor$. In this heuristic, cost is not an important concern while maximum throughput is; however, it could lead to high cost of hiring and firing.

2.5.2 Conservative Heuristic

In this heuristic, the goal is to process only enough to satisfy the constraint on throughput. For selecting the number of workers, we follow the same rule as in Myopic Heuristic, where $M_t = \lceil L_t/\mu \rceil$. Note that constraints (2.3) and (2.4) can be rewritten as $X_n \geq W_{n-1} + A_n - W_{max}$. We then choose the lowest X_n such that the inequality is satisfied. This will incur low processing cost, but does not aim to get the items into the database quickly.

2.5.3 Sequential Heuristic

Since this is a multiperiod problem, we make decisions in every period. We tackel the problem starting from the first stage. In period 1, pick M_1 between 0 and the max value from Myopic Heuristic, $\lceil L_1/r \rceil$. Also, for simplicity, assume that $V(I) = kI$ where k is a constant. For the first period, given M_1, we choose X_1 such that it is the maximum number possible less than $f(M_1, L_1)$, since $pX_1 - V(I_1) = (p-k)X_1$ for the first period, as seen in the objective function of (CPP^1). Here we assume that $k > p$.

In subsequent periods, we first proceed with following algorithm for deciding on M_t:

```
if (condition1 = false) {
   while (condition2 = true) {
      M_t = M_t − 1
   }
} else {
   while (condition1 = true) {
      M_t = M_{t−1} + 1
   }
}
```

In the above, *condition1* $= c_H + c_t < (k+1)\frac{\partial f(M_t,L_t)}{\partial M_t}$, *and condition2* $= c_F - c_t < (k+1)\frac{\partial f(M_t,L_t)}{\partial M_t}$. Basically, we will only be hiring/firing additional worker if the benefit outweights the cost. Once we choose M_t, we can pick X_t in the same way we did for period 1. We assume obsolescence of items is negligible in this process.

2.5.4 Computational Results

We present a computational study to evaluate the performance of the heuristics. Based on arrival information we have from online citations company, we generated 50 sample data sets, each having 52-week time horizon comprising of 52 periods with arrivals in each week. Arrival rates are randomly generated based on a given range, and parameters were chosen among a range of reasonable values. Table 2.1 summarizes the parameters and their values used in the samples. We solved the optimization problems using Bonmin on NEOS server using AMPL[1]. Bonmin (Basic Open-source Nonlinear Mixed INteger programming) is an open-source C++ code for solving general MILP, and features several algorithms including branch-and-bound and decomposition algorithms[2].

To validate the need for heuristics, we submitted a test problm to the server

[1]http://www.neos-server.org/neos/solvers/minco:Bonmin/AMPL.html
[2]https://projects.coin-or.org/Bonmin

without using any heuristic. A 20-period problem in its original formulation from section 2.3 with a sample set of parameters picked from ranges defined in Table 2.1 ran for 24 hours on NEOS server, and returned with incomplete result error.

As a measure of performance, we compare our heuristic solutions to the Lagrangean lower bound from equation (2.9), which we denote LB_{Lg}. We calculated UB_1 through UB_3, which represent upper bounds obtained from myopic, conservative, and sequential heuristic. Just for the purpose of comparison, we include UB_{LP}, which is obtained by relaxing integrality constraints from the original problem (CPP), although this will almost always return infeasible solution. Following is the summary of results over all data sets, showing average cost and suboptimality gap relative to the Lagrangean lower bound, which is calculated as $\frac{UB_i - LB_{Lg}}{LB_{Lg}}$ for each heuristic i.

	UB_1	UB_2	UB_3	UB_{LP}
Average cost	377620	352943	339806	335267
Gap (%)	25.9	17.6	13.3	11.8

Among our three heuristics, sequential heuristic outperformed the other two with relatively small suboptimality gap. In the next section, we look at a problem faced by a real database company dealing with daily workload.

2.6 Auto-Cite Case Simulation

The Auto-Cite product was an on-line data base containing citations and references to legal opinions from Federal and State courts. The company processes the incoming legal opinions that are either electronic or on print and logs an entry into

Parameter	A_t	$f(M_t, L_t)$	c_t	c_H	c_F	p	$V(I_t)$
Value range	(1000,6000)	$\frac{5M_tL_t}{100+L_t}$	(10,15)	(20,200)	(20,100)	(1,5)	kI, $k \in (5, 10)$

Table 2.1: Parameters used in sample data sets

their database in predefined format. This workflow involves manual inputs from various teams. To be competitive in the citations services market, maintaining quick turnaround time from publishing of an opinion to when it was accessible in their database for clients was very important. (Karmarkar, 2014)

A new batch of information arrives to Auto-Cite six days a week at variable rates, but the company does not have workers on Saturday to process the new information coming in that day, and thus usually sees a high pile of unfinished cases on Monday. There is a tradeoff between cost of resources and speed of processing. We want to determine the optimal capacity and optimal number of cases to process in each period.

We start with an assumption that each week consists of two periods, a "weekdays" period and a weekend period, which we will distinguish using superscripts w and e, respectively. The two periods will not be equal in length, since a weekend period only consists of one day while there are five days in the weekdays period. However, this definition will be useful in capturing different characteristics in staffing full-time workers and part-time workers. We are given the number of cases that arrive in each period. Full-time workers, who work fixed hours, work only in weekday periods, and part-time workers only work in the weekend periods. The company has to decide how many workers of each type to use in each period for week 1, 2, ..., N. We introduce the following variables.

Parameters:

N =number of weeks in the planning horizon,

s_f = hourly salary of a full-time worker,

s_p = hourly salary of a part-time worker,

c_H = cost of hiring a worker,

c_F = cost of firing a worker,

r = processing rate for all workers (cases/hr),

Q_t^w =number of items in the queue at the beginning of period w in week t,

Q_t^e =number of items in the queue at the beginning of period e in week t,

A_t^w = number of items that arrive at the beginning of period w in week t,

A_t^e = number of items that arrive at the beginning of period e in week t,

L_t^w = workload in period w in week t,

L_t^e = workload in period e in week t,

$\bar{T}$ =upper bound for turnaround time,

$f(W, L)$, $f(H, L)$ =function of capacity and load for providing upper bound in processing.

Decision variables:

W_t = number of full-time workers used in week t,

H_t = number of part-time hours used in week t,

nh_t =number of workers hired in period t,

nf_t = number of workers fired in period t,

P_t^w = number of items processed during period w in week t,

P_t^e = number of items processed during period e in week t.

We assume the same processing rate for both full-time and part-time workers. However, hourly wage of a part-time worker is higher than that of a full-time worker, since part-time is fixed to the weekend periods. The objective of the company is to minimize the cost of workers while meeting certain turnaround time goal for the cases, which is formulated as:

$$\min_{W_t, H_t} \sum_{t=1}^{N} 40 s_f W_t + s_p H_t + c_H nh_t + c_F nf_t$$

subject to

$$Q_1^w = Q_{N+1}^w \tag{2.10}$$

$$H_1 = H_N \tag{2.11}$$

$$W_1 = W_{N+1} \tag{2.12}$$

$$P_t^i \leq \min(L_t^i, f_i(W_t, L_t^i)) \qquad \text{for } i = w, e,\ t = 1...N \tag{2.13}$$

$$L_t^i = Q_t^i + A_t^i \qquad \text{for } i = w, e,\ t = 1...N \tag{2.14}$$

$$Q_t^w = L_{t-1}^e - P_{t-1}^e \qquad \text{for } t = 2...N+1 \tag{2.15}$$

$$Q_t^e = L_t^w - P_t^w \qquad \text{for } t = 1...N \tag{2.16}$$

$$W_t = W_{t-1} + nh_{t-1} - nf_{t-1} \qquad \text{for } t = 2...N+1 \tag{2.17}$$

$$\frac{L_t^w + A_t^e}{P_t^w + P_t^e} \leq \bar{T} \qquad \text{for } t = 1...N \tag{2.18}$$

$$W_t, H_t, P_t^i, Q_t^w \in \mathbb{Z}, \qquad \text{for } i = w, e,\ t = 1...N. \tag{2.19}$$

Constraints (2.10), (2.11), and (2.12) impose a requirement for the wrap-around solution. By wrapping around the last period to the first period, end-of-term approximation for the last period is no longer needed, and the solution represents an equilibrium of the system. Constraint (2.13) is setting an upper bound to the number of items processed using a clearing function in the same way as in section 2.3. Constraints (2.14) to (2.17) are flow balance equations for each period, and constraint (2.18) imposes a turnaround time target by using the ratio of current load to current throughput as a measure.

2.6.1 Four-week Problem

To get an idea of the tradeoff between turnaround time and cost, we start by solving instances of this problem with manageable size. Each instance has four-week horizon and different parameters for the target turnaround time $\bar{T}$. Table 2.2 lists the parameters used in the instances. We assumed hourly wage of a part-time worker

	A_t^w	A_t^e
$t = 1$	2000	300
$t = 2$	3000	600
$t = 3$	5000	800
$t = 4$	1000	300

N	r	s_f	s_p	c_H, c_F
4	5	10	15	200

Table 2.2: Parameters used in the 4-week example

who works on weekend to be 50% greater than that of a full-time worker. The cost of hiring and firing were set substantially higher than the hourly wage, since adjusting the number of workers during the week involved manual work at Auto-Cite.

In cases 1 and 3, no part-time hours are allowed. We varied the upper bound for turnaround time between the first two cases and the last two to see the variation in cost. The results are as follows:

Case	1	2		3	4	
$\bar{T}$	2.5	2.5		1.25	1.25	
Cost	27200	26830		33600	31690	
Staff	full-time	full-time	part-time(hrs)	full-time	full-time	part-time(hrs)
1	17	16	1	12	13	1
2	17	16	1	18	16	25
3	17	16	79	27	16	459
4	17	16	1	12	13	1

Allowing part-time resource leads to cost savings, and the savings is greater between cases 3 and 4 with over 5 percent, compared to about 2 percent in the first two cases where turnaround time bound is set twice as large. Although the arrival load differed from week to week, the number of full-time workers is consistent over the weeks in cases 1 and 2 due to the high cost of hiring and firing.

As a comparison, we then solved the same instances, with the cost of hiring and firing reduced to one-tenth. Table 2.3 shows the new results. Now in all cases,

Case	1	2		3	4	
$\bar{T}$	2.5	2.5		1.25	1.25	
Cost	26850	26830		27440	27435	
Staff	full-time	full-time	part-time(hrs)	full-time	full-time	part-time(hrs)
1	16	16	1	11	12	1
2	18	17	1	18	17	1
3	14	17	1	27	26	2
4	16	16	1	11	12	1

Table 2.3: Results when $C_H = C_F = 20$

N	r	s_f	s_p	c_H, c_F
20	5	10	15	200

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A_t^w	3	4	5	2.5	5.5	3.5	3	4	2	3	5	4	2	2.5	5.5	6	4	2.5	3.5	2
A_t^e	5	6	7	3	2.5	6	8	4	2.5	5	5	6	7	3	2.5	3	4	2.5	4.5	5

Table 2.4: Parameters used in the 20-week example (A_t^w in 000's, A_t^e in 00's)

number of full-time workers varies along the planning horizon, and the value of having part-time resource is not as pronounced with a small cost of hiring and firing.

We next look at a problem with a longer horizon with representative data from Auto-Cite case.

2.6.2 Twenty-week Problem

Auto-Cite has varying daily arrival rate. From the daily arrival data, we take sum of daily arrivals for each weekday period and weekend period over the period of one year to come up with A_t^w and A_t^e for our 20-week planning horizon. Table 2.4 shows the arrival rates and other parameters used in this problem.

We compared three different upper bounds for the turnaround time under this setting. Even-numbered cases had part-time workers included. The cost for each

case is displayed below.

Case	1	2	3	4	5	6
$\bar{T}$	5	5	2.5	2.5	1.25	1.25
Part-time?	N	Y	N	Y	N	Y
Cost	172600	172100	179800	175500	189196	180024
Savings		0.3%		2.4%		4.8%

As seen in the 4-week problem before, the benefit from having part-time resources is maximized when the target turnaround time is tight. Since this was the case for Auto-Cite which constantly competes on time with other database products in the market, incorporating Saturday part-time hours into their current scheduling policy can reduce their total cost by considerable amount.

2.7 Conclusions and Future Research

In this chapter we study capacity planning and processing planning for information content arriving periodically with a variable pattern. We model the problem as a mixed integer nonlinear program to minimize the cost of workers including hiring and firing costs and the cost of processing, while maximizing the value of the database as a function of database size . The problem was decomposed into two sub-problems for each period using Lagrangian relaxation, to obtain a lower bound to the optimal value. Each decomposed problem is a dynamic program that can be solved as convex optimization problem. We then proposed three heuristics, which provided upper bounds to the minimization problem.

In addition, we examined a specific problem faced by a legal citations database company (Auto-Cite, Karmarkar 2014), and showed that they can reduce the total cost by hiring part-time workers on weekends when they have to meet tight turnaround time requirement.

Another approach can be used for solving the problem, in which we linearize the nonlinear constraint linking the throughput and the number of workers. This

can be a possible extension to the current solution method. We can also modify how obsolescence is modeled to better capture realistic scenarios. The processing time requirement can also be included in a separate constraint that involves average system time, which is more complicated than simply setting an upper limit on the work-in-process, since we need to find out a way to capture the residence time of each processed item. The model presents many more opportunities for extensions and future research.

Appendix 2.1: Clearing Function

We would like to approximate the relationship between the average work-in-process (WIP) and the throughput rate for the system when there are multiple servers processing. Under $M/M/1$ assumption (i.e., items arrive according to Poisson process and get processed by a single server with exponentially distributed service time), clearing function can be derived using a few steps using Little's Law (as in Karmarkar 1989) to express the throughput as a function of WIP since there exists simple expression for average time in queue. Under $M/M/m$ assumption with arrival rate λ and service rate μ, however, the equation involving the expected throughput rate X, and the WIP W, assuming $\lambda = X$, becomes

$$\begin{aligned} W &= XL \\ &= X(T_q + \frac{1}{\mu}) \\ &= X\left(\frac{1}{m\mu - X}P(m^+) + \frac{1}{\mu}\right), \end{aligned} \tag{2.20}$$

where L is the leadtime, or the time in system, and T_q denotes the average time an item spends in queue. The third equality comes from the standard expression for the time in queue for $M/M/m$ queues, with probability of waiting (more than m items in the system) denoted $P(m^+)$. We cannot derive a closed from expression for X from equation (2.20) above.

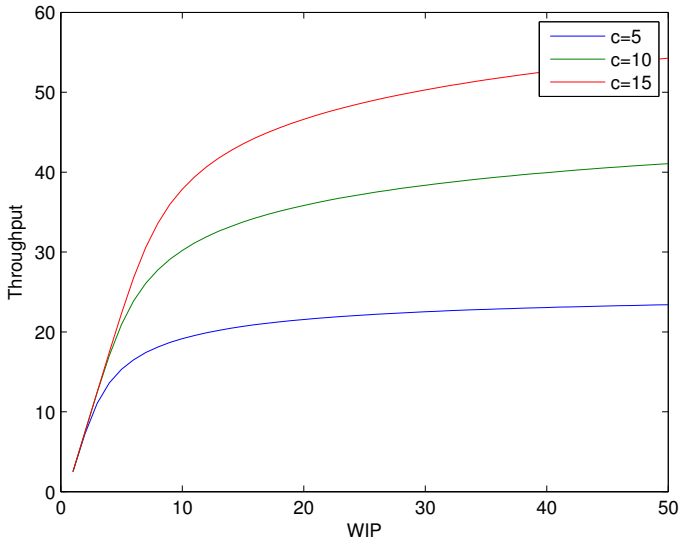


Figure 2.3: Throughput vs. WIP ($\mu = 5$)

We turn to the heavy traffic theory (see Köllerström 1974), which states that when the traffic intensity is less than, but close to, unity, the waiting time is approximately exponential with mean $\frac{var(s/k)+var(t)}{E(t)-E(s/k)}$, where s and t are service and arrival processes, respectively. Applying this to our $M/M/m$ scenario, the mean waiting time is

$$T_q = \frac{\lambda^2 + m^2\mu^2}{m\lambda\mu(m\mu - \lambda)}. \tag{2.21}$$

Combining with equation (2.20), we obtain

$$X = f(m, W) = \frac{m\mu(W + m) \pm m\mu\sqrt{(W + m)^2 - 4(m - 1)(W - 1)}}{2(m - 1)}. \tag{2.22}$$

Figure 2.3 illustrates the relationship between throughput and WIP for an arbitrary service rate, for different number of processors (denoted as c in the graph). The approximated function of throughput appears concave increasing in WIP as expected. To simplify the computation for CPP, we adopt the form used in Dobson and Karmarkar 2011 with

$$f(M, L) = \frac{MsL}{\alpha + L}, \tag{2.23}$$

where s is a service rate of a worker in a period and $\alpha \geq 0$ affects the shape of the function. For other forms of clearing functions, Armbruster and Uzsoy (2012) has an extensive discussion on various models where different types of clearing functions are used.

Chapter 3

Competing on Price and Release Time for Information Content

3.1 Introduction and Literature Review

Information chains acquire, process and distribute content in a way analogous to physical goods in supply chains. One of the aspects that distinguishes information chains from supply chains is that information may be held in "inventory" for a long time, since it does not deplete with sales or physically decay. However, information nonetheless can lose value over time. Examples that clearly show such characteristics include digital music, video content, financial data and more. In the previous chapters, we have studied problems in the capacity planning stage of the chain related to size of the content database and the processing time until content is available to users. In this chapter we look at distribution stage of the information chain, which comes after the acquisition, processing, and storage steps.

The most common and critical decision factor in the distribution stage for a supplier is pricing the goods, or the content in the case of information chains. While there is a large stream of literature addressing pricing and revenue management for products and capacities, there appears to be less work specifically directed towards online digital content. Mendelson and Whang (1990) derived an incentive-compatible pricing scheme for a service facility involving multiple classes of users, such as a computer system, a production line, or a communication facility. Cocchi et al. (1993) studied pricing in computer networks and suggested that optimal pricing of network services require that consumers are charged on the basis of service that they desire. Both works focus on optimizing the efficiency of the system or network. Jain and Kannan (2002) studied issues in pricing information products, which involved online servers choosing between search-based and subscription-fee pricing. They present conditions where subscription-fee pricing is optimal and also find that online servers can compete in the market each making profit. In the early years of the internet, network pricing problems addressed network congestion and loads which were not negligible at the time.

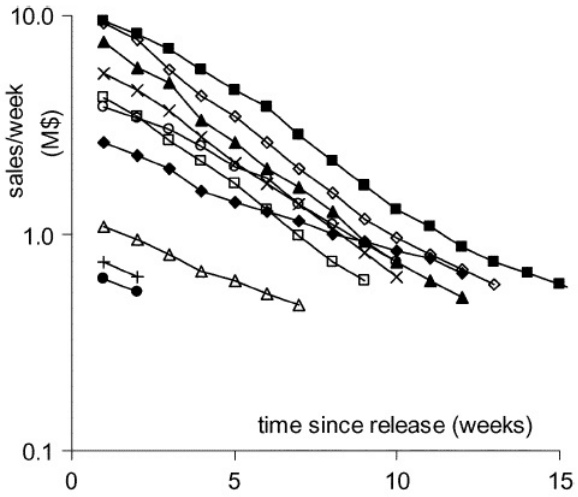


Figure 3.1: Plot of DVD sales on time

We study the pricing problem for information content in conjunction with deciding when to release the content to the market. Figure 3.1 shows a graph of sales revenue for 10 different DVD titles plotted weekly from the time they were released. The speed of reduction in sales is different for each title, and it depends on the distribution of consumer interest and sensitivity, but it is generally the case that popularity of content decreases with time and that release time affects the market demand. The importance of catering to individual needs has been increasing, and with modern technology, some providers like Netflix succeed in attracting customers by making recommendations based on past choices (Bennett and Lanning 2007). Recently, August et al. (2013) studied optimal timing of distributing a film's theatrical and video releases. They present a consumer choice model that examines trade-offs between substitutable products. Their model involves a market condition where a monopolist studio makes release time and price selections such that equilibrium strategies of the consumers are satisfied.

In general, for most content, releasing early is favored especially in secondary markets like Netflix. As discussed in Chapter 2, however, there is cost associated

with processing information content, and the cost for having shorter processing time is higher in general. We develop a model in which market demand depends on the utility of each individual customer, which is a function of price and release time. Our objective is to study the optimal decision suppliers have to make in order to maximize profit (or increase market share in case of more than one supplier). For a monopolist supplier, we define the optimal price and release time decision under different assumptions on customers' utility functions. We then study a competitive situation in which a supplier sells the content to two downstream channels who distribute it to the market. The problem of two channels deciding on price and release time for the content is a two player game with multiple parameters, a setting analogous to product and price competition in Moorthy (1988). We characterize conditions under which equilibrium prices exist.

3.2 Model Formulation - Monopolist Provider

For suppliers of information goods such as video content, the age of the content is as important as price, since customers are very sensitive to the currency of the content that they purchase. The value of the content declines as it ages, but getting the content available sooner requires more work for the provider. We define p and t to be the price and release time of the content respectively, which are decision variables that the provider sets. We assume that there is cost associated with a release at time t, denoted $c(t)$. Furthermore, there is a fixed cost for acquiring the content in addition to variable cost of processing which is proportional to the service rate, or processing rate of the content, denoted μ. If we impose, for simplicity, the Poisson arrival and service process assumption, we have from Little's Law that $\bar{t} = \frac{1}{\mu - \lambda}$, or $\mu = \frac{1}{t} + \lambda$. Thus we obtain that $c(t) = c_f + k(\frac{1}{t} + \lambda) = F + \frac{k}{t}$, with a given fixed cost F and a positive constant k representing the cost for service rate. This cost implies that the cost of instantaneous release (i.e., $t = 0$) is prohibitively large.

Given a price and a release time, each customer has an associated utility function that depends on personal parameters. We look at the problem faced by a profit-maximizing content provider, starting with a monopolist model.

3.2.1 Linear Decay: Single Release

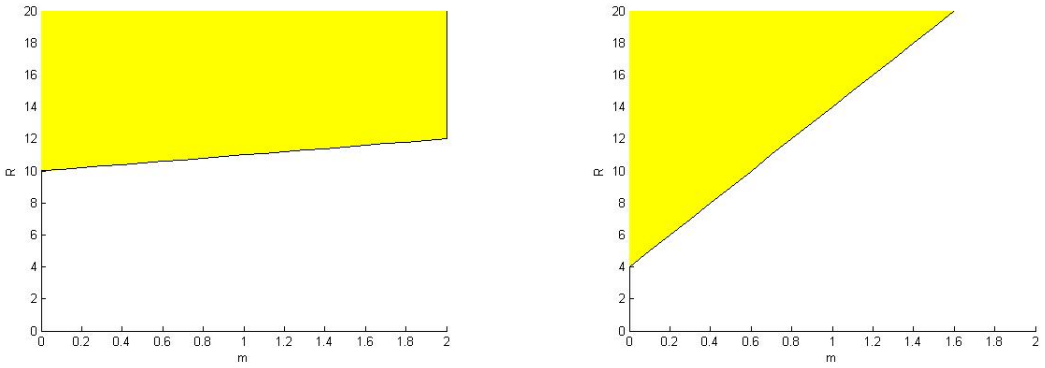
A monopolist content provider sets price and release time of the product to maximize profit. We assume that the demand for content is a certain fraction of the total market size M, and that this fraction depends on the distribution of customer preferences. Let $S(p,t) = \{\theta | u(p,t|\theta) \geq 0\}$, where θ represents a customer-specific characteristic (details will be discussed later). Thus a value θ is in set S if and only if the customer having that characteristic has nonnegative utility associated with given price and release time. We can then define demand as

$$D(p,t) = M \int_{\theta \in S} dF(\theta), \tag{3.1}$$

where $F(\theta)$ is the cumulative distribution function of θ. Then the profit maximization problem for monopolist provider can be stated as

$$\max_{p,t} \Pi(p,t) = pD(p,t) - c(t). \tag{3.2}$$

We use a specific utility function to introduce details to the above problem. Let $u(p,t) = f(t) - p$, where $f(t)$ can be viewed as a price each customer is willing to pay when the content has age t at time of release. We first consider a utility function which declines linearly with time, with $f(t) = R - mt$. R is the reservation price for when the content is released immediately (i.e., $t = 0$), and m is the impatience parameter of the customer, which is nonnegative. The higher the value of m, the faster the customer's utility decreases as t increases. Hence, R and m are the equivalent of θ defined before, and set S is then defined as $S(p,t) = \{(R,m) | R \geq p + mt\}$. Figures 3.2a and 3.2b illustrate "Buy Regions," colored areas representing the set of



(a) $S(10, 1)$ (b) $S(4, 10)$

Figure 3.2: Buy Regions for different price-time pairs.

customers (i.e., (R, m) values) who have positive utility given a certain price and release time.

Let $F(R, m)$ denote the joint cdf of the two parameters, and suppose that customers' reservation price and impatience parameter are independent and are distributed uniformly from 0 to $\bar{R}$ and 0 to $\bar{m}$, respectively. For tractability, we assume $\bar{R} > k$ and $\bar{R} > \bar{m}$. With $p \in (0, \bar{R})$ and $t > 0$, we can rewrite equation (3.1) for demand as

$$\begin{aligned} D(p,t) &= M \int dF(R,m) \qquad (3.3) \\ &= \begin{cases} \frac{M}{\bar{R}\bar{m}}(\bar{R} - p + \bar{R} - p - \bar{m}t)\bar{m}/2 = M(1 - \frac{2p+\bar{m}t}{2\bar{R}}) & \text{when } p \le \bar{R} - \bar{m}t, \\ \frac{M}{\bar{R}\bar{m}}(\bar{R} - p)(\frac{\bar{R}-p}{t})/2 = \frac{M(\bar{R}-p)^2}{2\bar{R}\bar{m}t} & \text{when } p > \bar{R} - \bar{m}t. \end{cases} \end{aligned}$$

We formulate a two-stage problem by separating price and time in the profit function. Let $V_p(t) = \max_p[pD(p,t)]$ represent the maximum revenue for a given release time t. Then equation (3.2) can be written as

$$\max_{p,t} \Pi(p,t) = \max_t [V_p(t) - c(t)]. \qquad (3.4)$$

For a given t, we determine the revenue maximizing price $p^*(t)$, which is characterized in the following proposition.

Proposition 5. *For any $t > 0$, the revenue $pD(p,t)$ is concave in p. If $t < \frac{2\bar{R}}{3\bar{m}}$, the revenue maximizing price is $p^*(t) = \frac{2\bar{R}-\bar{m}t}{4}$; otherwise, we have $p^*(t) = \frac{\bar{R}}{3}$.*

Proof. From equation (3.3), when $p \leq \bar{R} - \bar{m}t$ we have $\frac{\partial}{\partial p}\big(pD(p,t)\big) = M(1 - \frac{4p+\bar{m}t}{2\bar{R}})$, and $\frac{\partial^2}{\partial p^2}\big(pD(p,t)\big) = -\frac{2M}{\bar{R}} < 0$, which shows that the revenue function is concave in p in this region. Solving the first order condition gives $p^*(t) = \frac{2\bar{R}-\bar{m}t}{4}$, and if $\frac{2\bar{R}-\bar{m}t}{4} < \bar{R} - \bar{m}t$, or equivalently, $t < \frac{2\bar{R}}{3\bar{m}}$, revenue is maximized at $p^*(t)$. On the other hand, when $p > \bar{R} - \bar{m}t$ we have $\frac{\partial}{\partial p}\big(pD(p,t)\big) = \frac{M(\bar{R}-p)(\bar{R}-3p)}{2t\bar{R}\bar{m}}$, and $\frac{\partial^2}{\partial p^2}\big(pD(p,t)\big) = \frac{M(3p-2\bar{R})}{t\bar{R}\bar{m}}$, which shows concavity of revenue function for $p \leq \frac{2\bar{R}}{3}$. First order condition gives $p^*(t) = \frac{\bar{R}}{3}$. Similarly, if $\frac{\bar{R}}{3} \geq \bar{R} - \bar{m}t$, or $t \geq \frac{2\bar{R}}{3\bar{m}}$, then revenue is maximized at $p^*(t) = \frac{\bar{R}}{3}$. Note that for $t < \frac{2\bar{R}}{3\bar{m}}$, $p > \bar{R} - \bar{m}t > \frac{\bar{R}}{3}$ and first derivative is negative, which completes the proof that the revenue function is concave in p for any $t > 0$. □

It is interesting to note that for t above a threshold, optimal price does not depend on t. You can see from Figure 3.2b and the second part of equation (3.3) that when t is large, the revenue is less, and the derivative of the demand does not depend on t. By obtaining $V_p(t)$, the problem from equation (3.3) then becomes

$$\max_t[V_p(t) - c(t)] = \begin{cases} \max_t \left[\frac{M(2\bar{R}-\bar{m}t)^2}{16\bar{R}} - F - \frac{k}{t}\right] & \text{when } t < \frac{2\bar{R}}{3\bar{m}}, \\ \max_t \left[\frac{M\bar{R}(\bar{R}-\bar{R}/3)^2}{6t\bar{R}\bar{m}} - F - \frac{k}{t}\right] & \text{otherwise,} \end{cases} \tag{3.5}$$

and we can characterize the profit maximizing t as follows.

Proposition 6. *Assuming market size M is sufficiently large, profit $\Pi(t)$ is convex decreasing in t for $t \geq \frac{2\bar{R}}{3\bar{m}}$; and for $t < \frac{2\bar{R}}{3\bar{m}}$,* $\Pi(t)$ is concave for $t < \sqrt[3]{\frac{16k\bar{R}}{M\bar{m}^2}}$ and convex decreasing for $t \geq \sqrt[3]{\frac{16k\bar{R}}{M\bar{m}^2}}$, and it reaches a maximum in this region.

Proof. Based on equation (3.5), for $t \geq \frac{2\bar{R}}{3\bar{m}}$, $\Pi'(t) = \frac{27\bar{m}k - 2M\bar{R}^2}{27\bar{m}t^2} < 0$, and $\Pi''(t) = \frac{4M\bar{R}^2 - 54\bar{m}k}{27\bar{m}t^3} > 0$. For $t < \frac{2\bar{R}}{3\bar{m}}$, $\Pi(t) = \frac{M(2\bar{R}-\bar{m}t)^2}{16\bar{R}} - F - \frac{k}{t}$, and we have $\lim_{t\to 0^+}\Pi(t) =$

$-\infty$, with $\lim_{t\to 0^+}\Pi^{'}(t) > 0$. Also, $\Pi^{''}(t) = \frac{M\bar{m}^2}{8\bar{R}} - \frac{2k}{t^3}$, which shows that $\Pi(t)$ is concave for $t < \sqrt[3]{\frac{16k\bar{R}}{M\bar{m}^2}}$, and convex thereafter. At the boundary, we have $\Pi(\frac{2\bar{R}}{3\bar{m}}) = \frac{2M\bar{R}^2-27\bar{m}k}{18\bar{R}} > 0$, and $\Pi^{'}(\frac{2\bar{R}}{3\bar{m}}) = \frac{\bar{m}(27\bar{m}k-2M\bar{R}^2)}{12\bar{R}^2} < 0$. Thus, maximum profit is obtained at $t^* < \frac{2\bar{R}}{3\bar{m}}$ satisfying $\Pi^{'}(t^*) = \frac{M\bar{m}^2t}{8\bar{R}} - \frac{M\bar{m}}{4} + \frac{k}{t^2} = 0$. □

To find t^*, we can solve the first order condition numerically, a cubic equation in t.

3.2.2 Linear Decay: Multiple Releases

Consider a monopolist provider who now tries to capture different segments of customers in the market by introducing multiple releases. For example, a movie distribution company can first release the movie at theaters (high price and low release time), and may later release it on DVD at a lower price. The set of customers company can capture will grow with each additional release, since $S_1(p_1, t_1) = \{(R, m)|R \geq p_1 + mt_1\}$ and $S_2(p_2, t_2) = \{(R, m)|R \geq p_2 + mt_2\}$, which gives $S_1 \subset S_1 \cup S_2$, as shown in Figure 3.3. $S_1 \cup S_2$, the entire shaded area in the figure represents the group of customers whose (R, m) values fall within the set. Among the group, however, who will purchase the product at the first release and who will purchase at the next release is not immediately observed.

We first assume that the provider will announce the times and prices of two releases to customers, and assume that each customer will buy once to maximize utility based on the information given. Recall that the definition of utility was $u(p, t) = R - mt - p$. Thus, when consumers are well-informed, they will choose to buy a release (i.e., pick a price-time pair) which gives higher utility. Let R_1 and R_2 represent the set of customers who purchase at the first release and the set of those who purchase at the second release, respectively. Given p_1, t_1, and p_2, t_2, customers belong to R_1 if $(R,m) \in S_1(p_1, t_1)$ and $u(p_1, t_1) \geq u(p_2, t_2)$, or $(R, m) \in S_1(p_1, t_1)$ and $m \geq \frac{p_1-p_2}{t_2-t_1}$. Similarly, customers belong to R_2 if $(R, m) \in S_2(p_2, t_2)$ and $m < \frac{p_1-p_2}{t_2-t_1}$.

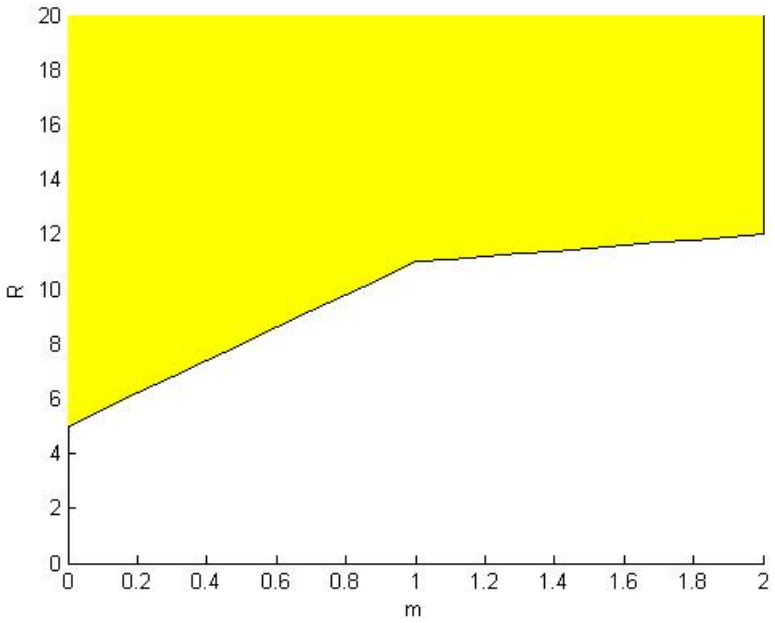


Figure 3.3: $S_1(10,1) \cup S_2(5,6)$

Intuitively, whether they choose to buy at the first or second release depends on the impatience of customers.

Figure 3.4 shows two segments of customers R_1 and R_2 as blue and yellow areas, and we can see that in the case of two releases, some of the customers who belong to S_1 will wait to buy at the second release at lower price p_2. Demands from two segments D_1 and D_2 can be derived using the general equation (3.1). Let $m^* = \frac{p_1 - p_2}{t_2 - t_1}$, and calculating the demands is as follows:

$$\begin{aligned} D_1(p_1, t_1) &= \frac{M}{\bar{R}\bar{m}}(\bar{R} - p_1 - m^* t_1 + \bar{R} - p_1 - \bar{m} t_1)(\bar{m} - m^*)/2 \\ &= \frac{M}{2\bar{R}\bar{m}}(2\bar{R} - 2p_1 - m^* t_1 - \bar{m} t_1)(\bar{m} - m^*), \end{aligned} \tag{3.6}$$

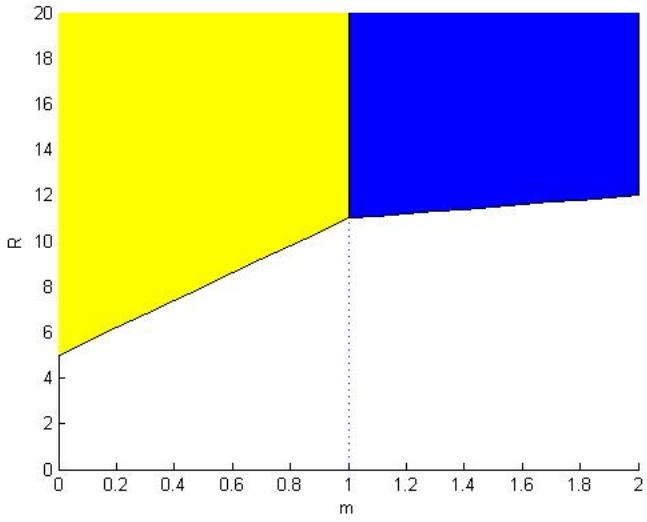


Figure 3.4: Segmentation of customers into R_1 and R_2

$$\begin{aligned} D_2(p_2, t_2) &= \frac{M}{\bar{R}\bar{m}}\big(\bar{R} - p_2 + \bar{R} - p_2 - m^* t_2\big)m^*/2 \\ &= \frac{Mm^*}{2\bar{R}\bar{m}}\big(2\bar{R} - 2p_2 - m^* t_2\big). \end{aligned} \tag{3.7}$$

Without loss of generality, the demands were derived under the assumption that $t_1 < t_2$, $p_1 > p_2$. Also, in this case, $m^* < \bar{m}$ and $p_1 + m^* t_1 < \bar{R}$. The maximization problem for the provider is to determine the price and release time of the two different releases, which is as follows:

$$\max_{p_1, t_1, p_2, t_2} \pi(p_1, t_1, p_2, t_2) = p_1 D_1(p_1, t_1) + p_2 D_2(p_2, t_2) - c(t_1) - c(t_2), \tag{3.8}$$

where we assume that cost of time is the same for both releases.

3.2.3 Exponential Decay

Starting with the same setting as in section 3.2.1, we define a utility function that is not linear. $u(p, t) = f(t) - p$, where $f(t) = Re^{-mt}$ with definitions for m

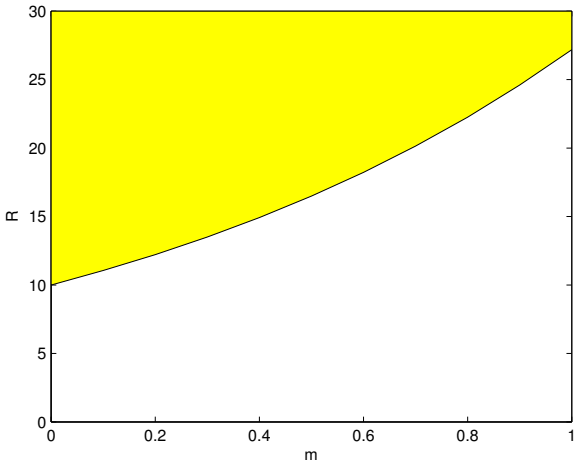


Figure 3.5: $S(10, 1)$, or "Buy Region"

and t unchanged. Now, the set S becomes $S(p, t) = \{(R, m)|R \geq pe^{mt}\}$. Figure 3.5 illustrates a "Buy Region," when $p = 10$, and $t = 1$, whose shape is different from Figures 3.2a and 3.2b.

As before, let $F(R, m)$ denote the joint cdf of the two parameters, and suppose customers' reservation price and impatience are independent and are distributed uniformly from 0 to $\bar{R}$ and 0 to $\bar{m}$. Then we can rewrite equation (3.1) for demand

as

$$\begin{aligned} D(p,t) &= M\int dF(R,m) \\ &= M\int_0^{\bar{m}}\int_{pe^{mt}}^{\bar{R}} \frac{1}{\bar{R}\bar{m}} dRdm \\ &= \frac{M}{\bar{R}\bar{m}}\int_0^{\bar{m}} (\bar{R}-pe^{mt})dm \\ &= \frac{M}{\bar{R}\bar{m}}\big(\bar{R}\bar{m}-\frac{p}{t}(e^{\bar{m}t}-1)\big) \\ &= M-\frac{Mp}{\bar{R}\bar{m}t}(e^{\bar{m}t}-1), \end{aligned} \tag{3.9}$$

and the maximization problem from equation (3.2) becomes

$$\begin{aligned} \max_{p,t}\Pi(p,t) &= pD(p,t)-c(t) \\ &= pM-\frac{Mp^2}{\bar{R}\bar{m}t}(e^{\bar{m}t}-1)-(F+\frac{k}{t}). \end{aligned} \tag{3.10}$$

We formulate a two-stage problem by separating price and time in the profit function. Let $V_p(t) = \max_p[pD(p,t)]$ represent the maximum revenue for a given release time t. Then equation (3.2) can be written as

$$\max_{p,t}\Pi(p,t) = \max_t[V_p(t)-c(t)]. \tag{3.11}$$

For any given t, we determine the revenue maximizing price $p^*(t)$, which is characterized in the following proposition.

Proposition 7. *For any t, the revenue $pD(p,t)$ is concave in p. The revenue maximizing price is $p^*(t) = \frac{\bar{R}\bar{m}t}{2(e^{\bar{m}t}-1)}$, which is monotonically decreasing in t.*

Proof. $\frac{\partial}{\partial p}\big(pD(p,t)\big) = M-\frac{2Mp}{\bar{R}\bar{m}t}(e^{\bar{m}t}-1)$, and $\frac{\partial^2}{\partial p^2}\big(pD(p,t)\big) = -\frac{2M}{\bar{R}\bar{m}t}(e^{\bar{m}t}-1) \leq 0$, which shows that the revenue function is concave in p. Solving the first order condition gives $p^*(t)$, and it is straightforward to show that the revenue maximizing price is monotonically decreasing in t, as $\frac{\partial}{\partial t}\big(p^*(t)\big) = -\frac{\bar{R}\bar{m}\left(e^{\bar{m}t}(\bar{m}t-1)+1\right)}{2(e^{\bar{m}t}-1)^2} \leq 0$. □

It is intuitive that the revenue maximizing price as a function of time decreases as release time increases. By obtaining $V_p(t)$, the problem from equation (3.11) then becomes

$$\max_t [V_p(t) - c(t)] = \max_t \left[\frac{M\bar{R}\bar{m}t}{4(e^{\bar{m}t} - 1)} - F - \frac{k}{t} \right]. \tag{3.12}$$

The shape of the profit function varies depending on the values of the constants, but it is possible to find profit-maximizing t under certain conditions as explained in the following proposition.

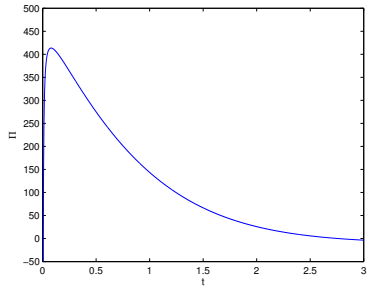
Proposition 8. *There exists a $t^* > 0$ that generates maximum positive profit $\Pi(t^*)$ under the following conditions:*

- $F < \frac{M\bar{R}}{4}$ is a necessary condition for having nonnegative profit.
- A sufficient condition for the existence of t^* is that for some $t > 0$, $\frac{M\bar{R}\bar{m}t}{4(e^{\bar{m}t}-1)} > F + \frac{k}{t}$.

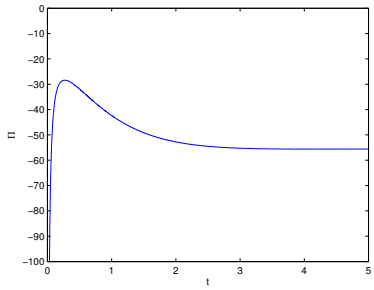
Proof. By definition, the domain of $\Pi(t) = \frac{M\bar{R}\bar{m}t}{4(e^{\bar{m}t}-1)} - F - \frac{k}{t}$ is $t > 0$, and it is continuous in the domain. We first characterize the behavior of the function by looking at its limits. We see that $\lim_{t\to 0^+} \Pi(t) = -\infty$, and $\lim_{t\to\infty} \Pi(t) = -F$. Since $\Pi(t)$ is continuous and bounded from above, $\Pi(t) > 0$ for some t implies that $\Pi(t) = 0$ for at least two distinct values of t in the domain, and therefore there must exist t^* with $\Pi'(t^*) = 0$, and t^* yields the maximum profit greater than zero.

Also, $\frac{M\bar{R}\bar{m}t}{4(e^{\bar{m}t}-1)}$ is monotonically decreasing in t, with $\lim_{t\to 0} \frac{M\bar{R}\bar{m}t}{4(e^{\bar{m}t}-1)} = \frac{M\bar{R}}{4}$. If $F \geq \frac{M\bar{R}}{4}$, $F + \frac{k}{t} > \frac{M\bar{R}}{4} \geq \frac{M\bar{R}\bar{m}t}{4(e^{\bar{m}t}-1)}$, which gives $\Pi(t) < 0$, proving the necessary condition. □

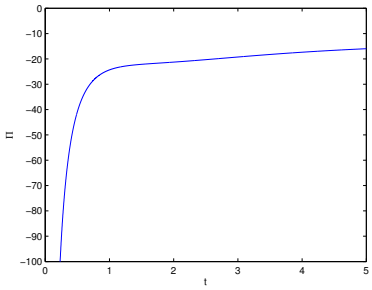
As expected, a sufficiently high fixed cost or service-rate cost will prevent positive profit. In Figure 3.6, profit functions under different parameters are illustrated with $\bar{R} = 20$ and $\bar{m} = 2$ fixed. We are only interested in situations represented in Figure 3.6a, as it is possible to have no positive profits as in Figures 3.6b or 3.6c. To find t^*, we can numerically solve the first order condition $\Pi'(t) = 0$.



(a) $M = 100$, $F = 10$, $k = 3$: $t^* = 0.08$, $\Pi(t^*) = 413.6$.



(b) $M = 10$, $F = 55$, $k = 3$: $t^* = 0.27$, $\Pi(t^*) = -28.4$.



(c) $M = 10$, $F = 10$, $k = 30$: no $t^* < \infty$.

Figure 3.6: Profit functions under different parameters

3.3 Supplier with Two Distribution Channels

We now consider a monopolist supplier who sells proprietary content to two separate distributors A and B, who then distribute it to customers. Let p_A, t_A, and p_B, t_B denote the parameters of distributor A and distributor B. The demand for channel A now depends on the aforementioned four parameters, the price and release time of channels A and B. We use the same utility function for customers, with $u(p,t) = -mt+R-p$, where m and R are impatience parameter and reservation price of each customer as defined in section 3.2, both uniformly distributed. Customers will only buy if it generates nonnegative utility, or when $R \geq p + mt$. Customers will buy from channel A if and only if $u(p_A, t_A) \geq u(p_B, t_B)$, or $m \geq \frac{p_A - p_B}{t_B - t_A}$. Figure 3.7 illustrates how customers are segmented by their value of m. In this generic example, $p_A = 5$, $p_B = 3$, $t_A = 1$, and $t_B = 3$. Customers whose (R, m) values fall in the yellow region will purchase from channel B, and those with values in the blue region will purchase from channel A. Figure 3.7 is analytical to Figure 3.4, except that demand is now split between two different distributors.

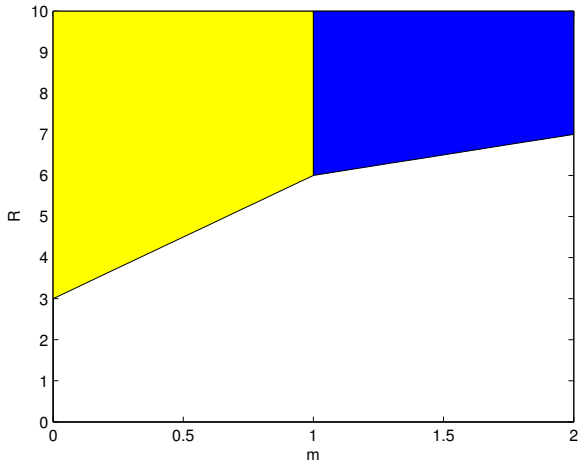


Figure 3.7: Segments of customers buying from providers A and B

3.3.1 Channel Competition

We start with a model where the supplier charges fixed price p_A^s and p_B^s to both distributors, and decides the release times t_A and t_B for them. In competition between two channels, the distributors only need to decide on prices to charge the customers. From equations (3.6) and (3.7), we have the demands for channels A and B from the end customers given p_A,p_B and t_A, t_B. Let v_A and v_B denote the variable cost of providing the content for two distributors. The profit for distributors A and B is calculated as:

$$\Pi_i(p_i) = (p_i - v_i)D_i - p_i^s, \tag{3.13}$$

for $i = A, B$, where D_A and D_B are from equations (3.6) and (3.7). Given the supplier's prices and release times, the pricing problem for distributor i is then $\max_{p_i}[(p_i - v_i)D_i - p_i^s]$, and two channels play a simultaneous game. To solve the problem, we examine each distributor's response price as a function of the other's price. Without loss of generality, we assume that $t_A < t_B$. We first establish the following Lemmas.

Lemma 9. *For channel B, the optimal response price is 0 when p_A is set to* 0.

Since $p_B > 0$ will result in zero demand based on our assumption on t_A and t_B, the result is straightforward.

Lemma 10. $\Pi_A(p_A)$ *is concave and obtains its maximum value for p_B below a threshold. For p_B greater than the threshold, it is optimal for distributor A to price just below p_B and capture the entire market.*

Proof. First order condition for provider A is $\frac{\partial \Pi_A}{\partial p_A} = D_A + (p_A - v_A)\frac{\partial D_A}{\partial p_A} = \frac{-2(v_A - p_A)}{t_B - t_A}(\frac{t_A(p_A - p_B)}{t_B - t_A} - R + p_A) - (\bar{m} - \frac{p_A - p_B}{t_B - t_A})(\frac{t_A(p_A - p_B)}{t_B - t_A} + \bar{m}t_A - 2\bar{R} - 2v_A + 4p_A)$, and second order condition gives $\frac{\partial^2 \Pi_A}{\partial p_A^2} = 2\frac{\partial D_A}{\partial p_A} + (p_A - v_A)D_A'' \leq 0$, which indicates concavity. $\Pi_A(p_A)$ is 0 for $p_A < p_B$ due to the assumption on t_A and t_B, so we only examine $p_A \geq p_B$. In order to have an optimal $p_A > p_B$, we need $\left.\frac{\partial \Pi_A}{\partial p_A}\right|_{p_A = p_B} > 0$. For $p_B > \frac{1}{2}(-\sqrt{2t_A^2\bar{m}^2 - 6t_A t_B \bar{m}^2 + 4t_B^2\bar{m}^2 + \bar{R}^2 - 2\bar{R}v_A + v_A^2} - 2t_A\bar{m} + 2t_B\bar{m} + \bar{R} + v_A)$, we have $\frac{\partial \Pi_A}{\partial p_A} < 0$ for all $p_A \geq p_B$, which proves that for p_B below the threshold, there exists $p_A^* > p_B$ maximizing $\Pi_A(p_A)$. For p_B greater than the threshold, $\Pi_A(p_A)$ is concave decreasing in p_A. □

Lemma 11. $p_A^* > 0$ *when* $p_B = 0$.

Proof. From Lemma 10, we can use first order condition to obtain p_A^*, which then proves the result. □

From the first order condition, we have $p_A^*(p_B) = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$, where $A = 6 + \frac{3t_A}{t_B - t_A}$, $B = \frac{1}{t_B - t_A}(t_A^2\bar{m} - t_A t_B \bar{m} - 4t_A p_B - 2t_A v_A) + 5t_A\bar{m} - 4(t_B\bar{m} + p_B + \bar{R} + v_A)$, and $C = \frac{1}{t_B - t_A}(t_A p_B^2 + t_A t_B \bar{m} p_B - t_A^2 \bar{m} p_B + 2t_A v_A p_B) + t_A\bar{m}(t_A\bar{m} - t_B\bar{m} - p_B - 2\bar{R} - 2v_A) + t_B\bar{m}(2\bar{R} + 2v_A) + 2\bar{R}p + 2v_A p_B + 2v_A\bar{R}$. We derive the slope of the response function which has the following property.

Lemma 12. *The derivative of the response function for A with respect to P_B is not greater than 1.*

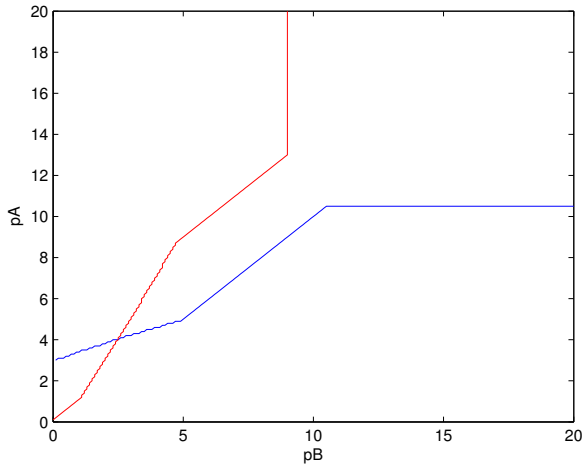


Figure 3.8: Response functions for A and B

Proof. Taking derivative of the response function, we have $\frac{\partial p_A^*(p_B)}{p_B} = \frac{t_A(p_B+2p_A-\bar{R}-2v_A)+t_B(\bar{R}+v_A-2p_A)}{2t_A(p_A+p_B-\bar{R}-v_A)+t_B(2\bar{R}+v_A-3p_A)}$. It can be shown that $\frac{\partial p_A^*(p_B)}{p_B} \leq 1 \iff t_B p_A - t_A p_B \leq \bar{R}(t_B - t_A)$, and since by assumption $\bar{R} \geq p_A \geq p_B$, this completes the proof. □

Thus, we can assume that given sufficiently large $\bar{R}$, $p_A^*(p_B)$ will start at a positive value, increases as p_B increases at a slower rate than p_B until p_B reaches the threshold, and then will have a slope of 45 degrees, since $p_A^*(p_B) = p_B - \epsilon$ for p_B greater than the threshold, as illustrated by the blue line in Figure 3.8.

Now we can similarly analyze response function $p_B^*(p_A)$ in order to characterize equilibrium prices.

Lemma 13. *For $2t_A \leq t_B$ and $p_B \leq T$, and for $2t_A \geq t_B$ and $p_B \geq T$, $\Pi_B(p_B)$ is concave, and it achieves a maximum.*

Proof. First order condition for provider B is $\frac{\partial \Pi_B}{\partial p_B} = D_B + (p_B - v_B)\frac{\partial D_B}{\partial p_B} = ((v_B - p_B)(p_A - p_B)t_B + (p_A - p_B)(2p_B - v_B - p_A)t_B)/(t_B - t_A)^2 + ((2R - 2p_B)(v_B - p_B) + (p_A - p_B)(2(v_B - p_B) + 2R - 2p_B))/(t_B - t_A)$, and second order condition gives

$\frac{\partial^2 \Pi_B}{\partial p_B^2} = (4R(t_A - t_B) + 4t_A(p_A + v_B) + 6t_B p_B - 12t_A p_B - 2t_B v_B$. Solving for $\frac{\partial^2 \Pi_B}{\partial p_B^2} \leq 0$, we obtain concavity for the two cases: $2t_A \leq t_B$ and $p_B \leq T$, and $2t_A \geq t_B$ and $p_B \geq T$, where the threshold $T = \frac{2t_A(p_A+R+v_B)-t_B(2R+v_B)}{6t_A-3t_B}$. □

Proposition 14. *There exists an equilibrium (p_A^*, p_B^*) pair.*

Proof. Unlike $\Pi_A(p_A)$, $\Pi_B(p_B)$ always achieves a maximum. From our original assumption that $t_A < t_B$, we have $p_B^* < p_A^*$. Combining this with Lemmas 9, 11, and 12, the response functions $p_A^*(p_B)$ and $p_B^*(p_A)$ cross, thus completing the proof. □

Although closed-form solutions for p_A^* and p_B^* are not available, we have shown that when the supplier presets the release times for the distributors, there exist equilibrium prices for distributors A and B.

3.3.2 Pricing Problem for the Supplier

Now we look at the supplier's problem of deciding on the prices for the content sold to distributors A and B. First assume that the release times t_A and t_B are set in advance due to technological limits and regulations. The supplier can either charge a one-time fixed price or a variable price proportional to each channel's revenue. The following proposition illustrates the case of fixed prices.

Proposition 15. *When the supplier charges a fixed price to the distributors, the optimal price is $p_i^{s*} = (p_i^* - v_i)D_i^*$ for $i = A, B$.*

Proof. Supplier's problem is $\max[p_A^s + p_B^s]$. From equation (3.13), distributor's problem was $\max_{p_i}[(p_i - v_i)D_i - p_i^s]$, and distributors will not purchase from the supplier if their expected profit is negative. □

If distributors pay the supplier a variable price based on the volume of sales, assuming fixed release times, the supplier's problem is

$$\max_{p_A^s, p_B^s}[p_A^s D_A + p_B^s D_B] \quad = \quad \max_{p_A^s, p_B^s}[p_A^s D_A(p_A^*) + p_B^s D_B(p_B^*)]. \tag{3.14}$$

In the case of variable price charged by the supplier proportional to the actual demand, we start by examining the case of equal price for both distributors, i.e., $p_A^s = p_B^s$. Let p^s denote this price. The supplier is solving a profit maximization problem in which every parameter involved depends on p^s, the price it sets for the distributors. Note that equation (3.13), the profit for distributor A and B, will now be $\Pi_i(p_i) = (p_i - v_i - p_i^s)D_i$, for $i = A, B$ since distributors do not pay a fixed fee to the supplier in this case. The following lemmas lead to the analysis of the solution method.

Lemma 16. *The derivatives of the response functions p_A^* and p_B^* with respect to p^s are nonnegative.*

Proof. p_A^* and p_B^* are optimal solutions to each distributor's profit maximization problem. If the optimal price for a distributor decreases when p^s goes up, the distributor's profit becomes negative for $p_i^* < p_i^s$, which is not optimal. □

This says that the prices set by the distributors increase as the supplier charges higher price, which is intuitive.

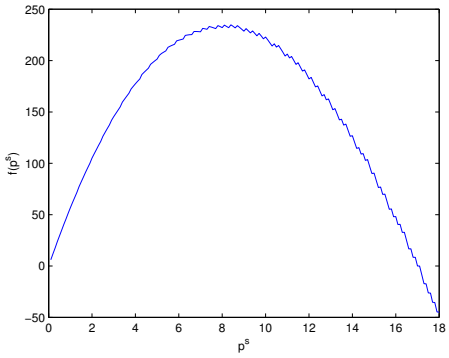
Lemma 17. *For $\bar{m}$ sufficiently large, the sum of the demands $D_A(p_A^*)$ and $D_B(p_B^*)$ decreases as p^s increases.*

Proof. We have equations for demands from equations (3.6) and (3.7). Let $p_A^{*'}$denote $\frac{\partial p_A^*(p^s)}{\partial p^s}$ for notational purpose. Taking derivative with respect to p^s, we have

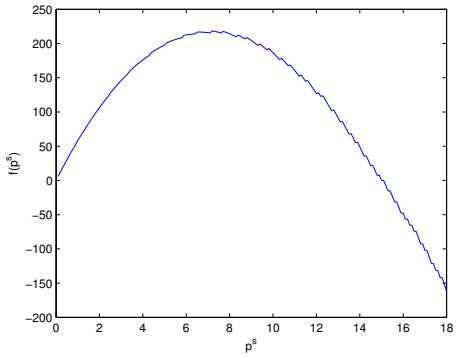
$$\frac{\partial D_A}{\partial p^s} = \frac{M}{2\bar{R}\bar{m}}\left[\frac{p_A^{*'} - p_B^{*'}}{t_B - t_A}(2\bar{R} - 2p_A^* - 2\bar{m}t_A) - 2\bar{m}p_A^{*'} + \frac{2p_A^{*'}(p_A^* - p_B^*)}{t_B - t_A}\right] \quad (3.15)$$

$$\frac{\partial D_B}{\partial p^s} = \frac{M}{2\bar{R}\bar{m}}\left[\frac{p_A^{*'} - p_B^{*'}}{t_B - t_A}(2\bar{R} - 2p_A^* - \bar{m}t_A) - \frac{2p_A^{*'}(p_A^* - p_B^*)}{t_B - t_A}\right]. \quad (3.16)$$

From the sum of the two derivatives, we get $\frac{\partial D_B}{\partial p^d} + \frac{\partial D_A}{\partial p^d} < 0$ if and only if $\bar{m} > \frac{p_A^{*'} - p_B^{*'}(4\bar{R} - 4p_A^*)}{2t_B p_A^{*'} + t_A(p_A^{*'} - 3p_B^{*'})}$. □



(a) Release times set to $t_A = 3$, $t_B = 10$



(b) $t_A = 5$, $t_B = 8$

Figure 3.9: Graph of supplier's profit on p^s

In order to solve the supplier's problem, $\max_{p^s}[p^s(D_A(p_A^*) + D_B(p_B^*))]$, we first characterize the first order and second order conditions of the objective function. Let $f(p^s) = p^s(D_A(p_A^*) + D_B(p_B^*))$. We have $f^{'}(p^s) = p^s(D_A^{'} + D_B^{'}) + D_A + D_B$, with $D_i^{'} = \frac{\partial D_i}{\partial p^s}$. Also, $f^{''}(p^s) = p^s(D_A^{''} + D_B^{''}) + 2(D_A^{'} + D_B^{'})$. And the following proposition partially characterizes the objective function.

Proposition 18. *For $\bar{m}$ sufficiently large, the objective function $f(p^s)$ is concave in p^s.*

Proof. From equations (3.15) and (3.16), we have $\frac{\partial^2 D_A}{\partial (p^s)^2} + \frac{\partial^2 D_b}{\partial (p^s)^2} = \frac{M}{2\bar{R}\bar{m}}\left[\frac{p_A^{*''} - p_B^{*''}}{t_B - t_A}(2\bar{R} - 2p_A^* - 3mt_A) - \frac{4(p_A^{*'} - p_B^{*'})p_A^{*'}}{t_B - t_A} - 2\bar{m}p_A^{*''}\right]$. Similar to the proof for lemma 17, we have $D_A^{''} + D_B^{''} < 0$ for $\bar{m} > \frac{p_A^{*''} - p_B^{*''}(4\bar{R} - 4p_A^*)}{2t_B p_A^{*''} + t_A(p_A^{*''} - 3p_B^{*''})}$. □

For both equal or different supplier prices, we are able to find the optimal prices numerically, with different t_A and t_B values. Figure 3.9 illustrates proposition 18.

3.4 Conclusion and Future Work

In this chapter we modeled and studied a decision problem faced by a supplier of information content. In today's fast-paced digital market, the speed of release of a new content is as important as its price. A lot of people are willing to pay premium to receive a new release sooner, although the willingness to pay varies across individuals. With the assumption that customers are sensitive to both price and release time of content, we studied the profit maximization problem for a monopolist provider under different assumptions on customer utility functions, and showed that the optimal price for the monopolist provider exists when underlying utility function is linear.

Then we considered the case of a supplier with two downstream distribution channels where the channels have to compete on setting their price and release time for the identical content to attract more customers. An example would be a movie provider as a single supplier, with different distributors as their downstream channels competing on price and release time. When supplier fixes the release time for the content and charges a one-time fixed price to the channels, we find that there is an equilibrium pair of prices for the channels to charge the customers. In the case of the supplier charging variable prices proportional to the customer demand for each distributor, we show that the objective function for the supplier is concave in the case of equal pricing structure.

Further research could extend the work to study the characteristics of optimal price and release times when there is more than one supplier in the market. It would be interesting to analyze the differences in supplier behavior for other customer utility models. Another extension would be to examine the channel-level problem of deciding on their own release times instead of having it fixed by the supplier.

We have so far defined customers with two parameters: impatience and reservation price, and assumed that they will always purchase content that gives them positive utility. As in August et al. (2013), we could add quality parameters to

the content to expand the customer utility model for choosing whether to buy content at certain time, which, though significantly more complex, could model the real behavior of buyers more closely.

Bibliography

Ancker, C. J., A. V. Gafarian. 1963a. Some queuing problems with balking and reneging–i. *Operations Research* **11**(1) 88–100.

Ancker, C. J., A. V. Gafarian. 1963b. Queuing with reneging and multiple heterogeneous servers. *Naval Research Logistics Quarterly* **10**(1) 125–149.

Apte, U., U. S. Karmarkar, C. Kieliszewski, Y. T. Leung. 2012. Exploring the representation of complex processes in information-intensive services. *International Journal of Services Operations and Informatics* **7**(1) 52–78.

Armbruster, Dieter, R. Uzsoy. 2012. Continuous dynamic models, clearing functions, and discrete-event simulation in aggregate production planning. P. Mirchandani, ed., *INFORMS TutORials in Operations Research*, vol. 9. Hanover, MD, 103–126.

Asmundsson, Jakob, Ronald L Rardin, Can Hulusi Turkseven, Reha Uzsoy. 2009. Production planning with resources subject to congestion. *Naval Research Logistics (NRL)* **56**(2) 142–157.

August, Terrence, Duy Dao, Hyoduk Shin. 2013. Optimal timing of sequential distribution: the impact of congestion externalities and day-and-date strategies. *Available at SSRN 1708226* .

Barrer, D. Y. 1957. Queuing with impatient customers and ordered service. *Operations Research* **5**(5) 650–656.

Bashyam, T. C. A. 2000. Service design and price competition in business information services. *Operations Research* **48**(3) 362–375.

Bashyam, T. C. A., U. S. Karmarkar. 2000. Aspect development. J. De La Torre, Y. Doz, T. Devinney, eds., *Managing the Global Corporation: Case Studies in Strategy and Management*. McGraw Hill.

Bashyam, T. C. A., U. S. Karmarkar. 2007. Service design, competition and market segmentation in business information services with data updates. Uday Apte, Uday Karmarkar, eds., *Managing in the Information Economy: Current Research Issues*, chap. 13. Springer Science.

Bennett, James, Stan Lanning. 2007. The netflix prize.

Bitran, Gabriel R, Devanath Tirupati. 1993. Hierarchical production planning. *Handbooks in operations research and management science* **4** 523–568.

Borst, Sem, Avi Mandelbaum, Martin I. Reiman. 2004. Dimensioning large call centers. *Operations Research* **52**(1) 17–34.

Boyd, S. P., L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Bright, L., P.G. Taylor. 1995. Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes. *Communications in Statistics. Stochastic Models* **11**(3) 497–525.

Cocchi, Ron, Scott Shenker, Deborah Estrin, Lixia Zhang. 1993. Pricing in computer networks: Motivation, formulation, and example. *IEEE/ACM Transactions on Networking (TON)* **1**(6) 614–627.

Connor, M. 1998. Weather services corporation. Tech. Rep. 9-396-052, Harvard Business School, Boston, MA.

Crabill, Thomas B. 1972. Optimal control of a service facility with variable exponential service times and constant arrival rate. *Management Science* **18**(9) 560–566.

De Vleeschauwer, Danny, Koenraad Laevens. 2009. Performance of caching algorithms for iptv on-demand services. *Broadcasting, IEEE Transactions on* **55**(2) 491–501.

Dobson, Gregory, Uday S. Karmarkar. 2011. Production planning under uncertainty with workload-dependent lead times: Lagrangean bounds and heuristics. Karl G Kempf, Pinar Keskinocak, Reha Uzsoy, eds., *Planning Production and Inventories in the Extended Enterprise*, *International Series in Operations Research & Management Science*, vol. 152. Springer New York, 1–14. doi:10.1007/978-1-4419-8191-2_1. URL `http://dx.doi.org/10.1007/978-1-4419-8191-2_1`.

Dyer, M. E., L. G. Proll. 1977. On the validity of marginal analysis for allocating servers in M/M/c queues. *Management Science* **23**(9) 1019–1022.

Feldman, Zohar, Avishai Mandelbaum, William A Massey, Ward Whitt. 2008. Staffing of time-varying queues to achieve time-stable performance. *MANAGEMENT SCIENCE* **54**(2) 324–338.

Garnett, O., A. Mandelbaum, M. Reiman. 2002. Designing a call center with impatient customers. *Manufacturing & Service Operations Management* **4**(3) 208–227.

George, Jennifer M., J. Michael Harrison. 2001. Dynamic control of a queue with adjustable service rate. *Operations Research* **49**(5) 720–731.

Haight, Frank. 1959. Queueing with reneging. *Metrika* **2**(1) 186–197.

Harrison, J. Michael. 1997. Manzana insurance: Fruitvale branch. Tech. Rep. 9-692-015, Harvard Business School, Boston, MA.

Hax, Arnoldo C. 1978. Aggregate production planning. *Handbook of operations research* **2** 127–172.

Ipeirotis, Panagiotis G, Alexandros Ntoulas, Junghoo Cho, Luis Gravano. 2005. Modeling and managing content changes in text databases. *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*. IEEE, 606–617.

Jain, Sanjay, PK Kannan. 2002. Pricing of information products on online servers: Issues, models, and analysis. *Management Science* **48**(9) 1123–1142.

Karmarkar, U. S., U. M. Apte. 2007. Operations management in the information economy: Information products, processes, and chains. *Journal of Operations Management* **25**(2) 438–453.

Karmarkar, Uday S. 1989. Capacity loading and release planning with work-in-progress (wip) and lead-times. *Journal of Manufacturing and Operations Management* **2** 105–123.

Karmarkar, U.S. 2014. Veralex services and the auto-cite product. Tech. rep., UCLA Anderson School, Los Angeles, CA.

Kharoufeh, Jeffrey P. 2011. *Level-Dependent Quasi-Birth-and-Death Processes*. John Wiley & Sons, Inc.

Köllerström, Julian. 1974. Heavy traffic theory for queues with several servers. i. *Journal of Applied Probability* 544–552.

Mendelson, Haim, Seungjin Whang. 1990. Optimal incentive-compatible priority pricing for the m/m/1 queue. *Operations research* **38**(5) 870–883.

Moorthy, K Sridhar. 1988. Product and price competition in a duopoly. *Marketing Science* **7**(2) 141–168.

Osogami, Takayuki. 2005. Analysis of Multi-server Systems via Dimensionality Reduction of Markov Chains. Ph.D. thesis, Carnegie Mellon University.

Papadimitriou, Athanassios. 2004. Efficient data inventory management. Ph.D. thesis, University of California at Los Angeles.

Selcuk, Baris, JC Fransoo, AG De Kok. 2008. Work-in-process clearing in supply chain operations planning. *IIE Transactions* **40**(3) 206–220.

Thompson, Gary M. 1997. Labor staffing and scheduling models for controlling service levels. *Naval Research Logistics (NRL)* **44**(8) 719–740.

Whitt, Ward. 2007. What you should know about queueing models to set staffing requirements in service systems. *Naval Research Logistics (NRL)* **54**(5) 476–484. doi:10.1002/nav.20243. URL `http://dx.doi.org/10.1002/nav.20243`.