Optimal Advertisement Scheduling
in Breaks of Random Lengths

by
Ajay S. Aravamudhan

Submitted to Lee Kong Chian School of Business in partial fulfillment
of the requirements for the Degree of
Master of Science in Operations Management

Thesis Committee
Pascale Crama (Supervisor / Chair)
Assistant Professor of Operations Management
Singapore Management University

Onur Boyabatli
Assistant Professor of Operations Management
Singapore Management University

Sharafali Moosa
Associate Professor of Operations Management (Education)
Singapore Management University

Singapore Management University
2011

©(2011) Ajay S. Aravamudhan
Abstract

Optimal Advertisement Scheduling in Breaks of Random Lengths

by Ajay S. Aravamudhan

Broadcasters generate a large part of their revenue through advertising, especially in live sports. Scheduling advertisements can be challenging in live broadcasting, however, for sports such as Cricket that have breaks of random lengths and number during which the ads are shown. This uncertainty, coupled with the high price of spots for major competitions, means that improving ad scheduling can add significant value to the broadcaster. This problem shares similarities with the stochastic cutting stock problem and the dynamic stochastic knapsack problem, with applications in the wood, steel and paper industry and the transportation industry respectively.

This dissertation adds to the existing literature on advertising scheduling by taking stochasticity in break sizes into consideration. We propose an optimal scheduling rule under simplifying assumptions and prove that our policy outperforms traditional scheduling methods. We also study the performance of several heuristics, and find that a flexible heuristic that does not depend on creating bundles performs the best.
Contents

1 Introduction 1

2 Literature Review 6
  2.1 Revenue Management with Random Capacity . . . . . . . . . . 6
  2.2 Dynamic Stochastic Knapsack Problem . . . . . . . . . . . . . 8
  2.3 Stochastic Cutting Stock Problem . . . . . . . . . . . . . . . 9
  2.4 Revenue Management in Media Applications . . . . . . . . . . 12

3 Optimal Policy 14
  3.1 Assumptions . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
  3.2 Known Break Size . . . . . . . . . . . . . . . . . . . . . . . 16
    3.2.1 Base Case . . . . . . . . . . . . . . . . . . . . . . . . 17
    3.2.2 Stochastic Number of Breaks . . . . . . . . . . . . . . 20
    3.2.3 Multiple Break Sizes . . . . . . . . . . . . . . . . . . 24
    3.2.4 Stochastic Number of Breaks of Multiple Sizes . . . . . 30
  3.3 Unknown Break Size . . . . . . . . . . . . . . . . . . . . . . 34
    3.3.1 Base Case . . . . . . . . . . . . . . . . . . . . . . . . 34
    3.3.2 Stochastic Number of Breaks . . . . . . . . . . . . . . 39
    3.3.3 Multiple Break Sizes and its variants . . . . . . . . . . 42

4 Numerical Analysis 45
  4.1 Performance with deterministic number of breaks . . . . . . . 45
    4.1.1 Parameters . . . . . . . . . . . . . . . . . . . . . . . 46
### CONTENTS

4.1.2 Impact of variation in air time sold .......................... 46  
4.1.3 Value of flexibility ............................................. 49  
4.1.4 Impact of Variability ............................................. 50

5 Applications in Practice: Scheduling ads for Cricket ............ 56  
5.1 Data Description ..................................................... 56  
5.1.1 Break Lengths ...................................................... 56  
5.1.2 Ad Lengths, Service Levels and Demands .................... 58  
5.1.3 Spot Values ......................................................... 59  
5.2 Assumptions .......................................................... 59  
5.3 Knapsack Model ....................................................... 60  
5.3.1 Knapsack model with service guarantee ..................... 61  
5.4 Heuristics ............................................................. 62  
5.4.1 Greedy Policy ....................................................... 62  
5.4.2 Certainty Equivalent Heuristic ................................ 63  
5.4.3 Dynamic Certainty Equivalent Heuristic ..................... 63  
5.4.4 Dynamic Modified Certainty Equivalent Heuristic ........ 63  
5.4.5 Perfect Information ............................................... 63  
5.5 Comparative Statics with Service Constraints .................. 64  
5.5.1 Parameters ......................................................... 64  
5.5.2 Results ............................................................. 65  
5.6 Comparison of Greedy and Optimal Policies .................... 66  
5.6.1 Results ............................................................. 67

6 Conclusion .............................................................. 70

Bibliography .............................................................. 72

A Proofs of theorems ..................................................... 76  
A.1 Proof of Theorem 3.1 ............................................... 76  
A.2 Proof of Theorem 3.2 ............................................... 78  
A.3 Proof of Theorem 3.3 ............................................... 81
A.4 Proof of Theorem 3.4 ........................................ 83
A.5 Proof of Theorem 3.5 ........................................ 87
A.6 Proof of Theorem 3.6 ........................................ 89

B Tables ........................................ 91
B.1 Revenues ........................................ 92
B.2 Service Levels ........................................ 94
B.3 Utilization of breaks .................................... 98
Acknowledgements

It is a pleasure to thank the many people who made this thesis possible.

Firstly, I would like to thank Dr. Pascale Crama, my thesis advisor, for teaching me the ways of research and the art of writing well. She has been friendly, approachable, and always right. Her gentle encouragement and words of advice have been invaluable in shaping my development as a research student.

I would also like to thank Dr. Dana Popescu, for clarifying the assumptions and the model we would work with. Her enthusiasm and insight made my work easier.

I would like to express my gratitude to the unnamed sports broadcaster who first introduced the problem to us, and the managers therein. Without them, this thesis would not have been conceivable.

I wish to thank my lecturers during my Masters, who provided me with the foundations I required to work on this dissertation. In particular, I would like to thank Dr. Onur Boyabatli and Dr. Sharafali Moosa, who gave me valuable advice during my thesis proposal.

I am indebted to Dr. Reddi Kotha for encouraging me to apply to SMU. I wish I could have lived up better to the potential he saw in me.

Most importantly, I would like to thank my wife, Gina Han, firstly for never complaining while I struggled to provide her with the comforts she deserves, and secondly for being a rock and providing encouragement when I struggled with self-doubt. I look forward to the years in her company.
Chapter 1

Introduction

Revenue for television broadcasters is generated primarily through the sale of local, regional, and national advertising on the local stations and their networks. At CBS, the most-watched U.S. broadcast network, TV advertising accounted for two-thirds of its revenue\(^1\). Major sporting events, such as the Super Bowl, the Olympics or the Football World Cup, greatly increase advertising revenues as advertisers are willing to pay a premium to air their ads during live broadcast.

In India, cricket is the main revenue earner for sports broadcasting networks. According to a study from TAM Media Research’s advertising measurement arm AdEx, ad volumes in cricket saw a growth of more than three times in the five years from 2002 to 2007 with the volumes showing an extra spurt during the World Cups in 2003 and 2007\(^2\). The biggest spurt was seen in the 2007 World Cup where the volumes rose nearly 100%, with 22% of the advertising volumes in live cricket telecasts.

In recent times, the importance of advertising in cricket has increased even more, with the introduction of shorter formats of the game such as T-20, which is aimed at prime-time television viewers. Sony, the broadcaster for the Indian Premier League 2010, was expected to have earned approximately

---

\(^1\)Bloomberg Businessweek 2010  
\(^2\)www.indiantelevision.com/
Figure 1.1: Growth of investment in advertising in live cricket, 2002-2007

Figure 1.2: Slice of revenue earned through live cricket broadcasts, 2007
USD 150 millions from live broadcasting for the tournament alone, with advertising spots valued at more than USD 1100 per second\textsuperscript{3}. Thus, even a small percentage gain in advertising revenue can translate to a large increase in earnings.

Scheduling advertisements in live broadcasting can be challenging, however, especially for sports that have breaks that are non-deterministic. This is the case for cricket, in which the duration and also the number of breaks can be random. Breaks in cricket are taken between overs, when a wicket falls, or there is a break in the game for refreshments or due to an injury. While the length of the game is usually predictable (especially in the case of limited overs games such as ODIs or T-20s), it is not fixed. Teams have to take up to ten wickets within the allotted overs, however overs may run out before all wickets are taken. Teams may also find that all their wickets have fallen before they have managed to bat out the overs. Thus, the number of breaks, time when they occur, and lengths of breaks in a game cannot be foretold, and broadcasters have to make ad scheduling decisions while the break is ongoing.

To do this, networks usually employ several people with specialized knowledge of the sport when scheduling commercials. These ad schedulers have a view of the live game as it happens from a centralized control center, which also has a list of ads available on a mainframe, along with the orders (number of times each ad has to be shown). Before the match starts, the ad scheduling team creates a few sample bundles, giving priority to tournament sponsors. The bundles are initially scheduled as planned, but are broken and ads are scheduled on the fly because break sizes are stochastic. The scheduling team is advised by an on-field director, who can judge the state of the game and inform them about how long the break could be. Based on this advice, and the known break size distribution, the schedulers select ads to be aired in each break.

\textsuperscript{3}Wall Street Journal, Jan 2010
The job of the ad schedulers is stressful because of the need to develop good schedules while under constant pressure to satisfy scheduling constraints. For instance, if live action begins while an ad is running, then the ad has to be stopped midway (the ad is said to have ‘crashed’) in order to air live action, thus forgoing any revenue from the crashed ad. Furthermore, networks have to satisfy service levels promised to agencies that get them the ad contracts, and these service levels are based on sponsorship statuses and geographical location. Thus, by reducing dependency on human intervention, there is an opportunity to maximize revenues by automating the commercial scheduling process while generating near optimal schedules to meet all goals.

In this dissertation, we consider two approaches to study the problem. We begin with Chapter 2, where we review literature from the streams of advertising, random yield, stochastic knapsack and stochastic programming with recourse to help us gain insights for doing our study.

In Chapter 3, we approach the problem analytically, and study optimal scheduling policies when faced with stochastic break sizes for a simplified setting, in which we do not consider constraints other than crashing. We consider two cases, where the scheduling team either has prior information about the break duration for the ongoing break, or it does not. In each case, we assume that the inventory contains ads of two sizes, $S$ and $2S$, and consider scenarios where break sizes range from zero to any multiple of $S$, and where the number of breaks can be stochastic. In Chapter 4, we do a numerical analysis to study the sensitivity of the Optimal Policy to variations in problem parameters relative to the myopic Greedy rule, which we define as the ad schedule which generates the maximum revenue for the break in hand but does not consider the subsequent breaks.

Finally, in Chapter 5, we study several heuristics inspired by the current scheduling practice in order to improve a broadcaster’s revenues. We use data provided by a major cricket broadcaster to analyze the performance of
several heuristics which create bundles beforehand. We consider generating bundles at various points during a match, and compare performance with the standard Greedy heuristic. We include some constraints for this study, such as minimum service levels for each client, to analyze how creating bundles beforehand affects revenue earned, and how often they should be created.

Finally, we summarize our findings and present our conclusions with directions for future research in Chapter 6.
In this dissertation, we take two approaches to study the problem of optimally scheduling ads for random breaks: first we propose optimal policies for scheduling ads during breaks of random durations under simplifying assumptions, and secondly, we analyze several heuristics inspired by current scheduling practice hoping to provide managerial insight to sports broadcasters. This research has links to revenue management with random capacity, the dynamic stochastic knapsack problem, the stochastic cutting stock problem and revenue management in media applications.

2.1 Revenue Management with Random Capacity

Our problem is related to revenue management with random capacity / random yield, with typical applications in production planning. The most common choice to model random yield has been stochastically proportional yield, in which the yield is proportional to the order.

Ciarallo et al. (1994) [7] are the first to explore the impact of random capacity. The authors find that an order-up-to policy is optimal to minimize production costs. The order target includes a safety stock to account for
random capacity in future periods and is higher than the myopic order-up-to level. Wang and Gerchak (1996) [27] revisit the results found in Ciarallo et al. (1994) to offer a more rigorous proof of the order-up-to policy. Khang and Fujiwara (2000) [15] prove under which conditions the myopic order-up-to policy is optimal in a multi-period setting. Hwang and Singh (1998) [14] extend the analysis to a multi-stage production process and find an optimal policy characterized by a sequence of two critical numbers for each stage: a minimum input level below which no production takes place and a maximum desired production level. Finally, Wang and Gerchak (1996) [26] incorporate random yield and capacity and show that the optimal policy is characterized by a single reorder point in each period. That critical point is not a constant and depends on the inventory in hand.

Yano and Lee (1995) [29] review the literature in the area of lot sizing with random yields, focusing on single-stage continuous review models and single-stage periodic review models. They cover modeling of costs, modeling of yield uncertainty, and measures of performance of the system. Grosfeld-Nir and Gerchak (2004) [13] review papers discussing multiple lot sizing in production to order in multistage systems, and review situations where both yield and demand are random. Bollapragada and Morton (1999) [6] provide heuristics for dealing with random inventory by focusing on inventory at the end of the period, after the demand is met. They show that the random yield problem is analogous to the newsvendor problem with the demand distribution dependent on the quantity ordered. Their research supports the argument that myopic and near-myopic methods are useful across a wide spectrum of stochastic inventory problems.

Our model differs in two important aspects from the random yield and random capacity papers above. First, we maximize revenues rather than minimize costs. All the papers assume a single product whereas we work in a multi product setting with different prices and production costs. Therefore we need to schedule those products based on their profitability and their
capacity usage. Second, we assume integer units or fixed-size order runs. Therefore, we cannot simply use capacity to its maximum and hold inventory to complete an order across multiple periods. Each order needs to be entirely processed within one production period.

2.2 Dynamic Stochastic Knapsack Problem

The knapsack problem is one of the simplest and thus oldest formulation of a maximization problem. The knapsack problem has been extensively studied in operations research, and has various industrial applications in areas such as resource allocation, capital budgeting, portfolio selection problem, cargo loading, and cutting stock problems. Knapsack problems of this type are deterministic because all parameters are known with certainty. However, in many situations, these parameters may be random variables having a certain distribution. Kleywegt and Papastavrou have written a series of papers on this topic ((1996) [19], (2001) [20]), that define the Dynamic Stochastic Knapsack problem as one in which items to be packed arrive according to a known distribution, and determine the optimal policy that maximizes expected value, given the costs associated with waiting. They expand their research to cases where the rewards associated with an item are stochastic, and when the size of each item is also stochastic. Our research, however, attempts to solve a problem with multiple knapsacks, whose sizes are stochastic, and the items are of known size and value.

More recently, Perry and Hartman (2009) [22] model a multi-period, single resource capacity reservation problem as a dynamic, stochastic, multiple knapsack problem with stochastic dynamic programming. They propose an approximation approach which utilizes simulation and deterministic dynamic programming in order to allow for the solution of longer horizon problems and ensure good time zero decisions. Their simulation based approach, however, does not sufficiently capture the complexities of our problem.
Witchakul, Ayudhya, Charnsethikul (2008) [28] discuss random Knapsack capacity with deterministic weights and costs. They model the Knapsack’s capacity as a random variable with a known distribution. They use the expectation of Knapsack size, and both underage and overage penalty costs, to estimate optimum selection of ads. They provide a heuristic for solving Stochastic Knapsack with Continuous/Discrete Random Capacity, and prove the validity of their heuristic analytically and numerically using a Monte Carlo Simulation. To the best of our knowledge, this is one of the few papers that address stochastic knapsack sizes. The problem we face, however, can be seen as a modified bin packing problem where multiple bins of stochastic capacity have to be optimally filled. Besides, unlike the above paper, we do not consider overage and underage penalties.

2.3 Stochastic Cutting Stock Problem

The cutting stock problem originated as a knapsack problem which minimizes unused capacity rather than maximize revenue from the included items. It is based on industry applications which require to solve how to cut stock of a certain dimension into smaller, heterogeneous order sizes in such a way as to minimize waste of material, e.g. in the paper or steel industry. The problem was introduced by Gilmore and Gomory (1961) [10], and over a series of papers, the authors proposed a set of specialized techniques to solve the cutting stock problem (Gilmore and Gomory 1963 [11], 1965 [12]). One line of extensions to this problem looks at stock with stochastic dimensions. The randomness can be due to the nature of the stock, e.g., raw material like wood or stone slabs may come in unequal sizes, quality variation within the stock or defects at the edges of the stock. Scull (1981) [23] introduces a stochastic cutting stock problem in which the uncertainty in the stock length is due to defects at the edges. The stock is then cut into standard-length units, and the authors find the optimal distance from the edge at which
to start cutting in order to minimize expected waste if inspection of the stock and its defects is not possible. Ghodsi and Sassani (2005) [9] introduce quality and length variability of the stock and the orders. A cut pattern needs to be decided upon arrival of each piece of stock. The authors propose a dynamic algorithm which first prioritizes the orders based on their quality level and quantity and then proposes a suitable cut pattern for the incoming stock. Even though the orders have different quality requirements, their revenue is assumed constant, and the objective is to minimize waste. Fathi and Kianfar (2009) [16] acknowledge that variability in quality may also lead to difference in revenue, and formulate a similar problem with quality and length variability with the objective to maximize revenue. They formulate the cut pattern problem as a dynamic program and conduct a numerical experiment to show that it is feasible to solve this problem in real time. The authors, however, do not comment on the performance of their algorithm with respect to revenue or compare it across different heuristics.

Much research has been devoted in the field of stochastic programming to solve multi-stage recourse problems, and Birge (1997)[3] gives a summary of formulations and solution techniques. He gives a general model of the multistage stochastic linear programming with recourse. This formulation shares some characteristics with the problem at hand, since the inventory of ads available changes from one break to the next based on the realization of break size. Birge goes on to describe solution procedures such as extreme-point methods, interior point methods and column splitting. However, the challenges we face are different as our problem is an integer programming problem, while considerations of integrality are not touched upon.

Techniques for solving Stochastic Integer Programs are available in Birge and Louveaux (1997) [4]. The modified L-Shaped method suggested by them integrates branch and bound with the standard L-shaped method, thus an extra step is added where integrality constraints are checked for every feasibility cut introduced as part of the L-Shaped method. However they also state
that “loosely stated, for this class of problems, is very unlikely that an algorithm will be found that would solve the problem in a number of operations polynomial in the problem data... If the second stage of a stochastic problem corresponds to an NP-hard problem, it is pointless to design an exact method that would require the solution of the second stage for each realization of the random variable”. Thus, it warrants a study of heuristics or alternate algorithms that can actually run in polynomial time - even if their solutions are global sub-optimal - that can improve on the performance of schedules created manually. More recently, Hanveld and Van der Vlerk(1999) [18] survey structural properties of and algorithms for stochastic integer programming models, mainly considering linear two stage models with mixed integer recourse (and their multi-stage extensions). However, they also observe that “special purpose algorithms will turn out to be necessary to obtain good computational results for many real-life applications.” Sen (2005) [24] studies algorithms for both two-stage as well as multi-stage stochastic mixed-integer programs. He presents stage wise (resource directive) decomposition methods for two-stage models, and scenario (price directive) decomposition methods for multi-stage models. He also studies a variety of structures ranging from models that allow randomness in all data elements, to those that allow only specific elements to be influenced by randomness. He discusses branch and price and Lagrangian relaxation for multi-stage SMIP, but states that stage wise decomposition algorithms for the two stage case but states that scalability of the stage wise decomposition to multi-stage scenarios is suspect.

The literature survey on integer stochastic programming with recourse conveys that a specialized algorithm is in order for us to solve the multistage stochastic integer program we have in hand.
2.4 Revenue Management in Media Applications

Literature for revenue management for advertising in TV broadcasting has looked at the joint order acceptance and scheduling problem with deterministic break lengths. For example, Kimms and Muller-Bungart (2007) [17] formulate an integer program that maximizes the broadcaster's revenue, while taking into account non-conflicting product constraints and specific scheduling requests. The authors also propose several heuristics and conduct extensive numerical analyses that compare performance across the different solution methods. Bollapragada and Garbiras (2003) [5] also discuss ad scheduling but assume a deterministic audience distribution and given client preferences. They automate the commercial scheduling process while generating near optimal schedules to meet constraints (such as product conflict requirements and position percentage), and have implemented it in NBC. Zhang (2006) [30] uses a hierarchical structure using a model that uses a two step hierarchical approach, where winners (advertisers) are selected first and then slots are assigned to selected commercials.

There has also been work done in the area of slot allocation and contract selection for deciding on the inventory of ads a network has at the time of broadcasting. Araman and Popescu (2007) [1] develop a model for allocating advertising slots between up front and scatter markets under audience uncertainty in up front and operational planning decisions. Kimms and Muller Bungart (2007) [17] discuss simultaneous optimization of optimal contract selection and ad scheduling. They provide heuristics for optimal ad scheduling based on contract constraints (such as at which position in the break an ad can be shown, etc.) Our focus in this dissertation, however, will be on the optimal selection of ads in a given break and not on contract selection.

Finally, Degraeve and DeReyck (2003) [8] discuss broadcast ad scheduling using SMS. Their model uses a linear decomposition of three schedules that
are prepared before the broadcast begins, given a limited capacity of broad-
cast time slots, maximizing customer response and revenues from retailers
paying for each broadcast. The problem we analyze has different parameters
from those outlined in the papers above. Traditional ad scheduling heuris-
tics consider deterministic break sizes, and differ only in the number and
type of constraints they face. To the best of our knowledge, the problem
of scheduling ads in stochastic break sizes under similar constraints as those
faced when break sizes are deterministic has never been studied before.
Chapter 3

Optimal Policy

In this chapter, we will determine optimal selection policies that maximize expected revenue under some simplifying assumptions. Our objective is to find a general set of rules that aids ad schedulers when faced with random break lengths.

These rules are also applicable for a class of bin scheduling and cutting stock problems where bin sizes are non-deterministic and items are of known weight and value. Previous research in this area has focused on randomness in item value and weight; our contribution will be to add to this literature by considering randomness in bin sizes, and extend it to cases where the number of bins is also stochastic. We will use the terms “breaks” and “ads” throughout this document, but these can be substituted with “bins” and “items” for the general stochastic bin scheduling problem.

We propose a dynamic programming solution methodology where the return function is a preference selection criterion and illustrate the conditions required to guarantee optimality of the selection. We also contrast the behavior of the Optimal Policy with that of the Greedy Policy, and draw insights and implications.
3.1 Assumptions

We consider a scenario where breaks \( \{b_1, b_2, \ldots, b_n\} \) occur sequentially. The number of breaks has an upper bound \( N \geq k \geq 1 \), and the capacities of the breaks follow a known distribution, and are IID.

A decision has to be made as to what ads are to be put in the next available break. For simplicity, we assume that we are always planning for break \( b_1 \), and the index \( n \) is the number of remaining breaks expected to occur.

We assume that we only have two types of ads in the inventory: small ads of size \( S \) and large ads of size \( L = 2S \). This is representative of the types of ads currently used in American television networks, where ads are usually of either 15 second or 30 second durations.

We assume that the number of ads we have in our inventory is infinite, with possibly a fraction of those ads having a non-zero value. Further, the ads are arranged in descending order of value.

Thus, if \( \mathbb{S} \) and \( \mathbb{L} \) are the sets of small and large ads respectively, then

\[
\mathbb{S} = \{s_1, s_2, \ldots, 0, \ldots\} \\
\mathbb{L} = \{l_1, l_2, \ldots, 0, \ldots\}
\]

where \( s_i \geq s_{i+1} \geq 0 \) and \( l_i \geq l_{i+1} \geq 0 \ \forall i \). We consider two scenarios: the size of the current break is either known to the scheduler before he begins scheduling, or it is unknown. For each scenario, we consider the following cases:

1. A base case, where the size of each break is limited to either \( S \) or \( L \), and the number of breaks is fixed and known in advance,

2. An extension where the number of breaks is bounded but not fixed,

3. An extension where break sizes are any bounded multiple of \( S \),
4. An extension where break sizes are any bounded multiple of $S$ and the number of breaks is stochastic.

We assume that the revenue earned by each ad includes the sponsor status of an advertiser. We do not discuss per-client service levels in this chapter, instead focusing on service levels of large and small ads in general. We discuss per-client service levels in Chapter 5.

We define the Greedy Policy as follows:

**Definition 3.1** (Greedy Policy). *For each break, select the combination of ads that earns the highest revenue in that break.*

We also assume that ads that are not fully aired do not earn any revenue. Thus, by Definition 3.1, the Greedy Policy only selects ads that can be aired completely within the break.

### 3.2 Known Break Size

Our motivation to study this scenario comes from Cricket broadcasts where the on-field director can predict the length of the current break, depending on the type of break being taken and the state of the game. The ad schedulers then decide what ads to show in the current break based on the advice given by the on-field director, keeping in mind that the sizes of subsequent breaks are unknown.

This scenario could also occur in freight shipping where the shipper knows the capacity available in the next arriving ship but not those of subsequent ships, and has to build an appropriate consignment given that an certain number of ships are expected to follow.

Although the size of the current break is known, the scheduler should look ahead to decide his selection, so that revenue earned over all $k$ breaks is maximized. The Greedy Policy is not globally optimal because it fails to consider the subsequent breaks and the stochasticity in their sizes.
We now look at the four cases mentioned in section 3.1, and discuss Optimal Policy for each.

### 3.2.1 Base Case

We assume that breaks are either of size $S$ or of size $L = 2S$, and there is no uncertainty about the number of breaks remaining. Breaks of size $S$ arrive with probability $p$ and breaks of size $L$ arrive with probability $1 - p$.

**One break**

Only one ad of size $S$ can be scheduled in a small break. Thus, if $b_1 = S$, $s_1$ is always packed. It is trivial to prove that this is the case irrespective of the number of breaks left, hence we will not discuss selection policy when the current break size is $S$ in subsequent sections.

In a large break, however, one ad of size $L$ or two ads of size $S$ can be aired. Selection for a large break when $k = 1$ reduces to a Greedy Policy. The scheduler chooses between the more profitable of $(s_1, s_2)$ and $l_1$. We state the following lemma without proof:

**Lemma 3.1.** When $k = 1$, and $b_1 = L$, the optimal policy is to:

1. select $l_1$ if $l_1 \geq s_1 + s_2$
2. select $(s_1, s_2)$ if $s_1 + s_2 > l_1$

**Two breaks**

For $k = 2$, we can write an exhaustive list of all the possible cases. These are listed out in Table 3.1 From the table, we see that when $b_1 = L$, the scheduler would choose $(s_1, s_2)$ if:

$$
(s_1 + s_2) + ps_3 + (1 - p) \max\{l_1, s_3 + s_4\} \geq l_1 + ps_1 + (1 - p) \max\{l_2, s_1 + s_2\}
$$

(3.1)
Optimal Policy

We use the above equation to prove the optimal policy.

Lemma 3.2. When $k = 2$, and $b_1 = L$, the optimal policy is to:

1. select $l_1$ if $l_1 \geq s_2 + s_3$

2. select $(s_1, s_2)$ if $l_1 < s_2 + s_3$

Proof. Case 1:

Let $l_1 \geq s_2 + s_3$. Let $R_l$ and $R_s$ be the revenues earned by selecting $l_1$ first and $s_1 + s_2$ first respectively. From equation 3.1, we can write:

$$R_l - R_s$$
$$= [l_1 + ps_1 + (1-p) \max\{l_2, s_1 + s_2\}]$$
$$- [(s_1 + s_2) + ps_3 + (1-p) \max\{l_1, s_3 + s_4\}]$$
$$\geq [l_1 + ps_1 + (1-p)(s_1 + s_2)] - [(s_1 + s_2) + ps_3 + (1-p)l_1]$$
$$\quad (\because l_1 \geq s_2 + s_3 \geq s_3 + s_4)$$
$$\geq p(l_1 - (s_2 + s_3))$$
$$\geq 0 \quad (\because l_1 \geq s_2 + s_3)$$

Therefore, when $l_1 \geq s_2 + s_3$, $R_l \geq R_s$, so $l_1$ is packed first. Similarly, we can prove Case 2 by showing that $R_s - R_l \geq 0$ when $l_1 < s_2 + s_3$.\hfill \square
While the Greedy Policy compares \( l_1 \) and \( s_1 + s_2 \), Lemma 3.2 compares \( l_1 \) and \( s_2 + s_3 \), setting a lower threshold for \( l_1 \) to be optimal, and improving its chances of being selected.

If \( b_2 = L \) and Case 1 applies, then the Optimal Policy selects \( s_1 + s_2 \) for \( b_2 \), (from Lemma 3.1), while the Greedy Policy selects \( l_1 \), and both policies earn equally.

Consider, however, the case when \( b_2 = S \). The Greedy Policy then earns \( s_1 + s_2 + s_3 \). The Optimal Policy, however, earns \( l_1 + s_1 \) which is more than the revenue earned by the Greedy Policy, since \( l_1 \geq s_2 + s_3 \).

When \( l_1 < s_2 + s_3 \), and we schedule \( l_1 \) in \( b_1 \), then we are forced to schedule \( s_1 \) in \( b_2 \) when \( b_2 = S \), earning \( l_1 + s_1 \). However, we know that \( s_1 + s_2 + s_3 > l_1 + s_1 \), therefore it is sub-optimal to schedule \( l_1 \) in \( b_1 \). Thus, the condition in Case 1 describes the optimal threshold for \( l_1 \) to be an attractive candidate for \( b_1 \).

We next take a look at the multiple break case.

**Multiple breaks**

When the number of breaks remaining is greater than two, we see that the policy outlined in Lemma 3.2 extends to look ahead to all the remaining breaks.

**Theorem 3.1.** When \( k = n \), and \( b_1 = L \), the optimal policy is to:

1. select \( l_1 \) if \( l_1 \geq s_n + s_{n+1} \)

2. select \((s_1, s_2)\) if \( l_1 < s_n + s_{n+1} \)

**Proof.** Proof is in Appendix A.

Consider the situation where all breaks subsequent to \( b_1 \) are of size \( S \). If we scheduled \((s_1, s_2)\) in \( b_1 \), we would have scheduled ads \((s_1, s_2, \ldots, s_n, s_{n+1})\) at the end of the planning period. Therefore, we get a higher revenue by selecting \((l_1, s_1, s_2, \ldots, s_{n-1})\), since \( l_1 \geq s_n + s_{n+1} \).
It is trivial to see that if one or more large breaks arrive instead, the Optimal Policy would earn at least as much as the Greedy.

As the number of breaks remaining increases, the threshold above which it becomes optimal to select $l_1$ decreases, and selecting $l_1$ becomes more attractive. Service levels of large ads, therefore, are higher with the Optimal Policy than with the Greedy Policy when breaks of size $L$ occur and values of large ads fall in the range $(s_n + s_{n+1}, s_1 + s_2)$.

Service levels of small ads are higher with the Greedy Policy than with the Optimal Policy, since the Greedy Policy has a higher threshold for scheduling large ads, and is more likely to schedule small ads for the large breaks as well.

In extreme cases, the Greedy Policy can schedule all small ads in the long breaks and be left with small ads of value zero, losing the opportunity to earn from small breaks.

### 3.2.2 Stochastic Number of Breaks

In this section, we consider a scenario where the number of breaks remaining has an upper bound, but is not fixed. To model this scenario, we assume that all breaks arrive, but that some breaks have size zero. We assign probabilities $p_0$ that a break arrives with size zero, $p_1$ that the break has size $S$ and $p_2$ that the break has size $L$, where $p_0 + p_1 + p_2 = 1$.

The motivation to model this scenario comes from cases where broadcasting networks cannot predict how many breaks they can take in the game (but know the maximum possible number of breaks possible). In cricket, for instance, a break is taken every time a wicket falls. The maximum number of wickets that can fall in an innings is ten, however, the actual number of wickets that fall in each innings may be lower. This scenario can occur in other situations as well, for instance in freight shipping, where ships can arrive but have no space for accommodating the consignment to be shipped.

We study the decisions of the scheduler when the number of breaks remaining are one and two, then use induction to find the optimal policy for
the general case.

One break

Selecting ads to be scheduled for one break of known length is trivial, and exactly the same as outlined in Lemma 3.1. The introduction of breaks of size zero does not affect ad selection because we already know the size of the (one) break that has to be scheduled. As before, if the break is of size $S$, we schedule $s_1$, and if it is $L$, we choose the larger of $l_1$ and $(s_1 + s_2)$. If it is of size zero, we schedule nothing.

Two breaks

When there are two breaks in the planning period, the Optimal Policy must look ahead to $b_2$ to decide on the optimal selection for $b_1$. The second break could be of size zero, which would make the two break case the same as the one break case. Therefore, with probability $p_0$, the greater of $l_1$ and $(s_1 + s_2)$ should be selected. However, with probability $p_1 + p_2$, $b_2$ could be non-zero, and Lemma 3.2 applies. We state the Optimal Policy for two breaks formally below.

Lemma 3.3. When $k = 2$, and $b_1 = L$, the optimal policy when $p_0 \geq 0$ is to:

1. select $l_1$ if $(p_0 + p_1)l_1 \geq p_0(s_1 + s_2) + p_1(s_2 + s_3)$

2. select $(s_1, s_2)$ otherwise

Proof. Case 1:

Let $(p_0 + p_1)l_1 \geq p_0(s_1 + s_2) + p_1(s_2 + s_3)$, and let $R_l$ and $R_s$ be the revenues earned by selecting $l_1$ first and $s_1 + s_2$ first respectively. Then:

$$R_l - R_s = [l_1 + p_00 + p_1s_1 + p_2\max\{l_2, s_1 + s_2\}]$$

$$- [(s_1 + s_2) + p_00 + p_1s_3 + p_2\max\{l_3, s_3 + s_4\}]$$
However,

\[(p_0 + p_1)l_1 \geq p_0(s_1 + s_2) + p_1(s_2 + s_3)\]

\[\implies (p_0 + p_1)l_1 \geq p_0(s_3 + s_4) + p_1(s_3 + s_4)\]

\[\implies l_1 \geq (s_3 + s_4) \text{ assuming } p_0 + p_1 > 0\]

We assume that \(p_0 + p_1 > 0\), otherwise we would have the degenerate case where \(p_2 = 1\) and all breaks would be of the same size \(L\). Substituting for \(R_l - R_s\), we get:

\[R_l - R_s \geq [l_1 + p_1s_1 + p_2(s_1 + s_2)] - [(s_1 + s_2) + p_1s_3 + p_2l_1]\]

\[\geq (p_0 + p_1)l_1 - p_0(s_1 + s_2) - p_1(s_2 + s_3)\]

\[\geq 0 \quad (\because (p_0 + p_1)l_1 \geq p_0(s_1 + s_2) + p_1(s_2 + s_3))\]

Therefore, when \((p_0 + p_1)l_1 \geq p_0(s_1 + s_2) + p_1(s_2 + s_3)\), \(R_l \geq R_s\), so \(l_1\) is selected. Similarly, we can prove Case 2 by showing that \(R_s - R_l \geq 0\) when \((p_0 + p_1)l_1 < p_0(s_1 + s_2) + p_1(s_2 + s_3)\).

If \(b_2 = 0\), we compare \(l_1\) and \((s_1 + s_2)\), since nothing can be scheduled into \(b_2\). In other words, select \(l_1\) if:

\[p_0l_1 \geq p_0(s_1 + s_2)\]  \hspace{1cm} (3.2)

If \(b_2 = S\), selecting \(l_1\) in \(b_1\) earns \(l_1 + s_1\) and selecting \((s_1, s_2)\) in \(b_1\) earns \((s_1 + s_2 + s_3)\), so we would select \(l_1\) if:

\[p_1(l_1 + s_1) \geq p_1(s_1 + s_2 + s_3)\]

\[\implies p_1l_1 \geq p_1(s_2 + s_3)\]  \hspace{1cm} (3.3)

Finally, when \(b_2 = L\), the Optimal Policy selects \((s_1, s_2)\) in \(b_2\) and the Greedy Policy selects \(l_1\) in \(b_2\), therefore both policies earn equally. There-
fore, Lemma 3.3 checks the expected value earned by selecting $l_1$ against the expected value earned by selecting $(s_1, s_2)$ when $b_2$ is not of size $L$. Combining Equation 3.2 and Equation 3.3, we get the condition described in Case 1.

We next look at the multiple break case and use induction to prove the Optimal Policy.

**Multiple Breaks**

When there are multiple breaks remaining, we look ahead to all remaining breaks to form the rule. We state the Optimal Policy formally as follows.

**Theorem 3.2.** When $k = n$, $p_0 \geq 0$, and $b_1 = L$, the optimal policy is to:

1. select $l_1$ if $(p_0 + p_1)^{n-1}l_1 \geq \sum_{i=0}^{n-1} \left[ \binom{n-1}{i} p_0^{(n-1)-i} p_1^i (s_{i+1} + s_{i+2}) \right]
2. select $(s_1, s_2)$ otherwise

*Proof.* Proof is in Appendix A.

The intuition behind Theorem 3.2 is similar to that of Lemma 3.3. If a break of size $L$ were to occur subsequent to the current break, both the Greedy Policy and the Optimal Policy would earn equal revenues.

The Optimal Policy estimates the value of scheduling $l_1$ assuming that none of the subsequent breaks are of size $L$. The reasoning is similar to the one used in Equation 3.2 and Equation 3.3, except in the multiple break case where the number of breaks is $n - 1$, $i$ breaks of size $S$ and $n - 1 - i$ breaks of size zero can occur with probability $\binom{n-1}{i} p_0^{(n-1)-i} p_1^i$. The rest of the reasoning follows.

The threshold above which $l_1$ is an attractive ad to be scheduled has increased due to the introduction of $p_0$. For instance, in the base case, for $l_1$ to be optimal when two breaks remained, $l_1$ had to be greater than $s_2 + s_3$. 
However, \( l_1 \) now has to be greater than

\[
\frac{1}{(p_0 + p_1)}(p_0(s_1 + s_2) + p_1(s_2 + s_3)) \geq s_2 + s_3
\]

The Optimal Policy weighs the advantages of scheduling the large ad against the probability that many of the subsequent breaks could be of size zero and hence earn nothing. It does this by adjusting the threshold for scheduling \( l_1 \) based on the break size distribution.

The Optimal Policy helps networks decide on their preferred mix of ads based on the distribution of break sizes. When there is randomness in the number of breaks, networks require a higher value for large ads to be shown, compared to the case where the number of breaks is fixed.

### 3.2.3 Multiple Break Sizes

In this section, we study the Optimal Policy when break sizes are distributed between \([S, 2S, \ldots, MS]\). In addition, the size of the current break is assumed to be \( mS \), where \( m \leq M \). As before, we study Optimal Policy when the number of breaks remaining are one and two, and use induction to prove the Optimal Policy for the multiple break case.

**One Break**

For the one break case, we choose a set of ads that give us the best possible revenue within the known break size \( mS \). This corresponds exactly with the Greedy Policy.

**Lemma 3.4.** If \( b_1 = mS \) and \( k = 1 \), then the Optimal Policy is to select \((l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda})\), where \( \lambda \) is the largest index such that:

- \( l_\lambda \geq s_{m-2\lambda+1} + s_{m-2\lambda+2} \)
- \( 2\lambda \leq m \)
Proof. Let \( \hat{O} = (l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda}) \) be the set that we want to prove is optimal. To prove optimality of \( \hat{O} \), we have to prove that any change to \( \hat{O} \) will cause the revenue to decrease.

We note that if \( \exists \lambda : l_\lambda \in \hat{O} \), then \( \forall i \leq \lambda, \ l_i \in \hat{O} \), since

\[
l_i \geq l_\lambda \geq s_{m-2\lambda+1} + s_{m-2\lambda+2} \geq s_{m-2i+1} + s_{m-2i+2} \quad \text{and} \quad 2i \leq 2\lambda \leq m
\]

Now consider the case where we do not select some \( l_i \in [l_1, l_\lambda] \). Then, we can either include \((s_{m-2\lambda+1}, s_{m-2\lambda+2})\), or we can include \( l_{\lambda+1} \).

Selecting \((s_{m-2\lambda+1}, s_{m-2\lambda+2})\) is inferior because \( l_i \geq l_\lambda \geq (s_{m-2\lambda+1} + s_{m-2\lambda+2}) \), therefore our revenue will decrease. Similarly, \( l_i \geq l_{\lambda+1} \), so substituting \( l_i \) with \( l_{\lambda+1} \) will also decrease our revenue. It is trivial to prove that changing any \( s_i \in [s_1, s_{m-2\lambda}] \) with \( s_{m-2\lambda+1} \) will similarly cause a drop in revenue. Therefore \( \hat{O} \) is the optimal set of ads to be scheduled when \( b = 1 \). \( \square \)

The proof for Lemma 3.4 shows that every \( l_i \in [l_1, l_\lambda] \) should be scheduled for maximum revenue. In general, it is sufficient to prove that \( l_\lambda \) (as defined in Lemma 3.4) must be scheduled; since every \( l_i \geq l_\lambda \) belongs to the optimal set, optimality of selecting \( l_\lambda \) implies optimality of selecting \( l_i \).

We now look at the two break case.

Two Breaks

The two breaks case follows Lemma 3.2; we now check if the large ad \( l_\lambda \) earns more than the sum of the small ads at indexes \( m - 2\lambda + 2 \) and \( m - 2\lambda + 3 \).

We state the Lemma formally below.

**Lemma 3.5.** If \( b_1 = mS \) and \( k = 2 \), then the Optimal Policy is to select \((l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda})\), where \( \lambda \) is the largest index such that:

- \( l_\lambda \geq s_{m-2\lambda+2} + s_{m-2\lambda+3} \)
- \( 2\lambda \leq m \)
Proof. As before, let \( \hat{O} = (l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda}) \).

Let:

\[
O_l = (l_1, \ldots, l_\lambda, l_{\lambda+1}, s_1, \ldots, s_{m-2\lambda-2})
\]

\[
O_s = (l_1, \ldots, l_{\lambda-1}, s_1, \ldots, s_{m-2\lambda+2})
\]

Let \( R_o \), be the revenue earned by selecting the ads in \( \hat{O} \), and let \( R_l \) and \( R_s \) be the revenues earned by selecting \( O_l \) and \( O_s \) respectively.

We have to prove that:

1. \( R_o \geq R_l \)
2. \( R_o \geq R_s \)

It is trivial to prove that if \( l_{\lambda+1} \geq s_{m-2\lambda} + s_{m-2\lambda+1} \) and \( 2(\lambda+1) > m \), then substituting \( l_i \geq l_{\lambda+1} \) from \( \hat{O} \) and introducing \( l_{\lambda+1} \) is sub optimal (for the same reason that we pack \( l_1 \) in a break of length \( L \), not \( l_2 \)).

We use \( V_i(l, s) \) to denote expected revenue earned from scheduling ads for break \( i \) onwards, and the indexes of the first large ad and the first small ad in our inventory are \( l \) and \( s \) respectively. Then,

\[
R_o = \sum_{i=1}^{\lambda} l_i + \sum_{j=1}^{m-2\lambda} s_j + V_2(\lambda+1, m-2\lambda+1) \tag{3.4}
\]

\[
R_l = \sum_{i=1}^{\lambda+1} l_i + \sum_{j=1}^{m-2\lambda-2} s_j + V_2(\lambda+2, m-2\lambda-1) \tag{3.5}
\]

\[
R_s = \sum_{i=1}^{\lambda-1} l_i + \sum_{j=1}^{m-2\lambda+2} s_j + V_2(\lambda, m-2\lambda+3) \tag{3.6}
\]

Case 1:

\[
R_o - R_l = \sum_{i=1}^{\lambda} l_i + \sum_{j=1}^{m-2\lambda} s_j + V_2(\lambda+1, m-2\lambda+1)
- \left[ \sum_{i=1}^{\lambda+1} l_i + \sum_{j=1}^{m-2\lambda-2} s_j + V_2(\lambda+2, m-2\lambda-1) \right]
\]
\[ = -l_{\lambda+1} + s_{m-2\lambda-1} + s_{m-2\lambda} \\
+ V_2(\lambda + 1, m - 2\lambda + 1) - V_2(\lambda + 2, m - 2\lambda - 1) \]

If \( l_{\lambda+1} < s_{m-2\lambda} + s_{m-2\lambda+1} \), then \( l_{\lambda+2} < s_{m-2\lambda-2} + s_{m-2\lambda-1} \). Then by induction, if \( b_2 \geq 2\beta \), \( V_2(\lambda + 2, m - 2\lambda - 1) \) would earn \((s_{m-2\lambda-1} + s_{m-2\lambda})\) followed by the sum of values of ads selected from the set \( \{l_{\lambda+2}, \ldots, s_{m-2\lambda+1}, \ldots\} \).

For a break of corresponding size, \( V_2(\lambda + 1, m - 2\lambda + 1) \) would earn at least \( l_{\lambda+1} \) followed by the sum of values of ads from the set \( \{l_{\lambda+2}, \ldots, s_{m-2\lambda+1}, \ldots\} \).

Thus, we can write:

\[
V_1(\lambda + 1, m - 2\lambda + 1) - V_1(\lambda + 2, m - 2\lambda - 1) \\
\geq p_1(s_{m-2\lambda+1} - s_{m-2\lambda-1}) \\
+ \sum_{r=2}^{M} p_r (l_{\lambda+1} - [s_{m-2\lambda-1} + s_{m-2\lambda}]) \tag{3.7}
\]

Substituting for \( R_o - R_l \), we get:

\[
R_o - R_l \geq -l_{\lambda+1} + s_{m-2\lambda-1} + s_{m-2\lambda} \\
+ p_1(s_{m-2\lambda+1} - s_{m-2\lambda-1}) \\
+ \sum_{r=2}^{M} p_r (l_{\lambda+1} - [s_{m-2\lambda-1} + s_{m-2\lambda}]) \\
\geq p_1 [-l_{\lambda+1} + s_{m-2\lambda-1} + s_{m-2\lambda} + s_{m-2\lambda+1} - s_{m-2\lambda-1}] \\
\geq p_1 [-l_{\lambda+1} + s_{m-2\lambda} + s_{m-2\lambda+1}] \\
> 0 \quad (\because l_{\lambda+1} < s_{m-2\lambda} + s_{m-2\lambda+1})
\]

**Case 2:** \( R_o - R_s \geq 0 \) can be similarly proved:

\[
R_o - R_s = \sum_{i=1}^{\lambda} l_i + \sum_{j=1}^{m-2\lambda} s_j + V_1(\lambda + 1, m - 2\lambda + 1) \\
- \left[ \sum_{i=1}^{\lambda-1} l_i + \sum_{j=1}^{m-2\lambda+2} s_j + V_1(\lambda, m - 2\lambda + 3) \right]
\]
= l_\lambda - (s_{m-2\lambda+1} + s_{m-2\lambda+2}) \\
+ V_1(\lambda + 1, m - 2\lambda + 1) - V_1(\lambda, m - 2\lambda + 3)

Using similar arguments as before,

\[ V_1(\lambda + 1, m - 2\lambda + 1) - V_1(\lambda, m - 2\lambda + 3) \geq p_1(s_{m-2\lambda+1} - s_{m-2\lambda+3}) \]
\[ + \sum_{r=2}^{M} p_r (-l_\lambda + [s_{m-2\lambda+1} + s_{m-2\lambda+2}]) \]

After substituting for \( R_o - R_s \), we get:

\[ R_o - R_s \geq l_\lambda - (s_{m-2\lambda+1} + s_{m-2\lambda+2}) \]
\[ + p_1(s_{m-2\lambda+1} - s_{m-2\lambda+3}) \]
\[ + \sum_{r=2}^{M} p_r (-l_\lambda + [s_{m-2\lambda+1} + s_{m-2\lambda+2}]) \]
\[ \geq p_1(l_\lambda - (s_{m-2\lambda+2} + s_{m-2\lambda+3})) \]
\[ \geq 0 \quad \therefore l_\lambda \geq s_{m-2\lambda+2} + s_{m-2\lambda+3} \]

By setting the threshold on the least valuable large ad that can be scheduled, the scheduler only needs to check backwards from \( l_{\lfloor m/2 \rfloor} \) for the least valuable large ad that satisfies Case 1. When the appropriate ad is found, all large ads that have greater value are scheduled, and the remaining time in the break is filled with the most valuable small ads.

**Multiple Breaks**

The multiple breaks case uses induction, and the intuition behind the proof is similar to that used in Lemma 3.5. We state the Theorem formally below.
Theorem 3.3. If \( b_1 = mS \) and \( k = n \), then the Optimal Policy is to select \((l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda})\), where:

- \( l_\lambda \geq s_{m-2\lambda+n} + s_{m-2\lambda+n+1} \)
- \( 2\lambda \leq m \)

Proof. Proof is given in Appendix A.

Consider the case where \( k = 1 \), and the break of size \( mS \) is split into \( \lfloor \frac{m}{2} \rfloor \) breaks of size \( 2S \) (and an additional break of size \( S \), if \( m \) is an odd number). Then by Theorem 3.1, \( l_1 \) would be compared to \((s_{m/2} + s_{m/2+1})\), and if it is lower, to \((s_{m/2+1} + s_{m/2+2})\), and so on until either \( l_1 \) is greater than some combination of \( s_i + s_{i+1} \), where \( i > m/2 \), or the large breaks are exhausted. If \( l_1 \) is not scheduled in the first \( \lfloor \frac{m}{2} \rfloor - 1 \) breaks, then \( l_1 \) will finally be compared to \((s_{m-1} + s_m)\), which corresponds to \((s_{m-2\lambda+1} + s_{m-2\lambda+2})\) where \( \lambda = 1 \).

To see why this is true for any \( l_\lambda \) where \( \lambda > 1 \), assume that it is true for \( l_1, \ldots, l_{\lambda-1} \). If \( l_{\lambda-1} \) has been selected for airing, then \( l_{\lambda-1} \geq (s_{m-2\lambda+3} + s_{m-2\lambda+4}) \). If \( l_{\lambda-1} \) is scheduled for the last large break, then ads \((s_{m-2\lambda+3}, s_{m-2\lambda+4})\) are not scheduled, and the last small ads to be scheduled are \((s_{m-2\lambda+1}, s_{m-2\lambda+2})\).

For \( l_\lambda \) to be an attractive candidate to be scheduled, therefore, it has to be more valuable than the two least valuable small ads which have been selected: \((s_{m-2\lambda+1}, s_{m-2\lambda+2})\).

When \( k > 1 \), the index of small ads that \( l_i \) has to be compared against increases by exactly \( k \), because the optimal policy assumes the worst case where every break subsequent to the current one is small, similar to the intuition in Section 3.2.1.

From a managerial perspective, the Optimal Policy reduces complexity; the Greedy Policy would have to generate every combination of ads that fits the break and select the most profitable. Thus, despite the added complexity of having breaks of multiple sizes, the Optimal Policy scales well.
3.2.4 Stochastic Number of Breaks of Multiple Sizes

In this section, we study the Optimal Policy when break sizes are distributed between $[0, S, 2S, \ldots, MS]$. The size of the current break is assumed to be $mS$, where $m \leq M$. As we did in Section 3.2.2, we allow breaks of size zero to model the case where the number of breaks is stochastic. We use probability $p_i$ to denote the probability of a break of size $i$ occurring, where $i \in [0, M]$ and $\sum_{i=0}^{M} p_i = 1$.

We study the Optimal Policy when the number of breaks remaining are one and two, and use induction to prove the Optimal Policy for the multiple break case.

One Break

When the current break is the only break to be scheduled, and the size of the break is known, we use the same policy as outlined in Lemma 3.4. We reiterate the lemma here without proof.

**Lemma 3.6.** If $b_1 = mS$, $p_0 \geq 0$, and $k = 1$, then the Optimal Policy is to select $(l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda})$, where $\lambda$ is the largest index such that $2\lambda \leq m$ and:

$$l_\lambda \geq s_{m-2\lambda+1} + s_{m-2\lambda+2}$$

Two Breaks

We know that Lemma 3.6 applies for $b_2$ when it is the only break remaining. We will use induction to prove optimality when two breaks remain.

As we did for Lemma 3.5, we prove that it is optimal to select $l_\lambda$ when it satisfies the rule for optimality, from which we can infer optimality of selecting all $l_i \geq l_\lambda$.

**Lemma 3.7.** If $b_1 = mS$, $p_0 \geq 0$, and $k = 2$, then the Optimal Policy is to select $(l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda})$, where $\lambda$ is the largest index such that $2\lambda \leq m$ and:
and:

\[(p_0 + p_1)l_\lambda \geq p_0(sm_{2\lambda+1} + sm_{2\lambda+2}) + p_1(sm_{2\lambda+2} + sm_{2\lambda+3})\]

Proof. Let:

\[\hat{O} = (l_1, \ldots, l_\lambda, s_1, \ldots, sm_{-2\lambda})\]
\[O_l = (l_1, \ldots, l_\lambda, l_{\lambda+1}, s_1, \ldots, sm_{-2\lambda-2})\]
\[O_s = (l_1, \ldots, l_{\lambda-1}, s_1, \ldots, sm_{-2\lambda+2})\]

Let \(R_o, R_l,\) and \(R_s\) denote the revenues earned by selecting \(\hat{O}, O_l\) and \(O_s\) respectively. We have to prove that

1. \(R_o \geq R_l\)
2. \(R_o \geq R_s\)

We again use \(V_i(l, s)\) to denote expected revenue earned when selecting ads for break \(i\) onwards, when the indexes of the first large ad and the first small ad are \(l\) and \(s\) respectively.

**Case 1:** By definition:

\[(p_0 + p_1)l_\lambda \geq p_0(sm_{-2\lambda+1} + sm_{-2\lambda+2}) + p_1(sm_{-2\lambda+2} + sm_{-2\lambda+3})\] (3.8)

and \((p_0 + p_1)l_{\lambda+1} < p_0(sm_{-2\lambda-1} + sm_{-2\lambda}) + p_1(sm_{-2\lambda} + sm_{-2\lambda+1})\) (3.9)

Then,

\[
R_o - R_l = \sum_{i=1}^{\lambda} l_i + \sum_{j=1}^{m-2\lambda} s_j + V_2(\lambda + 1, m - 2\lambda + 1)
- \left[ \sum_{i=1}^{\lambda+1} l_i + \sum_{j=1}^{m-2\lambda-2} s_j + V_2(\lambda + 2, m - 2\lambda - 1) \right]
= -l_{\lambda+1} + sm_{-2\lambda-1} + sm_{-2\lambda}
+ V_2(\lambda + 1, m - 2\lambda + 1) - V_2(\lambda + 2, m - 2\lambda - 1)
\]
From eq. 3.9, we have:

\[
(p_0 + p_1)l_{\lambda + 1} < p_0(s_{m - 2\lambda - 1} + s_{m - 2\lambda}) + p_1(s_{m - 2\lambda} + s_{m - 2\lambda + 1})
\]

\[
< p_0(s_{m - 2\lambda - 2} + s_{m - 2\lambda - 1}) + p_1(s_{m - 2\lambda - 1} + s_{m - 2\lambda})
\]

\[
\implies (p_0 + p_1)l_{\lambda + 2} < p_0(s_{m - 2\lambda - 2} + s_{m - 2\lambda - 1}) + p_1(s_{m - 2\lambda - 1} + s_{m - 2\lambda})
\]

(since \(l_{\lambda + 2} \leq l_{\lambda + 1}\))

Then by induction, for \(b_2 \geq 2S\), \(V_2(\lambda + 2, m - 2\lambda - 1)\) earns us \((s_{m - 2\lambda - 1} + s_{m - 2\lambda})\) followed by the sum of values of ads selected from the set \(\{l_{\lambda + 2}, \ldots, s_{m - 2\lambda + 1}, \ldots\}\). On the other hand, with \(V_2(\lambda + 1, m - 2\lambda + 1)\) we earn at least \(l_{\lambda + 1}\) followed by the sum of values of ads from the set \(\{l_{\lambda + 2}, \ldots, s_{m - 2\lambda + 1}, \ldots\}\). Thus, we can write:

\[
V_2(\lambda + 1, m - 2\lambda + 1) - V_2(\lambda + 2, m - 2\lambda - 1)
\]

\[
\geq p_1(s_{m - 2\lambda + 1} - s_{m - 2\lambda - 1})
\]

\[
+ \sum_{i=2}^{M} p_i (l_{\lambda + 1} - [s_{m - 2\lambda - 1} + s_{m - 2\lambda}])
\]

Substituting for \(R_o - R_l\), we get:

\[
R_o - R_l \geq -l_{\lambda + 1} + s_{m - 2\lambda - 1} + s_{m - 2\lambda}
\]

\[
+ p_1(s_{m - 2\lambda + 1} - s_{m - 2\lambda - 1})
\]

\[
+ \sum_{i=2}^{M} p_i (l_{\lambda + 1} - [s_{m - 2\lambda - 1} + s_{m - 2\lambda}])
\]

\[
\geq (p_0 + p_1) [-l_{\lambda + 1} + s_{m - 2\lambda - 1} + s_{m - 2\lambda}] + p_1(s_{m - 2\lambda + 1} - s_{m - 2\lambda - 1})
\]

\[
\geq -(p_0 + p_1)l_{\lambda + 1} + p_0(s_{m - 2\lambda - 1} + s_{m - 2\lambda}) + p_1(s_{m - 2\lambda} + s_{m - 2\lambda + 1})
\]

\[
> 0 \quad \text{(from eq. 3.9)}
\]

**Case 2** \(R_o - R_s \geq 0\) can be similarly proved using eq. 3.8.

It can be seen that the intuition behind Lemma 3.7 is similar to our
discussion in Section 3.2.3: the revenue earned by the Optimal Policy is equal to the case where we have \( \lfloor m/2 \rfloor \) breaks of size \( 2S \) (and one break of size \( S \), if \( m \) is an odd number). Since \( p_0 \geq 0 \), Theorem 3.2 would apply for each break of size \( 2S \).

The intuition behind the probabilities follows the discussion in Section 3.2.2. The Optimal Policy adjusts the threshold above which large ads are attractive based on the break size distribution, and these thresholds increase as \( p_0 \) increases and other probabilities decrease.

Thus the case of multiple break sizes with stochastic number of breaks can be seen as a combination of Sections 3.2.2 and 3.2.3.

Multiple Breaks

The multiple breaks case uses induction, and the intuition behind the proof is similar to that of Lemma 3.7. We state the Theorem formally below.

**Theorem 3.4.** If \( b_1 = mS \), \( p_0 \geq 0 \), and \( k = n \), then the Optimal Policy is to select \( (l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda}) \), where \( \lambda \) is the largest index such that \( 2\lambda \leq m \) and:

\[
(p_0 + p_1)^{n-1}l_\lambda \geq \sum_{i=0}^{n-1} \left[ \binom{n-1}{i} p_0^{n-1-i} p_1^i (s_{m-2\lambda+i+1} + s_{m-2\lambda+i+2}) \right]
\]

**Proof.** Proof is given in Appendix A.

As discussed in section 3.2.2, the threshold has once again increased for \( l_1 \); this is because of the introduction of breaks of size zero. Our insights from Section 3.2.2 still apply. The Optimal Policy sets a threshold, based on the break size distribution, that each large ad should exceed to be scheduled. As we have studied, the higher the probability of breaks of size zero, the higher the value that large ads should have to make them attractive candidates for scheduling, given the same set of small ads.

In this section, we have studied the Optimal Policy for scheduling ads in breaks of stochastic sizes, when the size of the break for which we are
currently scheduling is known but those of subsequent breaks is not. We conclude with the most general case, Theorem 3.4 where setting $M$ and $p_0$ to appropriate values will give us the rules described in Theorem 3.1, Theorem 3.2 and Theorem 3.3. Thus we have described the rule for the full set of scenarios when the scheduler is aware of the size of the break to be scheduled next.

In the next section, we study the Optimal Policy for cases when the scheduler does not know the size of any of the breaks, but knows only the distribution of break sizes based on which he can create a sequence of ads to be scheduled.

### 3.3 Unknown Break Size

In this section we study scenarios where we do not know the size of any of the breaks at the time of scheduling. Our motivation arises from cases where the break begins without the on-field director being able to advice the schedulers on what the break length is expected to be, and an ad schedule has to be made based only on the break length distribution. We assume that break sizes are IID.

As before, we discuss Optimal Policy for the four cases mentioned in section 3.1.

#### 3.3.1 Base Case

In the Base Case, breaks can be of two sizes, small ($S$) and large ($L = 2S$). A small break occurs with probability $p$, and a large break occurs with probability $1 - p$. We study our selection options when a choice has to be made between selecting either $(s_1, s_2)$ or $l_1$ before the break size is observed.
One break

When scheduling for one break unknown size, we choose the maximum of the expected revenues from selecting either $l_1$ or $(s_1, s_2)$. If a small break occurs, selecting $(s_1, s_2)$ earns $s_1$, since only $s_1$ can be completely aired. Selecting $l_1$ earns nothing, since it cannot be fully aired. If a large break occurs, selecting $(s_1, s_2)$ and $l_1$ earn $s_1 + s_2$ and $l_1$ respectively.

Therefore the expected revenue from selecting $(s_1, s_2)$ is $ps_1 + (1-p)(s_1 + s_2)$, and the expected revenue from selecting $l_1$ is $(1-p)l_1$.

We state the following lemma without proof:

**Lemma 3.8.** If $k = 1$, and the break size is unknown, the Optimal Policy is to:

1. select $l_1$ if $(1-p)l_1 \geq ps_1 + (1-p)(s_1 + s_2)$
2. select $(s_1, s_2)$ otherwise

We next look at the Optimal Policy when two breaks remain.

Two breaks

Unlike Lemma 3.2, where the Optimal Policy compared $l_1$ and $s_n + s_{n+1}$, the absence of ex-ante information forces us to choose myopically between the expected revenues earned by selecting $l_1$ and $(s_1, s_2)$ irrespective of the number of breaks remaining. We state the optimal policy formally as follows.

**Lemma 3.9.** When $k = 2$, and the break size is unknown, the optimal policy is to:

1. select $l_1$ if $(1-p)l_1 \geq ps_1 + (1-p)(s_1 + s_2)$
2. select $(s_1, s_2)$ if $(1-p)l_1 < ps_2 + (1-p)(s_2 + s_3)$
3. select either $l_1$ or $(s_1, s_2)$ otherwise
Proof. From Lemma 3.8, the rule holds good for one break. Let $R_l$ and $R_s$ denote the expected revenues from scheduling $l_1$ and $(s_1, s_2)$ respectively. Let $V_i(l, s)$ denote the revenue earned from break $i$ onwards, when the indexes of the first large ad and the first small ad are $l$ and $s$ respectively.

**Case 1:** When $(1-p)l_1 \geq ps_1 + (1-p)(s_1 + s_2)$,

$$R_l - R_s = p(0 + V_2(1, 1)) + (1-p)(l_1 + V_2(2, 1)) - p(s_1 + V_2(1, 2)) - (1-p)(s_1 + s_2 + V_2(1, 3))$$

$$= (1-p)l_1 - ps_1 - (1-p)(s_1 + s_2) + p[V_2(1, 1) - V_2(1, 2)] + (1-p)[V_2(2, 1) - V_2(1, 3)]$$

$$\geq (1-p)l_1 - ps_1 - (1-p)(s_1 + s_2) + p(V_2(1, 1) - (1-p)l_1)$$

$$+ (1-p)[p(s_1) + (1-p)(s_1 + s_2) - (1-p)l_1]$$

$$(\because V_2(2, 1) \geq p(s_1) + (1-p)(s_1 + s_2))$$

$$\geq p[(1-p)l_1 - ps_1 - (1-p)(s_1 + s_2)]$$

$$\geq 0$$

$R_l - R_s \geq 0 \implies$ it is optimal to select $l_1$ first.

**Case 2:** When $(1-p)l_1 < ps_2 + (1-p)(s_2 + s_3)$

$$R_s - R_l = ps_1 + (1-p)(s_1 + s_2) - (1-p)l_1 + p[V_2(1, 2) - V_2(1, 1)] + (1-p)[V_2(1, 3) - V_2(2, 1)]$$

$$\geq ps_1 + (1-p)(s_1 + s_2) - (1-p)l_1 + p[(1-p)l_1 - ps_1 - (1-p)(s_1 + s_2)]$$

$$+ (1-p)[(1-p)l_1 - ps_1 - (1-p)(s_1 + s_2)]$$

$$(\because V_2(1, 2) \geq (1-p)l_1)$$

$$\geq (1-p)[ps_1 + (1-p)(s_1 + s_2) - (1-p)l_1]$$

$$\geq 0$$
$R_s - R_l \geq 0 \implies$ it is optimal to select $(s_1, s_2)$ first.

**Case 3:** When $p s_1 + (1-p)(s_1 + s_2) > (1-p)l_1 \geq p s_2 + (1-p)(s_2 + s_3)$.

\[ R_l - R_s = (1-p)l_1 - ps_1 - (1-p)(s_1 + s_2) + p[V_2(1,1) - V_2(1,2)] + (1-p)[V_2(2,1) - V_2(1,3)] = (1-p)l_1 - ps_1 - (1-p)(s_1 + s_2) + p[ps_1 + (1-p)(s_1 + s_2) - (1-p)l_1] + (1-p)[ps_1 + (1-p)(s_1 + s_2) - (1-p)l_1] = 0 \]

$R_l = R_s \implies$ we are indifferent between selecting $l_1$ and $(s_1, s_2)$. \qed

The Optimal Policy shows that there is a region where selecting $l_1$ is strongly preferable, a region where selecting $(s_1, s_2)$ is strongly preferable, and a region where we are indifferent between selecting $l_1$ and $(s_1, s_2)$. We shall look into the multiple break case before discussing the implications of such a partition.

**Multiple breaks**

**Theorem 3.5.** When $k = n$, and the break size is unknown, the Optimal Policy is to:

1. select $l_1$ if $(1-p)l_1 \geq ps_1 + (1-p)(s_1 + s_2)$
2. select $(s_1, s_2)$ if $(1-p)l_1 < ps_n + (1-p)(s_n + s_{n+1})$
3. select either $l_1$ or $(s_1, s_2)$ otherwise

**Proof.** Proof is given in Appendix A \qed

Theorem 3.5 extends Lemma 3.9. There are two regions where the scheduler has a strong preference over the possible choices, separated by a region of indifference.
In the first region (Case 1), \( l_1 \) is strongly preferable based on a myopic comparison of the expected revenues earned from selecting \( l_1 \) and \((s_1,s_2)\). Since \( s_1 \) and \( s_2 \) are the most profitable small ads, \( l_1 \) earns a higher expected revenue than any combination of small ads to be selected.

The same can be said of the third region (Case 2); since expected revenue from selecting \( l_1 \) is lower than that from selecting \((s_n,s_{n+1})\), we select the smaller ads first. If we are faced with a series of only small breaks, we schedule \((s_1,\ldots,s_n)\) through the match, earning more than if we had scheduled \( l_1 \) first. If a large break were to arrive subsequently, the strategy would check if Case 1 applies, earning at least as much as the Greedy Policy.

The indifference exists because when breaks can only be of size \( S \) or \( L \), and \((1-p)l_1 \geq s_n + (1-p)s_{n+1}\), then the Optimal Policy expects to schedule \( l_1 \) when \((1-p)l_1 \) is greater than some \((s_j + (1-p)s_{j+1})\), where \( j > 1 \), and Case 1 applies.

In the worst case, suppose \((1-p)l_1 = s_n + (1-p)s_{n+1}\), and small ads have been scheduled for the first \( n-1 \) breaks, which are found to be short. For the \( n^{th} \) break, expected revenue from selecting \( l_1 \) will be compared to expected revenue from scheduling \((s_n,s_{n+1})\), and Case 1 applies. Therefore \( l_1 \) is guaranteed to be selected for some break in the match.

Similarly, since selecting \((s_1,s_2)\) earns higher expected revenue than any large ad, then in the worst case, \( l_1 \) is repeatedly selected for the first \( n-1 \) breaks which turn out to be short. Then for the \( n^{th} \) break, Case 2 will apply and \((s_1,s_2)\) will be selected. Thus \((s_1,s_2)\) is also guaranteed to be selected for some break in the match.

Therefore we are indifferent between selecting \( l_1 \) and \((s_1,s_2)\) for \( b_1 \).

From a managerial perspective, the region of indifference gives the network flexibility when accepting orders with client constraints at the start of the match. Consider a scenario where a client wishes the network to schedule a large ad when a particular wicket falls (and breaks in the match can only be of size \( S \) or \( L \)). The network can accept this scheduling constraint on
a long ad as long as the expected value from airing the ad is higher than the expected value of airing two small ads in their inventory which fall in the \( n^{th} \) and \((n+1)^{th}\) position, where \( n \) is the total number of breaks in the match. It is possible, however, that the above condition is satisfied, but the ad is not the most valuable large ad in the inventory when the break occurs. The network can still use the Optimal Policy and the break size distribution to quote a value to the client to make the ad a viable candidate for that break. The exact methodology is beyond the scope of our current discussion, however pricing policies based on the Optimal Policy is an area for future research.

### 3.3.2 Stochastic Number of Breaks

In this subsection we consider an extension where the number of breaks is stochastic. As in Section 3.2.2, we introduce a probability \( p_0 \) of having a break of size 0, while we assume that break could be of size \( S \) with probability \( p_1 \), and of size \( L = 2S \) with probability \( p_2 \). We do not consider breaks of size greater than \( L \).

As before, we assume that a break of size zero might arrive at any point in the match, and that we are always aware of the arrival of such a break, and discount the number of breaks remaining accordingly. All other assumptions and notations as listed in Section 3.1 still remain.

**One break**

With only one break possible, the scheduler chooses greedily between the expected values of selecting \( l_1 \) and \((s_1,s_2)\). If the last break turns out to be of size zero, either strategy earns zero; if it is of size \( S \), we earn zero with \( l_1 \) and \( s_1 \) with \((s_1,s_2)\); if it is of size \( L \) we earn \( l_1 \) and \( s_1 + s_2 \) respectively.

As before, we propose a lemma without proof for the one break case:
Lemma 3.10. When \( k = 1, p_0 \geq 0, \) and break sizes are unknown, the Optimal Policy is to:

1. select \( l_1 \) if \( p_2 l_1 \geq p_1 s_1 + p_2 (s_1 + s_2) \)
2. select \((s_1, s_2)\) otherwise

Two breaks

For the two break case, the strategy continues to be a myopic choice between the expected revenues earned from selecting \( l_1 \) and \((s_1, s_2)\), despite the introduction of breaks of size zero.

Lemma 3.11. When \( k = 2, p_0 \geq 0, \) and break sizes are unknown, the Optimal Policy is to:

1. select \( l_1 \) if \( p_2 l_1 \geq p_1 s_1 + p_2 (s_1 + s_2) \)
2. select \((s_1, s_2)\) otherwise

Proof. Let \( p_2 l_1 = t + p_1 s_1 + p_2 (s_1 + s_2) \). We use \( R_l, R_s \), and \( V_i(l, s) \) as defined in the previous subsections. From Lemma 3.10, the rule holds good for one break.

With the introduction of \( p_0 \), expected revenues earned are as follows:

\[
R_l = (p_0 + p_1)V_1(1, 1) + p_2(l_1 + V_1(2, 1)) \\
R_s = p_0 V_1(1, 1) + p_1(s_1 + V_1(1, 2)) + p_2(s_1 + s_2 + V_1(1, 3))
\]

Case 1: When \((1 - p)l_1 \geq ps_1 + (1 - p)(s_1 + s_2)\),

\[
R_l - R_s = (p_0 + p_1)V_1(1, 1) + p_2(l_1 + V_1(2, 1)) \\
- p_0 V_1(1, 1) - p_1(s_1 + V_1(1, 2)) - p_2(s_1 + s_2 + V_1(1, 3)) \\
= p_1(V_1(1, 1) - s_1 - V_1(1, 2)) + p_2(l_1 + V_1(2, 1) - (s_1 + s_2) - V_1(1, 3)) \\
= t + p_1[V_1(1, 1) - V_1(1, 2)] + p_2[V_1(2, 1) - V_1(1, 3)]
\]
Thus, $R_l - R_s \geq 0 \implies$ it is optimal to selection $l_1$ first.

**Case 2:** When $(1 - p)l_1 < ps_1 + (1 - p)(s_1 + s_2)$, let $t + p_2 l_1 = p_1 s_1 + p_2 (s_1 + s_2)$, where $t > 0$.

Proceeding along similar lines as **Case 1**, we can prove that $R_s - R_l \geq p_0 t \geq 0$.

**Case 1** and **Case 2** correspond with the cases in Lemma 3.9. The addition of $p_0$ does not affect the two regions of strong preference because the expected revenue when the break is of size zero is the same whether we selection $l_1$ or $(s_1, s_2)$. Therefore, our decision is solely based on $p_1$ and $p_2$, and the expected revenues earned thereby.

There is no **Case 3** corresponding to Lemma 3.8 because $S$ and $L$ are not the only break sizes possible. When the expected revenue from selecting $l_1$ is less than that from selecting $(s_1, s_2)$, **Case 2** applies. From the proof for Lemma 3.11, the difference between the two expected revenues is at least $p_0 t$, where $t \geq 0$ is the difference in expected revenue. Since $p_0 > 0$, we are indifferent between scheduling $l_1$ and $(s_1, s_2)$ only when $t = 0$; i.e. $p_2 l_1 = p_1 s_1 + p_2 (s_1 + s_2)$.

**Multiple breaks**

From Lemmas 3.10 and 3.11, we can see that the myopic rule holds good when we plan for either one break or two. In this section, we use the previous results and prove by induction that the rule holds good for any number of breaks remaining.
Theorem 3.6. When \( k = n, \ p_0 \geq 0, \) and break sizes are unknown, the Optimal Policy is to:

1. select \( l_1 \) if \( p_2 l_1 \geq p_1 s_1 + p_2 (s_1 + s_2) \)
2. select \((s_1, s_2)\) otherwise

Proof. Proof is given in Appendix A.

Theorem 3.6 can be seen as a special case of Theorem 3.5 where \( p_0 > 0. \) As before, we have two regions where the scheduler has strong preferences, but there is no region of indifference (except for the point where expected revenue from selecting \( l_1 \) equals that from selecting \((s_1, s_2)\)). Therefore, when the number of breaks is not fixed and break sizes are not known in advance, the scheduler is forced to select between \( l_1 \) and \((s_1, s_2)\) myopically based on the expected revenues earned.

3.3.3 Multiple Break Sizes and its variants

In this section, we consider the case where breaks can be of size \((S, 2S, 3S, ..., MS)\), and the size of the break is not known in advance.

As shown in Theorem 3.5 when breaks are of sizes \( S \) or \( 2S \), the Optimal Policy when break sizes are unknown is based on a myopic comparison of the expected revenues earned. When the maximum break size is \( 2S \), we choose between \( l_1 \) and \((s_1, s_2)\). This policy is independent of the number of breaks remaining, since breaks are IID and break size is unknown for each break.

When the maximum break size is \( MS \), the Optimal Policy should provide the permutation of ads to be scheduled based on the break size distribution. Consider, for example, the case when \( M = 3 \). Let us assume that the probability of breaks of sizes \( S, 2S \) and \( 3S \) occurring is \( p_1, p_2 \) and \( p_3 \), where \( p_1 + p_2 + p_3 = 1 \). Then the ad schedules that can be generated and the revenues earned are shown in Table 3.2.
3.3 Unknown Break Size

<table>
<thead>
<tr>
<th>Schedule</th>
<th>Revenue earned</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1, s_2, s_3$</td>
<td>$s_1 + (p_2 + p_3)s_2 + p_3s_3$</td>
</tr>
<tr>
<td>$s_1, l_1$</td>
<td>$s_1 + p_3l_1$</td>
</tr>
<tr>
<td>$l_1, s_1$</td>
<td>$(p_2 + p_3)l_1 + p_3s_1$</td>
</tr>
</tbody>
</table>

Table 3.2: Revenues earned with each possible schedule

As shown, the revenue earned with each schedule is different, and the Optimal Policy should select the schedule which generates the maximum revenue based on the probabilities $p_i$ and the values of the ads scheduled.

It can be shown that the number of possible combinations of ads for each value of $M$ is a Fibonacci sequence as shown in table 3.3.

<table>
<thead>
<tr>
<th>$M$</th>
<th>Number of ad schedules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
</tbody>
</table>

Table 3.3: Number of possible ad schedules for each value of $M$

This can be explained as follows: let us assume that each break is divided into ‘slots’ of length $S$, thus when $M = n$, there are $n$ slots available to be scheduled. Let $\sigma_n$ and $\sigma_{n+1}$ be the number of ad schedules possible when $M = n$ and $M = n + 1$ respectively. Then for $M = n + 2$, an additional slot is added at the end of $n + 1$ slots, and this slot can either be programmed with an ad of size $S$ or an ad of size $L = 2S$, starting from slot $n + 1$. If an ad of size $S$ is scheduled in slot $n + 2$, the previous $n + 1$ slots can be scheduled in $\sigma_{n+1}$ ways. If, however, an ad of size $L$ is scheduled across slots $n + 1$ and $n + 2$, then the previous $n$ slots can be scheduled in $\sigma_n$ ways, thus giving $\sigma_{n+2} = \sigma_n + \sigma_{n+1}$.

By the well known Binet’s formula\(^1\), when $M = n$, the number of possible

\(^1\)Theory of Binet formulas for Fibonacci and Lucas $p$-numbers[25]
Optimal Policy

Combinations $\Phi(n)$ is

$$\Phi(n) = \frac{\varphi^n - (1 - \varphi)^n}{\sqrt{5}} = \frac{\varphi^n - (-1/\varphi)^n}{\sqrt{5}}$$

Where $\varphi = \frac{1 + \sqrt{5}}{2} \approx 1.6180339887\ldots$

The number of ad schedules that can be programmed increases exponentially with the value of $M$, resulting in a 'Hughes effect', or a 'curse of dimensionality'\(^2\), a classical problem that arises when dealing with problems of stochastic recourse. For any $M = n$, the Optimal Policy would be the maximum of the expected revenues earned from each of the $\Phi(n)$ schedules. As discussed in Section 2.3, the problem is better solved with specialized heuristics, rather than attempting an analytical solution for the general $n$-case.

\(^2\)Dynamic programming[2]
Chapter 4

Numerical Analysis

In this chapter we perform a numerical analysis of the Optimal Policy discussed in Chapter 3. We compare the Optimal Policy to the Greedy Policy, in order to find the conditions under which the Optimal Policy most outperforms the Greedy Policy. We also study the impact of service level commitments and the impact of uncertainty on the performance of the Optimal Policy.

4.1 Performance with deterministic number of breaks

We begin with cases where the number of breaks is fixed, and there are no breaks of size zero. A study of how the Optimal Policy performs as parameters change will give us an idea of how stable it is, and allow us to find conditions where it is most beneficial to use the Optimal Policy.

We begin with listing the parameters we will use for the study, and subsequently study the impact of various parameters on revenues and service levels.
4.1.1 Parameters

To evaluate the performance of the policies, we generate the values for the large and small ads based on the parameters listed in Table 4.1.

Results were averaged over 500 runs, and at each iteration the inventory and spot values were changed. This was done to ensure that we tested the strategies across a broad range of data sets, and that the results were representative of average scenarios.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of breaks</td>
<td>50</td>
</tr>
<tr>
<td>Break size (s)</td>
<td>10 sec or 20 sec</td>
</tr>
<tr>
<td>Number of large ads</td>
<td>50</td>
</tr>
<tr>
<td>Number of small ads</td>
<td>100</td>
</tr>
<tr>
<td>Large ads values ($)</td>
<td>1000 – 1200</td>
</tr>
<tr>
<td>Small ads values ($)</td>
<td>200 – 1000</td>
</tr>
</tbody>
</table>

Table 4.1: Parameters for numerical analysis of Optimal Policy and Greedy Policy

In Table 4.1, the number of breaks, 50, is typical of the number of breaks found in a T-20 match, where each of two innings has twenty overs breaks, and ten wickets are expected to fall during the course of the match. We assume that breaks can be of sizes 10 seconds or 20 seconds only with equal probability, hence the mean break length is 15 seconds. The range of values earned by long ads and short ads are typical of orders received by major sports broadcasters for international T-20 tournaments, and we select random values within these ranges.

We next look at performances of the Optimal Policy and Greedy Policy when air time sold (i.e., number of ads available in the inventory) varies.

4.1.2 Impact of variation in air time sold

We analyze how change in the amount of air time sold affects the performance of the Optimal Policy and the Greedy Policy. We start with 50 large ads and 100 short ads, as listed in Table 4.1, and remove two short ads for every large
ad removed from the inventory, to keep the ratio of air time between large and small ads constant. Air time sold ranged from half the expected air time over the course of a match, to more than twice.

Cricket broadcasters oversell air time for important tournaments, particularly those that involve India, and expect to make good the ads not shown during live broadcast in non-live segments later. By overselling, networks have more flexibility in what ads they show, and are able to cash in on opportunities when actual break time in matches exceeds the expected break time significantly.

Underselling air time is usually done when the penalty of not achieving service levels is severe, or the network expects a curtailed match, for instance, due to rain. By decreasing the air time sold, networks often aim to provide higher service levels, while putting a premium on the spot value per second. We observe changes in service levels of large and small ads, along with change in revenues earned for each policy as the amount of air time sold is varied.

The result of the numerical analysis when lengths of breaks and their number are known at the time of scheduling is summarized in Table 4.2.

When air time sold is close to the expected air time available (∼750s), we see that the Optimal Policy outperforms the Greedy Policy by almost 3%. This translates to an average of $1262 per match, equivalent to the expected value earned by airing two small ads more per match than the Greedy Policy.

Service levels for the Optimal Policy and the Greedy Policy show that the Optimal Policy consistently schedules more large ads than then Greedy Policy does, whereas the Greedy Policy relies more on small ads. As a consequence, we see that when the air time sold is 800s, approximately 97% of the small ads have been shown by the Greedy Policy, yet almost one small break in a 50 break match has nothing scheduled in it. This is because the Greedy Policy schedules small ads even for large breaks early in the match and runs out of ads to schedule when small breaks occur.

This is an important result for broadcast networks. The over reliance of
the Greedy Policy on small ads to earn revenue may lead to lost opportunities, whereas the Optimal Policy schedules large ads whenever possible and holds a reserve of small ads for small breaks, leading to improved service levels and revenues overall.

![Figure 4.1: Average total revenue versus air time sold](image)

We plot the revenues earned by the Optimal Policy and the Greedy Policy against air time sold in Figure 4.1. The difference in revenues earned is pronounced when the service level is between 80% and 90%, which corresponds to airtime sold of around 800 seconds (from Table 4.2), which is roughly equal to the expected air time. The service level mark of 80% is significant, since this is the service level usually promised by the network to advertisers during most tournaments. Thus the Optimal Policy is significantly better than the Greedy Policy in conditions that approach real world situations.

We next consider the case where the number of breaks in a match is not fixed. The maximum difference between the Optimal Policy and the Greedy Policy occurs when the time sold is 800 seconds, which corresponds to 40 small ads and 10 large ads sold. Since the number of breaks is 50, the Greedy Policy schedules the small ads in the large breaks and is left with almost one small break left unscheduled (Table 4.2). When airtime sold is higher, both policies have a greater choice of ads to choose from and the
difference between policies reduces; and when the airtime sold is lower, both policies suffer from a lack of ads equally.

The efficient frontier helps managers determine what service level is most optimal. Promising lower service levels can yield higher revenues, which should be balanced, however, with the possible loss of goodwill. Managers can thus decide on a target service level by considering both the benefits and costs involved.

### 4.1.3 Value of flexibility

Small ads can be shown in both small and large breaks, whereas large ads can only be shown in large breaks. Thus, small ads offer more flexibility to the broadcaster as to which break they can be shown in. To analyze the value of flexibility, we study the change in revenue as we change the mix of small and large ads in the inventory. We begin with the parameters as outlined in Table 4.1, and split the large ads randomly into two, thus creating equally valued small ads, and increasing the ratio of small ads to large ads in the inventory.

We begin with a numerical analysis when the number of breaks per match is fixed. Table 4.3 shows the average revenue earned per match by the Optimal Policy and the Greedy Policy as the mix of large and small ads is varied and the number of breaks is fixed.

The percentage change in revenue for each policy is the amount of change that each policy earns in comparison to the case when the ratio of air time is equally split between small ads and large ads, which is our starting ratio. The last column lists the percentage difference in revenues between the Greedy Policy and the Optimal Policy. We see that as the ratio of small ads to large ads increases, the difference between the two policies decreases. We plot the percentage change of the Optimal Policy and the Greedy Policy in Figure 4.2. The Optimal Policy does not vary much from its original value, whereas the Greedy Policy displays an increase of almost 2.5% from its original.
The increase in the revenue earned by the Greedy Policy is explained by the preference of the Greedy Policy for small ads: as discussed in section 4.1.2, the Greedy Policy schedules more small ads than long ads, therefore revenues increase when the number of small ads increase.

From a managerial perspective, the ability of the Optimal Policy to maintain revenue earned despite substantial change in the ratio of small and large ads is of importance. Thus the Optimal Policy is a robust strategy despite changing inventory mix, and ensures the network a stable revenue regardless of the inventory composition.

Figure 4.3 plots the difference between the Optimal Policy and the Greedy Policy. We see that the difference is most significant when the air time is equally divided between the large ads and the small ads, and this difference decreases as the proportion of small ads increases. As expected, when the inventory consists only of small ads, there is no difference in revenue earned between the Optimal Policy and the Greedy Policy.

### 4.1.4 Impact of Variability

We next investigate the impact of variability on the performance of the Optimal Policy, as defined in Theorem 3.3. The distribution of break lengths
Figure 4.3: Difference between Greedy and Optimal with split of large ads and the size of our inventory are listed in Table 4.4, and all other parameters are the same as listed in Table 4.1.

To simulate variability, we use a Uniform Distribution with a mean of 60 seconds. Although the break lengths were generated from a Uniform Distribution, they are rounded down to the nearest multiple of 10 seconds, which is the size of the small ad. Large ads are of size 20 seconds, as before. We vary the support for the break size distribution from a constant 60 to [10, 110]. This corresponds to varying the standard deviation between \([0, \frac{50}{\sqrt{3}}]\), and the range between \([0, 100]\). As before, however, break sizes that are not a multiple of 10 are rounded down to the nearest multiple of 10, since the remaining break time will remain unutilized. For simplicity, we will only consider the range when discussing variability. Finally, we note that the expected air time has increased substantially, necessitating an increase in the size of our inventory as shown in Table 4.4.

The results have been tabulated in Table 4.5. We find that both the Optimal Policy and the Greedy Policy are not affected significantly by increase in variability. The ‘percentage change’ row for each policy shows the change in revenues earned compared to zero variability case. In this case, we see that there is negligible change in the revenues earned by the Optimal Policy and
the Greedy Policy as variability increases. Service levels also do not show significant changes with variability, thus supporting our inference that the effect of variability on revenues earned is negligible. Figure 4.4 shows the percentage change in revenues with increase in range.

![Figure 4.4: Change in revenue with variability](image)

From a managerial perspective, as long as the expectation of break lengths is steady, increased variability does not significantly affect revenues. As variability increases, breaks of large sizes are complemented by breaks of small sizes, keeping the overall revenues earned from fluctuating. The order in which large breaks and small breaks occur causes the small gains for the Optimal Policy over the Greedy Policy. The Greedy Policy has a preference for scheduling small ads in the large breaks, so when matches have a sequence of large breaks followed by small ones, the Greedy Policy is left with less valuable small ads for the small breaks. In contrast, the Optimal Policy has a lower threshold for scheduling large ads as discussed in section 3.2.3, therefore it is able to gain more from the short breaks in the latter part of the match.

In this chapter, we studied the behavior of the Optimal Policy numerically, and derived insights from the results. In the next chapter, we will discuss specialized algorithms and heuristics that will attempt to solve harder problems, such as having ads of multiple lengths with diversity constraints.
### Table 4.2: Revenues and Service levels with change in air time sold

(L: large, S: small, OP: Optimal Policy, GP: Greedy Policy)

<table>
<thead>
<tr>
<th>Air time sold (sec)</th>
<th>2000</th>
<th>1600</th>
<th>1200</th>
<th>800</th>
<th>400</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Revenue</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>54,935.46</td>
<td>52,349.08</td>
<td>49,523.60</td>
<td>44,489.78</td>
<td>23,034.12</td>
</tr>
<tr>
<td>GP</td>
<td>54,733.86</td>
<td>51,927.26</td>
<td>48,673.94</td>
<td>43,227.58</td>
<td>23,034.12</td>
</tr>
<tr>
<td>% Difference</td>
<td>0.37</td>
<td>0.81</td>
<td>1.72</td>
<td>2.84</td>
<td>0.00</td>
</tr>
<tr>
<td><strong>Service Levels (%)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L ads</td>
<td>22.52</td>
<td>39.26</td>
<td>67.15</td>
<td>99.44</td>
<td>100.00</td>
</tr>
<tr>
<td>S ads</td>
<td>52.37</td>
<td>54.42</td>
<td>57.92</td>
<td>87.34</td>
<td>100.00</td>
</tr>
<tr>
<td>GP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L ads</td>
<td>16.50</td>
<td>29.31</td>
<td>50.35</td>
<td>88.11</td>
<td>100.00</td>
</tr>
<tr>
<td>S ads</td>
<td>58.40</td>
<td>64.37</td>
<td>74.72</td>
<td>96.62</td>
<td>100.00</td>
</tr>
<tr>
<td><strong>Unused Breaks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L breaks</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
<td>14.64</td>
</tr>
<tr>
<td>S breaks</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.01</td>
<td>5.39</td>
</tr>
<tr>
<td>GP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L breaks</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.02</td>
<td>10.70</td>
</tr>
<tr>
<td>S breaks</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.84</td>
<td>13.15</td>
</tr>
<tr>
<td>Long ads split (%)</td>
<td>Optimal Policy</td>
<td>Greedy Policy</td>
<td>Difference (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------</td>
<td>--------------</td>
<td>---------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average Value</td>
<td>% change</td>
<td>Average Value</td>
<td>% change</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>48,304</td>
<td>0</td>
<td>47,132</td>
<td>0</td>
<td>2.43</td>
</tr>
<tr>
<td>10</td>
<td>48,358</td>
<td>+0.11</td>
<td>47,965</td>
<td>+1.77</td>
<td>0.81</td>
</tr>
<tr>
<td>20</td>
<td>48,340</td>
<td>+0.08</td>
<td>48,195</td>
<td>+2.26</td>
<td>0.30</td>
</tr>
<tr>
<td>30</td>
<td>48,248</td>
<td>-0.17</td>
<td>48,215</td>
<td>+2.30</td>
<td>0.07</td>
</tr>
<tr>
<td>40</td>
<td>48,199</td>
<td>-0.22</td>
<td>48,183</td>
<td>+2.23</td>
<td>0.03</td>
</tr>
<tr>
<td>50</td>
<td>48,267</td>
<td>-0.08</td>
<td>48,261</td>
<td>+2.40</td>
<td>0.01</td>
</tr>
<tr>
<td>60</td>
<td>48,347</td>
<td>+0.09</td>
<td>48,343</td>
<td>+2.57</td>
<td>0.01</td>
</tr>
<tr>
<td>70</td>
<td>48,275</td>
<td>-0.06</td>
<td>48,274</td>
<td>+2.42</td>
<td>0</td>
</tr>
<tr>
<td>80</td>
<td>48,369</td>
<td>+0.14</td>
<td>48,369</td>
<td>+2.62</td>
<td>0</td>
</tr>
<tr>
<td>90</td>
<td>48,229</td>
<td>-0.15</td>
<td>48,229</td>
<td>+2.33</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>48,306</td>
<td>+0.00</td>
<td>48,306</td>
<td>+2.49</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.3: Value of Flexibility

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break lengths</td>
<td>$\sim \mathcal{U} (\mu - \delta, \mu + \delta)$</td>
</tr>
<tr>
<td></td>
<td>$\mu = 60, \delta \in [0, 50]$</td>
</tr>
<tr>
<td>Number of large ads</td>
<td>100</td>
</tr>
<tr>
<td>Number of small ads</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 4.4: Parameters for analyzing impact of variability on Optimal Policy
4.1 Performance with deterministic number of breaks

<table>
<thead>
<tr>
<th>Range</th>
<th>0</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP</td>
<td>191,411</td>
<td>191,335</td>
<td>191,184</td>
<td>191,420</td>
<td>191,564</td>
<td>191,404</td>
</tr>
<tr>
<td>% change</td>
<td>0.00</td>
<td>-0.04</td>
<td>-0.12</td>
<td>+0.01</td>
<td>+0.08</td>
<td>-0.003</td>
</tr>
<tr>
<td>GP</td>
<td>191,411</td>
<td>191,313</td>
<td>191,180</td>
<td>191,408</td>
<td>191,559</td>
<td>191,396</td>
</tr>
<tr>
<td>% change</td>
<td>0.00</td>
<td>-0.05</td>
<td>-0.12</td>
<td>-0.001</td>
<td>+0.08</td>
<td>-0.01</td>
</tr>
</tbody>
</table>

Table 4.5: Impact of variability

(L: large, S: small, OP: Optimal Policy, GP: Greedy Policy)
Chapter 5

Applications in Practice:
Scheduling ads for Cricket

In previous chapters, we have studied the Optimal Policy for a stylized model of the real world problem. The motivation for this research came from our discussions with a major cricket broadcaster, who also provided us with real-world data, based on which we generated parameters for numerical analysis. We now test several scheduling heuristics under more constraints and present the analysis of the data, a study of the heuristics tested and the results obtained, and create relevant managerial insights.

5.1 Data Description

In order to have an estimate of the parameters and constraints in which ad scheduling was done, we received production logs of ads aired and the spot price for each of those ads during a T-20 tournament.

5.1.1 Break Lengths

Our findings are shown in Table 5.1, and the break length distribution is shown in Fig 5.1. The breaks recorded here were measured between the end
of an over and the start of the next one, creating a slight skew towards the right (since actual time available to show ads is shorter than the time between two such overs).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average break length</td>
<td>68.4</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>23</td>
</tr>
<tr>
<td>Minimum break length</td>
<td>10</td>
</tr>
<tr>
<td>Maximum break length</td>
<td>170</td>
</tr>
<tr>
<td>Number of samples (breaks)</td>
<td>983</td>
</tr>
</tbody>
</table>

Table 5.1: Analysis of break length data

For our numerical simulation, we estimated the break length to have a mean of 40 seconds and to be uniformly distributed between 10 seconds and 70 seconds.

For our study of T-20 matches, we estimated a total of 50 breaks per match (40 breaks in between overs and 10 wicket breaks). As a simplification of match conditions, we set the number of breaks to be 50 while estimating the performance of our heuristic.
5.1.2 Ad Lengths, Service Levels and Demands

The lengths of the ads contracted from clients was given directly by the sports broadcaster, and a summary of the data is shown in Table 5.2. For

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average ad length</td>
<td>20.82</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>7.84</td>
</tr>
<tr>
<td>Minimum ad length</td>
<td>10</td>
</tr>
<tr>
<td>Maximum ad length</td>
<td>40</td>
</tr>
<tr>
<td>Number of samples (ads)</td>
<td>2393</td>
</tr>
</tbody>
</table>

Table 5.2: Analysis of ad length data

our analysis, we consider ad lengths of 10 seconds, 20 seconds, and 30 seconds. This agrees closely with the data from the broadcaster. On average, ads from 20 advertisers were shown in each match, and their demands were as shown in Table 5.3:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Demand</td>
<td>83.82</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>9.84</td>
</tr>
<tr>
<td>Minimum Demand</td>
<td>20</td>
</tr>
<tr>
<td>Maximum Demand</td>
<td>180</td>
</tr>
<tr>
<td>Number of samples (matches)</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 5.3: Analysis of Client Demands

In our discussions with the broadcaster, we found that the broadcaster had to satisfy service level commitments of between 75% to 80% of the total demand of each advertiser (i.e. 75%-80% of the demand had to be successfully aired). These commitments could be satisfied across the duration of the tournament, but for our study we limit our service level commitments to each game. For our simulation, we estimate demands from advertisers such that we can satisfy the service levels promised to most, if not all advertisers. To achieve this, we use ‘penalties’ in the model, so that unsatisfied demand
below the promised service level decrease the profits earned. The IP model with service level guarantee is given in Section 5.3.1.

### 5.1.3 Spot Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Rev/Sec</td>
<td>4110</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>583</td>
</tr>
<tr>
<td>Minimum Rev/Sec</td>
<td>3714.28</td>
</tr>
<tr>
<td>Maximum Rev/Sec</td>
<td>5714.29</td>
</tr>
<tr>
<td>Number of samples</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5.4: Analysis of spot values

Spot values in the tournament we analyzed were linear with ad length. This made it easier to characterize the value of each advertiser in terms of how much revenue per second each of his ads earned, so broadcaster concerns such as giving higher value to ‘sponsors’ of a tournament could be incorporated by adding value to the revenue per second that that advertiser earns. For our simulation, we draw random values from the range [3500, 5500]. The number of advertisers per match varied in the tournament, with an average of 20 advertisers per match.

From the data collected, we could characterize the advertisers, their demands, and the break lengths for each match and run simulations that closely reflected real world requirements. We discuss the heuristics considered for the simulation below.

### 5.2 Assumptions

To simplify the models we examine, we assume the following:

1. Spot prices for ads are linear in ad length. This is supported by the data we received from the sports broadcaster (see Section 5.1.3).
2. Two ads from the same advertiser cannot be shown in the same break; but there is no restraint on showing two ads from one advertiser in subsequent breaks.

In later sections, we will add an assumption that service level commitments must be met, and the broadcaster pay a penalty if he doesn’t meet those constraints.

5.3 Knapsack Model

In this section we study the basic Knapsack model and a variation with service level guarantees. The Knapsack model aims to fit the best possible combination of ads into a break. The size of the knapsack is taken as the expected length of the break. Similar to Witchakul et al [28] we consider penalties for crashes and underutilization, but unlike them, we build a model for multiple periods, where ad inventory changes from break to break.

The model is as follows:

$$\text{max } z = \sum_{n} \sum_{a} \sum_{l} r_{al} \cdot x_{aln}$$

subject to:

$$\sum_{l} x_{aln} \leq 1 \, \forall a, n \quad (5.1)$$

$$\sum_{a} \sum_{l} (l \cdot x_{aln}) \leq b \, \forall n \quad (5.2)$$

$$\sum_{n} x_{aln} \leq N_{al} \, \forall a, l \quad (5.3)$$

$$x_{aln} = 0 \text{ or } 1 \quad (5.4)$$

where

$a$ is the index of advertisers,

$l$ is the length of each ad
5.3 Knapsack Model

\( n \) is the break sequence number

\( b \) is the expected / predicted break length

\( N_{al} \) is the number of ads of length \( l \) contracted from advertiser \( a \)

\( x_{aln} \) is the decision variable

**Constraint (5.1)** restricts the number of times an advertiser’s ad can be shown in a break;

**Constraint (5.2)** specifies that the sum of all ads per break should be lower than the break length expected;

**Constraint (5.3)** ensures that we only show as many ads as we have a contract for;

**Constraint (5.4)** makes this model a binary integer programming model

### 5.3.1 Knapsack model with service guarantee

Sports broadcasters, ensure service levels (usually of around 80%) to their clients, and usually have to make good or pay a penalty when the service level commitments are not met. We include this constraint to the earlier Knapsack model discussed in section 5.3:

\[
\begin{align*}
\text{max } z &= \sum_{n} \sum_{a} \sum_{l} r_{al} \times x_{aln} - P \sum_{a} s_{a} \\
\text{subject to:} \quad &\sum_{l} x_{aln} \leq 1, \forall a, n \quad (5.5) \\
&\sum_{a} \sum_{l} (l \times x_{aln}) \leq b, \forall n \quad (5.6) \\
&\sum_{n} x_{aln} \leq N_{al}, \forall a, l \quad (5.7) \\
&\sum_{n}^{n+1} x_{aln} \leq 1, \forall a, l, n \quad (5.8)
\end{align*}
\]
\[ \sum_l \sum_n (l \cdot x_{aln}) + s_a \geq S \sum_l (l \cdot N_{al}), \forall a \]  \hspace{1cm} (5.9)

\[ x_{aln} = 0 \text{ or } 1 \]  \hspace{1cm} (5.10)

\[ s_a \geq 0 \]  \hspace{1cm} (5.11)

where

\( P \) is the penalty for not meeting the promised service level

\( s_a \) is the duration by which the service level was not met

\( S \) is the promised service level

and all other variables have the same meaning as before (see section 5.3).

**Constraint (5.9)** ensures that the service level guarantees, if not met, are penalized in the Objective.

### 5.4 Heuristics

In this section we describe a few heuristics that we evaluate for recommendation to the sports broadcaster. These heuristics work on data as outlined in section 5.1. Finally, we compare the revenues earned by each approach with the revenue earned in the perfect information case (PI), where the lengths of all breaks are known in advance of the schedule generation, which is the theoretical upper bound.

#### 5.4.1 Greedy Policy

The Greedy Policy assumes that break lengths are known in advance, and then uses the Knapsack model for each (known) break without planning for subsequent breaks. We have compared a simplified version of the Greedy Policy to the Optimal Policy in Chapter 3, and we will extend that study and compare the Greedy Policy with other heuristics.
5.4.2 Certainty Equivalent Heuristic

The Certainty Equivalent heuristic (CE) builds a schedule of ads based on the expected break length (based on the knapsack model with service level guarantee outlined in section 5.3.1). Having generated the breaks (which are all of length equal to the mean break length), we schedule them against the actual breaks and find out how it performs. This gives us a lower bound on how any variation of the certainty equivalent heuristic should perform.

5.4.3 Dynamic Certainty Equivalent Heuristic

For Dynamic Certainty Equivalent (DCE), we generate ‘bundles’ to fit an expected distribution of break lengths and that satisfies all constraints. The IP is as outlined in section 5.3.1. The scheduler, who knows the length of the break, schedules the bundle that best matches the break size. If there are multiple bundles of equal size that fit in the break, the scheduler chooses the first one among them.

5.4.4 Dynamic Modified Certainty Equivalent Heuristic

The Dynamic Modified Certainty Equivalent Heuristic (DMCE) is a variation of DCE, we now generate bundles not only at the beginning of the match but also periodically during of the match. The periods of bundle generation could be varied: bundles could be generated at specific points during the match, or when a break occurs with no perfectly matching bundles at hand. We study both cases and report insights.

5.4.5 Perfect Information

The Perfect Information heuristic (PI) is the theoretical ‘upper bound’, so we can compare the performance of heuristics as a percentage the maximum
revenue attainable. We assume that sizes of all breaks are known before the first break, and run a knapsack that schedules ads with the given constraints in all breaks.

5.5 Comparative Statics with Service Constraints

In this section, we study results of numerical analysis done based on the heuristics proposed in Section 5.4, under service constraints. The main aim of this study was to find out which of the heuristics was most promising, and to be able to suggest the most promising direction in which the sports broadcaster may direct their efforts to maximize the revenue in real world situations.

5.5.1 Parameters

Our assumptions for the following sections are as given in Section 5.2. The parameters for the numerical simulation are listed in Table 5.5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of breaks per match</td>
<td>50</td>
</tr>
<tr>
<td>Number of advertisers</td>
<td>20</td>
</tr>
<tr>
<td>Ad lengths (s)</td>
<td>10, 20, 30</td>
</tr>
<tr>
<td>Revenue per second per advertiser</td>
<td>Randomly drawn from [3500, 5500]</td>
</tr>
<tr>
<td>Break length distribution (s)</td>
<td>( \sim U(10, 60) )</td>
</tr>
<tr>
<td>Number of random trials (matches)</td>
<td>100</td>
</tr>
<tr>
<td>Target Service Level</td>
<td>80% of time sold</td>
</tr>
<tr>
<td>Penalty for not reaching service level</td>
<td>1000$ for each second below target</td>
</tr>
</tbody>
</table>

Table 5.5: Parameters for Numerical Simulation

For this simulation, we assume that advertisers order advertising time from the network, and that the network commits to a certain service level that is a percentage of the time sold to each advertiser. In our simulation,
we assume that this value is 80%. Further, to discourage not meeting the service level, we set a penalty value of 1000$ for each second short of the promised service level. In the real world, networks either make good on their contract in subsequent tournaments, or show ads at the end of the game to make up on advertising time. For our simplified setup, a penalty of 1000$ suffices to show us the general direction in which we must direct our efforts.

5.5.2 Results

![Figure 5.2: Performance of heuristics with service constraints](image)

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>% of PI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>99%</td>
</tr>
<tr>
<td>CE</td>
<td>74%</td>
</tr>
<tr>
<td>DCE</td>
<td>90%</td>
</tr>
<tr>
<td>DMCE</td>
<td>93%</td>
</tr>
</tbody>
</table>

Table 5.6: Performance of heuristics relative to PI

Among the heuristics selected, while the Greedy does not create bundles beforehand, the CE, DCE and DMCE heuristics rely on creating bundles, either before the match or during the match. From the results, we see that the Greedy performs almost on par with the PI, while heuristics that depend on creating bundles before the match do not perform as well.

When break sizes are known before the commencement of the break,
creating bundles and trying to fit the best one offers us no advantage over myopically choosing the best ads to fit the break. Since the size of the bundle is fixed, when we run out of appropriately sized bundles to fit into the break, we are forced to schedule a bundle that is of smaller size than the break, hence losing out on earning opportunities. Having bundles that fit the break by regenerating them (DMCE) does not guarantee us optimum ad selection, since valuable ads that could have been scheduled in the current break may be included in a bundle of a different size, and hence not scheduled. Therefore, the Greedy is able to best capitalize on the advance knowledge of break sizes.

We note that the Optimal Policy as described in Chapter 3 does not rely on creating bundles before the realization of breaks. In our discussions with the sports broadcaster, we found that the ad scheduling team did create bundles beforehand, but the bundles were discarded when they didn’t have the right bundle for a break. The network should therefore stick to a flexible schedule that does not depend on pre-created bundles, affording flexibility in scheduling and giving them a better chance to earn higher revenues.

5.6 Comparison of Greedy and Optimal Policies

We next study numerically the conditions that determine how well the Optimal Policy performs over the Greedy Policy. In this study, we only consider the Base Case as presented in Section 3.2.1, where breaks and ads are of two sizes, short (15 seconds) and long (30 seconds), and the number of breaks is fixed.

We vary parameters across the relative value of small and large ads, the distribution of the two types of ads, and the Service Level, defined as the percentage of air time sold that is expected to be aired (the lower the percentage, the higher the air time sold). The parameters are summarized in
Table 5.7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio of values of small to large ads</td>
<td>1, 0.55, 0.50, 0.45, 0.40, 0.35, 0.30</td>
</tr>
<tr>
<td>Ratio of number of small to large ads</td>
<td>3, 2, 1, 1/2, 1/3</td>
</tr>
<tr>
<td>Service Level</td>
<td>80%, 90%, 100%, 110%</td>
</tr>
</tbody>
</table>

Table 5.7: Parameters for comparison of Greedy and Optimal Policy

The results are presented in Appendix B. The Revenues tables in Section B.1 show the potential revenue that could be earned if all orders that were accepted could be shown, as well as the performance of the Greedy Policy and the Optimal Policy as a percentage of this total. The Service Level tables shown in Section B.1 show the percentage of ads, small and long, that were shown, and the Utilization tables in Section B.3 show the percentage of total break time that was utilized for showing ads.

5.6.1 Results

The percentage gain of the Optimal Policy over the Greedy Policy against the variation in the relative value of small ads is shown in Figure 5.3, where each graph is drawn for a particular service level.

We observe that gains of Optimal Policy over Greedy Policy monotonically decrease as the value of small ads decreases in comparison to large ads. While the greatest gains are seen when small ads are almost as valuable as large ads, when the relative value of small ads is 0.45 or less, the Optimal Policy shows no gains over the Greedy Policy.

This behavior can be explained by the Optimal Policy having a lower threshold for large ads, and therefore its tendency to schedule more large ads than the Greedy Policy (as discussed in Section 3.2.1 and Section 4.1.2). When small ads are as valuable as large ads, the Greedy Policy schedules small ads up front in the large breaks, since it earns twice as much with two small ads than one large ad. This behavior causes it to run out of small ads
earlier than the Optimal Policy would, and it fails to schedule small ads in the small breaks that occur at some point after it runs out of small ads.

The difference in the service levels (tables in Section B.2) of large ads is acute when the ratio of small ads to large ads is 1:2, where we have half as many small ads as large ads, causing the Greedy Policy to run out of small ads early in a match.

When the ratio of small ads to large ads is 1:3, however, the gains made by Optimal Policy drop below those made when the ratio is 1:2. While the Optimal Policy does not exhaust its inventory of small ads as early as the Greedy Policy, the number of small ads is small enough for it to lose scheduling opportunities in small breaks at the later stages of a game. Tables in Section B.3 show that the utilization of breaks by the Optimal Policy when we have a 1:3 distribution is consistently less than 100%, and is always less than the utilization when ads are distributed by a 1:2 ratio.

When the relative value of small ads is 0.45 or less, the Greedy Policy schedules large ads just as often as the Optimal Policy, since two small ads no longer have as much value as one large ad. Thus we see no difference in either the revenues earned, or the service levels of small and large ads.

It can be argued that when the relative value of small ads is greater than 1, we would see that the gains made by the Optimal Policy decrease once again (compared to the case where the relative value of small ads is 1). Despite the Optimal Policy having a low threshold for large ads, the small ads be valuable enough for the most valuable large ad to not make that threshold, causing the Optimal Policy and the Greedy Policy to schedule similarly. Having small ads of relative value greater than 1, however, is only of academic interest, and we do not discuss it in detail.

From a managerial perspective, we see that the Greedy Policy is just as effective as the Optimal Policy when the inventory has small ads that are less than half the value of large ads. The Greedy Policy is easily implemented, and the network broadcaster need not invest in forward looking heuristics in
such a case. Conversely, as the relative value of small ads increases above the 0.5 mark, the network broadcaster can significantly improve his revenues by implementing the Optimal Policy.

We also note that the greatest gains made by using the Optimal Policy occur when the ratio of small ads to large ads is 1:2. This may occur when the broadcaster has priced his small ads to a level where advertisers see more value in buying large ads, thereby skewing the ad distribution.

Finally, we note that service levels between 80% and 90% provide the best returns for using the Optimal Policy, and we note that our discussions with the network broadcaster showed that service levels in that range were usually targeted.
Chapter 6

Conclusion

TV networks showing live sports are often challenged by having breaks of non deterministic size, and the high profit margins in live sports broadcasting demand a better way of scheduling ads in such situations. In this dissertation we have discussed a method to schedule ads optimally when breaks are of random size and number, and the broadcaster has ads of two lengths.

Earlier literature related to advertising scheduling assume fixed break durations, and do not sufficiently answer how ads must be scheduled when faced with uncertainty in break sizes. Literature related to Random Yield, Stochastic Knapsack, and Stochastic Recourse do not sufficiently match the setting typical of our problem.

We find that the Optimal Policy when faced with non-deterministic breaks is a forward looking Greedy implementation. Bundling strategies fail to sufficiently account for the stochasticity in break sizes, and earn less than a flexible heuristic such as the Greedy Policy. Further, we show that the Optimal Policy outperforms the Greedy Policy when small ads have a value equal to or greater than half the value of a large ad.

While we do not account for all the constraints that broadcasters face, our model is general enough to be applied to a class of bin packing problems, for instance, cargo shipping when containers have non-deterministic capacity. This simple model, however, does not fully meet the network broadcaster’s
requirements. There is scope for the following extensions to this work:

1. Inclusion of diversity constraints into the Optimal Policy: two ads from the same advertiser, or from competing advertisers, cannot be shown in a break.

2. Optimal Policy when breaks are not all IID: for instance a rain break or an injury break takes a longer time than mid-over breaks.

3. Incorporating service levels constraints for advertisers, agencies, and geographic regions.

4. Extension to help managers in making pricing decisions and accepting spot orders based on the Optimal Policy.

5. A study of Broadcaster-Advertiser behavior based on Game Theoretic principles when the broadcaster employs the Optimal Policy.

To conclude, the area of Optimal scheduling of items (ads, cargo, etc) in non-deterministic containers is an area that has many possibilities for research and development. It is hoped that this dissertation is a stepping stone in establishing improved heuristics in this area.
Bibliography


Appendix A

Proofs of theorems

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_i(l,s)$</td>
<td>Expected revenue earned from break $i$ onwards, when the indexes of the first large ad and the first small ad available are $l$ and $s$ respectively</td>
</tr>
<tr>
<td>$R_l$</td>
<td>Expected revenue earned by selecting $l_1$ in the current (large) break</td>
</tr>
<tr>
<td>$R_s$</td>
<td>Expected revenue earned by selecting $s_1 + s_2$ in the current (large) break</td>
</tr>
<tr>
<td>$b$</td>
<td>Number of breaks remaining</td>
</tr>
</tbody>
</table>

Table A.1: Summary of notation

A.1 Proof of Theorem 3.1

Proof. From Lemma 3.1 and Lemma 3.2, we know the rule to be true when one or two breaks remain.

Let us assume the rule be true for breaks 2, ..., $n$. We will use then use induction to prove this theorem.
Then,

\[ R_l = l_1 + V_2(2,1) \]
\[ R_s = s_1 + s_2 + V_2(1,3) \]
\[ V_n(1,1) = \max\{R_l, R_s\} \]

We prove that \( R_l - R_s \geq 0 \) when \( l_1 \geq s_n + s_{n+1} \).

\[
R_l - R_s = l_1 + V_2(2,1) - [s_1 + s_2 + V_2(1,3)] \\
= (l_1 - (s_1 + s_2)) + (V_2(2,1) - V_2(1,3))
\]

Expanding the above to a look ahead of two breaks, we get:

\[
R_l - R_s = (l_1 - (s_1 + s_2)) \\
+ p[s_1 + V_3(2,2)] - p[s_3 + V_3(1,4)] \\
+ (1-p)\max\begin{cases} \\
  l_2 + V_3(3,1), & \text{if } l_2 > s_{n-1} + s_n \\
  s_1 + s_2 + V_3(2,3) & \text{otherwise}
\end{cases} \\
- (1-p)(l_1 + V_3(2,3))
\]

Though we have no information about \( l_2 \) and \( s_{n-1} + s_n \), the \( \max \) operator guarantees that the value of \( V_2(2,1) \) must at least be \( s_1 + s_2 + V_3(2,3) \). Thus we get:

\[
R_l - R_s \geq (l_1 - (s_1 + s_2)) \\
+ p[s_1 + V_3(2,2) - s_3 - V_3(1,4)] \\
+ (1-p)[s_1 + s_2 + V_3(2,3) - l_1 - V_3(2,3)] \\
\geq p[l_1 - (s_2 + s_3) + V_3(2,2) - V_3(1,4)]
\]
Continuing enumeration to look ahead for $i < n$ breaks, we get

$$R_l - R_s \geq p^{i-1} \left[ l_1 - (s_i + s_{i+1}) + V_{n-i}(2,i) - V_{n-i}(1,i+2) \right]$$

Thus, for $i = n - 1$, we get:

$$R_l - R_s \geq p^{n-2} \left[ l_1 - (s_{n-1} + s_n) + V_n(2,n-1) - V_n(1,n+1) \right]$$

$$\geq p^{n-2} \left[ l_1 - (s_{n-1} + s_n) + p(s_{n-1} - s_{n+1}) + (1-p)(s_{n-1} + s_n - l_1) \right]$$

$$\geq p^{n-2} \left[ p(l_1 - (s_n + s_{n+1})) \right]$$

$$\geq 0 \quad (\because p \geq 0, \ l_1 \geq (s_n + s_{n+1})$$

Therefore we have proved that when the number of breaks remaining is $n$ and $l_1 \geq s_n + s_{n+1}$, it is optimal to pack $l_1$ first.

The reverse case, i.e. $R_s - R_l \geq 0$ when $s_n + s_{n+1} \geq l_1$ can be proved similarly. \qed

### A.2 Proof of Theorem 3.2

**Proof.** From Lemma 3.1 and Lemma 3.3, we know the rule to be true when one and two breaks remain.

Let us assume the rule be true when the number breaks remaining are $1, 2, \ldots, n - 1$. We will use then use induction to prove this theorem. As before, we use $R_l$ and $R_s$ to denote the revenues earned by scheduling $l_1$ and $(s_1, s_2)$ respectively in $b_1$. We have:

$$R_l = l_1 + V_2(2,1)$$

$$R_s = s_1 + s_2 + V_2(1,3)$$
Then,

\[ R_l - R_s = l_1 + V_2(2, 1) - [s_1 + s_2 + V_2(1, 3)] \]
\[ = (l_1 - (s_1 + s_2)) + (V_2(2, 1) - V_2(1, 3)) \]
\[ = (l_1 - (s_1 + s_2)) + \left[ p_0(V_3(2, 1) - V_3(1, 3)) \right. \]
\[ + \left. p_1(s_1 + V_3(2, 2) - s_3 - V_3(1, 4)) \right. \]
\[ + \left. p_2(\max\{l_1 + V_3(3, 1), s_1 + s_2 + V_3(2, 3)\} \right. \]
\[ - \left. \max\{l_1 + V_3(2, 3), s_3 + s_4 + V_3(1, 5)\}\right] \]

We can trivially prove that

\[(p_0 + p_1)^{n-1}l_1 \geq \sum_{i=0}^{n-1} \left[ C_i^{n-1} p_0^{n-1-i} p_1^i (s_{i+1} + s_{i+2}) \right] \]

\[\implies (p_0 + p_1)^{n-2}l_1 \geq \sum_{i=0}^{n-2} \left[ C_i^{n-2} p_0^{n-2-i} p_1^i (s_{i+3} + s_{i+4}) \right]\] (A.1)

Using Equation (A.1) and using induction, we can say that

\[ V_2(1, 3) = p_0V_3(1, 3) + p_1(s_3 + V_3(1, 4)) + p_2(l_1 + V_3(2, 3)) \]

Substituting for \( R_l - R_s \), we get:

\[ R_l - R_s \geq (l_1 - (s_1 + s_2)) \]
\[ + \left[ p_0(V_3(2, 1) - V_3(1, 3)) \right. \]
\[ + \left. p_1(s_1 + V_3(2, 2) - s_3 - V_3(1, 4)) \right. \]
\[ + \left. p_2(s_1 + s_2 + V_3(2, 3) - l_1 - V_3(2, 3)) \right. \]
\[ \geq (1 - p_2)(l_1 - (s_1 + s_2)) \]
\[ + \left[ p_0(V_3(2, 1) - V_3(1, 3)) \right. \]
\[ + \left. p_1(s_1 + V_3(2, 2) - s_3 - V_3(1, 4)) \right. \]
\[\geq (p_0 + p_1)l_1 - p_0(s_1 + s_2) - p_1(s_2 + s_3)\]
\[+ \left[p_0(V_3(2,1) - V_3(1,3))\right.\]
\[+ \left.\quad + p_1(V_3(2,2) - V_3(1,4))\right]\]

On expanding, and applying induction throughout,

\[p_0(V_3(2,1) - V_3(1,3)) + p_1(V_3(2,2) - V_3(1,4))\]
\[\geq p_0\left[p_0(V_4(2,1) - V_4(1,3))\right.\]
\[+ \left.\quad + p_1(s_1 + V_4(2,2) - s_3 - V_4(1,4))\right.\]
\[+ \left.\quad + p_2(s_1 + s_2 + V_4(2,3) - l_1 - V_4(2,3))\right]\]
\[+ p_1\left[p_0(V_4(2,2) - V_4(1,4))\right.\]
\[+ \left.\quad + p_1(s_2 + V_4(2,3) - s_4 - V_4(1,5))\right.\]
\[+ \left.\quad + p_2(s_2 + s_3 + V_4(2,4) - l_1 - V_4(2,4))\right]\]

Substituting for \(R_l - R_s\), we get:

\[R_l - R_s \geq (p_0 + p_1)^2l_1\]
\[- p_0^2(s_1 + s_2) - 2p_0p_1(s_2 + s_3) - p_1^2(s_3 + s_4)\]
\[+ p_0^2(V_4(2,1) - V_4(1,3))\]
\[+ 2p_0p_1(V_4(2,2) - V_4(1,4))\]
\[+ p_1^2(V_4(2,3) - V_4(1,5))\]
Continuing to enumerate in this fashion and applying induction, we get:

\[
R_l - R_s \geq (p_0 + p_1)^{n-2} l_1 - p_0^{n-2}(s_1 + s_2) - (n-2)p_0^{n-3}p_1(s_2 + s_3) - \ldots - p_1^{n-2}(s_{n-1} + s_n)
\]

\[
+ \quad p_0^{n-2}(V_n(2,1) - V_n(1,3))
\]

\[
+ \quad (n-2)p_0^{n-3}p_1(V_1(2,2) - V_1(1,4))
\]

\[
\vdots
\]

\[
+ \quad p_1^{n-2}(V_1(2,n-1) - V_n(1,n+1))
\]

\[
\geq (p_0 + p_1)^{n-1} l_1 - \sum_{i=0}^{n-1} \left[ C(n-1,i) p_0^{n-1-i} p_1^i (s_{i+1} + s_{i+2}) \right]
\]

\[
\geq 0 \text{ (by definition)}
\]

The proof for the reverse case, i.e. $R_s - R_l \geq 0$ when

\[
(p_0 + p_1)^{n-1} l_1 < \sum_{i=0}^{n-1} \left[ C(n-1,i) p_0^{n-1-i} p_1^i (s_{i+1} + s_{i+2}) \right]
\]

can be proved similarly.

\[\square\]

### A.3 Proof of Theorem 3.3

**Proof.** We have proved the strategy to be true when $k = 1$ (Lemma 3.4) and $k = 2$ (Lemma 3.5).

To prove the strategy is true when $k = n$, let us assume the strategy holds good for breaks $(2,3,\ldots,n)$, i.e. for all subsequent breaks.

As before, let $\hat{O} = (l_1,\ldots,l_\lambda,s_1,\ldots,s_{m-2\lambda})$.

Let $O_l = (l_1,\ldots,l_\lambda,l_{\lambda+1},s_1,\ldots,s_{m-2\lambda-2})$ and $O_s = (l_1,\ldots,l_{\lambda-1},s_1,\ldots,s_{m-2\lambda+2})$.

Let $R_o$, be the revenue earned by selecting the ads in $\hat{O}$, and let $R_l$ and $R_s$
be the revenues earned by selecting \( O_l \) and \( O_s \) respectively.

\[
R_o = \sum_{i=1}^{\lambda} l_i + \sum_{j=1}^{m-2\lambda} s_j + V_2(\lambda+1, m-2\lambda+1) \quad (A.2)
\]

\[
R_l = \sum_{i=1}^{\lambda+1} l_i + \sum_{j=1}^{m-2\lambda-2} s_j + V_2(\lambda+2, m-2\lambda-1) \quad (A.3)
\]

\[
R_s = \sum_{i=1}^{\lambda-1} l_i + \sum_{j=1}^{m-2\lambda+2} s_j + V_2(\lambda, m-2\lambda+3) \quad (A.4)
\]

We have to prove that:

1. \( R_o \geq R_l \) and
2. \( R_o \geq R_s \)

**Case 1:**

\[
R_o - R_l = \sum_{i=1}^{\lambda} l_i + \sum_{j=1}^{m-2\lambda} s_j + V_2(\lambda+1, m-2\lambda+1)
\]

\[
- \left[ \sum_{i=1}^{\lambda+1} l_i + \sum_{j=1}^{m-2\lambda-2} s_j + V_2(\lambda+2, m-2\lambda-1) \right]
\]

\[
= -l_{\lambda+1} + s_{m-2\lambda-1} + s_{m-2\lambda}
\]

\[
+ V_2(\lambda+1, m-2\lambda+1) - V_2(\lambda+2, m-2\lambda-1)
\]

Note that \( l_{\lambda+2} < s_{m-2\lambda+n-2} + s_{m-2\lambda+n-1} \). By a similar argument as given in Equation 3.7, we can write:

\[
V_2(\lambda+1, m-2\lambda+1) - V_2(\lambda+2, m-2\lambda-1)
\]

\[
\geq p_1 \left( s_{m-2\lambda+1} + V_3(\lambda+1, m-2\lambda+2) - s_{m-2\lambda-1} - V_3(\lambda+2, m-2\lambda) \right)
\]

\[
+ \sum_{k=2}^{M} p_k \left( l_{\lambda+1} - [s_{m-2\lambda-1} + s_{m-2\lambda}] \right)
\]
Substituting in $R_o - R_l$, we get:

\[
R_o - R_l \geq -l_{\lambda+1} + s_{m-2\lambda-1} + s_{m-2\lambda} + p_1 \left( s_{m-2\lambda+1} + V_3(\lambda+1, m-2\lambda+2) - s_{m-2\lambda-1} - V_3(\lambda+2, m-2\lambda) \right) + \sum_{k=2}^{M} p_k (l_{\lambda+1} - [s_{m-2\lambda-1} + s_{m-2\lambda}]) \\
\geq p_1 [-l_{\lambda+1} + s_{m-2\lambda} + s_{m-2\lambda+1} + V_3(\lambda+1, m-2\lambda+2) - V_3(\lambda+2, m-2\lambda)]
\]

Continuing enumeration of the above to expand to all $n$ breaks, we get:

\[
R_o - R_l \geq p_1^{n-1} [-l_{\lambda+1} + s_{m-2\lambda+n-2} + s_{m-2\lambda+n-1}] \\
\geq 0 \quad (\because l_{\lambda+1} < s_{m-2(\lambda+1)+n} + s_{m-2(\lambda+1)+n+1})
\]

Similarly, we can prove **Case 2** by showing that

\[
R_o - R_s \geq p_1^{n-1} [l_{\lambda} - s_{m-2\lambda+n} - s_{m-2\lambda+n+1}] \geq 0
\]

\[\square\]

### A.4 Proof of Theorem 3.4

**Proof.** We have proved the strategy to be true when $k = 1$ (Lemma 3.6) and $k = 2$ (Lemma 3.7).

To prove the strategy is true when $k = n$, let us assume the strategy hold good for breaks $(2, \ldots, n)$, i.e. for all subsequent breaks.

We define $\hat{O}, O_l, O_s$ as before.

Let $R_o$, $R_l$, and $R_s$ denote the revenues earned by selecting $\hat{O}$, $O_l$ and $O_s$ respectively. We have to prove that
1. \( R_o \geq R_l \)

2. \( R_o \geq R_s \)

**Case 1:**

By definition:

\[
(p_0 + p_1)^{n-1}l_\lambda \geq \sum_{i=0}^{n-1} \binom{n-1}{i} p_0^{n-1-i} p_1^i(s_{m-2\lambda+i+1} + s_{m-2\lambda+i+2})
\]  
(A.5)

and \((p_0 + p_1)^{n-1}l_{\lambda+1} < \sum_{i=0}^{n-1} \binom{n-1}{i} p_0^{n-1-i} p_1^i(s_{m-2\lambda+i-1} + s_{m-2\lambda+i})\)  
(A.6)

As before,

\[
R_o - R_l = -l_{\lambda+1} + s_{m-2\lambda-1} + s_{m-2\lambda}
+ V_2(\lambda + 1, m - 2\lambda + 1) - V_2(\lambda + 2, m - 2\lambda - 1)
\]

From eq. A.6,

\[
(p_0 + p_1)^{n-1}l_{\lambda+2} < \sum_{i=0}^{n-1} \binom{n-1}{i} p_0^{n-1-i} p_1^i(s_{m-2\lambda+i-1} + s_{m-2\lambda+i})
\]

\[
\Rightarrow (p_0 + p_1)^{n-1}l_{\lambda+2} < \sum_{i=0}^{n-1} \binom{n-1}{i} p_0^{n-1-i} p_1^i(s_{m-2\lambda+i-3} + s_{m-2\lambda+i-2})
\]

Then by induction, for \( b_2 \geq 2S \), \( V_2(\lambda + 2, m - 2\lambda - 1) \) earns us \((s_{m-2\lambda-1} + s_{m-2\lambda})\) followed by the sum of values of ads selected from the set \(\{l_{\lambda+2}, \ldots, s_{m-2\lambda+1}, \ldots\}\).

On the other hand, with \( V_2(\lambda + 1, m - 2\lambda + 1) \) we earn at least \( l_{\lambda+1} \) followed by the sum of values of ads from the set \(\{l_{\lambda+2}, \ldots, s_{m-2\lambda+1}, \ldots\}\).
Therefore:

\[
V_2(\lambda + 1, m - 2\lambda + 1) - V_2(\lambda + 2, m - 2\lambda - 1) \\
\geq p_0(V_3(\lambda + 1, m - 2\lambda + 1) - V_3(\lambda + 2, m - 2\lambda - 1)) \\
+ p_1(s_{m-2\lambda+1} - s_{m-2\lambda-1} + V_3(\lambda + 1, m - 2\lambda + 2) - V_3(\lambda + 2, m - 2\lambda)) \\
+ \sum_{i=2}^{M} p_i (l_{\lambda+1} - [s_{m-2\lambda-1} + s_{m-2\lambda}])
\]

Substituting for \( R_o - R_l \), we get:

\[
R_o - R_l \geq -l_{\lambda+1} + s_{m-2\lambda-1} + s_{m-2\lambda} \\
+ p_0(V_3(\lambda + 1, m - 2\lambda + 1) - V_3(\lambda + 2, m - 2\lambda - 1)) \\
+ p_1(s_{m-2\lambda+1} - s_{m-2\lambda-1} + V_3(\lambda + 1, m - 2\lambda + 2) - V_3(\lambda + 2, m - 2\lambda)) \\
+ \sum_{i=2}^{M} p_i (l_{\lambda+1} - [s_{m-2\lambda-1} + s_{m-2\lambda}])
\]
\[ l_{\lambda+1}(p_0 - p_1 + p_0(1 - p_0 - p_1) + p_1(1 - p_0 - p_1)) + s_{m-2\lambda-1}(p_0 - p_0 p_1 - p_0(1 - p_0 - p_1)) + s_{m-2\lambda}(p_0 + p_1 - p_0(1 - p_0 - p_1) - p_1^2 - p_1(1 - p_0 - p_1)) + s_{m-2\lambda+1}(p_1 + p_0 p_1 - p_1(1 - p_0 - p_1)) + s_{m-2\lambda+2}(p_1^2) + p_0^2(V_4(\lambda + 1, m - 2\lambda + 1) - V_4(\lambda + 2, m - 2\lambda - 1)) + 2p_0 p_1(V_4(\lambda + 1, m - 2\lambda + 2) - V_4(\lambda + 2, m - 2\lambda)) + p_1^2(V_4(\lambda + 1, m - 2\lambda + 3) - V_4(\lambda + 2, m - 2\lambda + 1)) \geq -l_{\lambda+1}(p_0 + p_1)^2 + p_0^2(s_{m-2\lambda-1} + s_{m-2\lambda}) + 2p_0 p_1(s_{m-2\lambda} + s_{m-2\lambda+1}) + p_1^2(s_{m-2\lambda+1} + s_{m-2\lambda+2}) + p_0(V_4(\lambda + 1, m - 2\lambda + 1) - V_4(\lambda + 2, m - 2\lambda - 1)) + 2p_0 p_1(V_4(\lambda + 1, m - 2\lambda + 2) - V_4(\lambda + 2, m - 2\lambda)) + p_1^2(V_4(\lambda + 1, m - 2\lambda + 3) - V_4(\lambda + 2, m - 2\lambda + 1)) \]

Continuing to enumerate in this fashion, and applying induction, after expanding up to \( V_n \) we get:

\[
R_o - R_t \geq -l_{\lambda+1}(p_0 + p_1)^{n-2} + \sum_{i=0}^{n-2} \binom{n-2}{i} p_0^{n-2-i} p_1^i (s_{m-2\lambda+i-1} + s_{m-2\lambda+i}) + \sum_{i=0}^{n-2} \binom{n-2}{i} p_0^{n-2-i} p_1^i (V_4(\lambda + 1, m - 2\lambda + i + 1) - V_4(\lambda + 2, m - 2\lambda + i - 1)) \geq -(p_0 + p_1)^{n-1} l_{\lambda+1} \]

\[
+ \sum_{i=0}^{n-1} \binom{n-1}{i} p_0^{n-1-i} p_1^i (s_{m-2\lambda+i-1} + s_{m-2\lambda+i}) \geq 0 \text{ from eq A.6}
\]

**Case 2:** \( R_o - R_s \geq 0 \) can be similarly proved using eq. A.5. \(\square\)
A.5 Proof of Theorem 3.5

Proof. We have proved all three cases to be true when one break and two breaks remain. Let the rule be true for all subsequent breaks, $b_2, ..., b_n$. We will then prove by induction that it is true for break $b_1$.

Case 1: Let $(1 - p)b_1 = ps_1 + (1 - p)(s_1 + s_2) + t$, where $t \geq 0$.

Then,

$$R_l = pV_2(1, 1) + (1 - p)(l_1 + V_2(2, 1))$$

$$R_s = p(s_1 + V_2(1, 2)) + (1 - p)(s_1 + s_2 + V(2, 1))$$

Thus, we get:

$$R_l - R_s = p[V_2(1, 1) - s_1 - V_2(1, 2)]$$

$$+ (1 - p)[l_1 - (s_1 + s_2) + V_2(2, 1) - V_2(1, 3)]$$

$$= t + p[V_2(1, 1) - V_2(1, 2)] + (1 - p)[V_2(2, 1) - V_2(1, 3)]$$

Expanding the above to look ahead two breaks, we get:

$$R_l - R_s = t + p[V_2(1, 1) - V_2(1, 2)] + (1 - p)[V_2(2, 1) - V_2(1, 3)]$$

$$\geq t + p \left[ pV_3(1, 1) + (1 - p)(l_1 + V_3(2, 1)) 
- pV_3(1, 2) - (1 - p)(l_1 + V_3(2, 2)) \right]$$

$$+ (1 - p) \left[ p(s_1 + V_3(2, 2)) + (1 - p)(s_1 + s_2 + V_3(2, 3)) 
- pV_3(1, 3) - (1 - p)(l_1 + V_3(2, 3)) \right]$$

$$\geq t + p \left[ p(V_3(1, 1) - V_3(1, 2)) + (1 - p)V_3(2, 1) \right]$$

$$+ (1 - p) \left[ ps_1 + (1 - p)(s_1 + s_2) - pV_3(1, 3) - (1 - p)l_1 \right]$$

$$\geq t + p \left[ p(V_3(1, 1) - V_3(1, 2)) 
+ (1 - p)(V_3(2, 1) - V_3(1, 3)) \right] - (1 - p)t$$
Continuing to expand the above to look ahead for \( n - 1 \) breaks, we get:

\[
R_l - R_s \geq p^{n-2} \left[ t + p[V_n(1,1) - V_n(1,2)] + (1 - p)[V_n(2,1) - V_n(1,3)] \right]
\]

\[
\geq p^{n-2} \left[ t + p[(1-p)l_1 - (1-p)l_1] + (1-p)[ps_1 + (1-p)(s_1 + s_2) - (1-p)l_1] \right]
\]

\[
\geq p^{n-2} \left[ t + (1-p)(-t) \right]
\]

\[
\geq p^{n-1}t
\]

\[
\geq 0, \quad \text{since} \quad t \geq 0, \quad p \geq 0
\]

Therefore \( R_l - R_s \geq 0 \implies \) selecting \( l_1 \) is profitable when \((1-p)l_1 \geq ps_1 + (1-p)(s_1 + s_2)\)

**Case 2:** Let \((1-p)l_1 < ps_n + (1-p)(s_n + s_{n+1})\) Then

\[
R_s - R_l = ps_1 + (1-p)(s_1 + s_2) - (1-p)l_1
\]

\[
+ p[V_2(1,2) - V_2(1,1)] + (1-p)[V_2(1,3) - V_2(2,1)]
\]

\[
\geq ps_1 + (1-p)(s_1 + s_2) - (1-p)l_1
\]

\[
+ p \left[ pV_3(1,2) + (1-p)(l_1 + V_3(2,2)) \right.
\]

\[
- p(s_1 + V_3(1,2)) - (1-p)(s_1 + s_2 + V_3(1,3)]
\]

\[
+ (1-p) \left[ pV_3(1,3) + (1-p)(l_1 + V_3(2,3)) \right.
\]

\[
- p(s_1 + V_3(2,2)) - (1-p)(s_1 + s_2 + V_3(2,3)]
\]

\[
\left[ \therefore \ V_2(1,i) \geq (1-p)(l_1 + V_3(2,i)) \forall i < n \right]
\]

\[
\geq 0
\]

\[
R_s - R_l \geq 0 \implies
\]
it is optimal to schedule $s_1 + s_2$ when $(1-p)l_1 < ps_n + (1-p)(s_n + s_{n+1})$.

**Case 3:** Let $ps_1 + (1-p)(s_1 + s_2) = (1-p)l_1 + t$ where $t \geq 0$

$$R_l - R_s = (1-p)l_1 - ps_1 - (1-p)(s_1 + s_2)$$
$$+ p[V_2(1, 1) - V_2(1, 2)] + (1-p)[V_2(2, 1) - V_2(1, 3)]$$
$$= -t + p[p(s_1 + V_3(1, 2)) + (1-p)(s_1 + s_2 + V_3(1, 3))$$
$$- pV_3(1, 2) - (1-p)(l_1 + V_3(2, 2))]$$
$$+ (1-p)[p(s_1 + V_3(2, 2)) + (1-p)(s_1 + s_2 + V_3(2, 3))$$
$$- pV_3(1, 3) - (1-p)(l_1 + V_3(2, 3))]$$

In the above step, note that for $V_2(1, 1)$, in break $b_2$, $(1-p)l_1$ need not exceed $ps_{n-1} + (1-p)(s_{n-1} + s_n)$, however $ps_1 + (1-p)(s_1 + s_2) \geq (1-p)l_1$. Since we assume that the policy holds good in subsequent breaks, either Case 2 ($(s_1, s_2)$ preferred) or Case 3 (indifference) will apply, therefore we select $(s_1, s_2)$. The expansion for the other terms is similarly explained.

$$\Rightarrow R_l - R_s = -t + p[t + p[V_3(1, 2) - V_3(1, 2)] + (1-p)[V_3(1, 3) - V_3(2, 2)]$$
$$+ p[V_3(2, 2) - V_3(1, 3)]$$
$$+ (1-p)[V_3(2, 3) - V_3(2, 3)]]$$
$$= -t + pt + (1-p)t$$
$$= 0$$

$R_l - R_s = 0 \Rightarrow$ selecting $l_1$ and $s_1 + s_2$ are both equally profitable.  

**A.6 Proof of Theorem 3.6**

*Proof.* Let $p_2l_1 = t + p_1s_1 + p_2(s_1 + s_2)$ where $t \geq 0$. 
Case 1:

\[ R_l - R_s = (p_0 + p_1)V_2(1,1) + p_2(l_1 + V_2(2,1)) \]

\[ = p_0V_2(1,1) - p_1(s_1 + V_2(1,2)) - p_2(s_1 + s_2 + V_2(1,3)) \]

\[ = p_1(V_2(1,1) - s_1 - V_2(1,2)) \]

\[ + p_2(l_1 + V_2(2,1) - (s_1 + s_2) - V_2(1,3)) \]

\[ = t + p_1[V_2(1,1) - V_2(1,2)] + p_2[V_2(2,1) - V_2(1,3)] \]

\[ \geq t + p_1 \left[ p_0V_3(1,1) + p_1(s_1 + V_3(1,2)) + p_2(s_1 + s_2 + V_3(1,3) \right] \]

\[ - (p_0 + p_1)(V_3(1,2) - p_2(l_1 + V_3(2,2)) \]

\[ + p_2 \left[ p_0V_3(2,1) + p_1(s_1 + V_3(2,2)) + p_2(s_1 + s_2 + V_3(2,3) \right] \]

\[ - (p_0 + p_1)(V_3(1,3) - p_2(l_1 + V_3(2,3)) \]

\[ \geq p_0 \left[ t + p_1(V_3(1,1) - V_3(1,2)) + p_2(V_3(2,1) - V_3(1,3)) \right] \]

Continuing enumeration, we get:

\[ R_l - R_s \geq p_0^n - 2 \left[ t + p_1(V_n(1,1) - V_n(1,2)) + p_2(V_n(2,1) - V_n(1,3)) \right] \]

\[ \geq p_0^n - 2 \left[ t + p_1(p_1s_1 + p_2(s_1 + s_2) - p_2l_1) + p_2(p_1s_1 + p_2(s_1 + s_2) - p_2l_1) \right] \]

\[ \geq p_0^n - 2 \left[ t + p_1(-t) + p_2(-t) \right] \]

\[ \geq p_0^n - 1 t \]

\[ \geq 0 \]

\[ R_l - R_s \geq 0 \implies \text{it is optimal to schedule } l_1 \text{ first.} \]

Case 2 can be similarly proved by showing that \( R_s - R_l \geq p_0^n - 1 t \geq 0 \).

For both cases above, \( R_l > R_s \) and \( R_s > R_l \) if \( t > 0 \) and \( p_0 > 0 \). There is no interval where \( R_l = R_s \), unlike the cases where the number of breaks is fixed. \( \square \)
Appendix B

Tables
In this section, we present the results of the study of Section 5.6.

### B.1 Revenues

#### Value of Small ad to Large ad = 1:1

<table>
<thead>
<tr>
<th>Service Level</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>562907</td>
<td>482600</td>
<td>86.39%</td>
<td>526768</td>
<td>449395</td>
<td>84.74%</td>
<td>468266</td>
<td>407903</td>
<td>87.02%</td>
<td>422645</td>
<td>365022</td>
<td>85.30%</td>
</tr>
<tr>
<td>OP</td>
<td>499790</td>
<td>449915</td>
<td>88.75%</td>
<td>470848</td>
<td>421150</td>
<td>90.00%</td>
<td>418699</td>
<td>373230</td>
<td>84.99%</td>
<td>373653</td>
<td>327070</td>
<td>83.51%</td>
</tr>
<tr>
<td>G</td>
<td>456156</td>
<td>402682</td>
<td>90.0%</td>
<td>411416</td>
<td>360844</td>
<td>93.36%</td>
<td>390884</td>
<td>34551</td>
<td>92.48%</td>
<td>34551</td>
<td>301423</td>
<td>75.26%</td>
</tr>
<tr>
<td>OP</td>
<td>467940</td>
<td>411416</td>
<td>94.21%</td>
<td>411416</td>
<td>360844</td>
<td>93.36%</td>
<td>390884</td>
<td>34551</td>
<td>92.48%</td>
<td>34551</td>
<td>301423</td>
<td>75.26%</td>
</tr>
</tbody>
</table>

#### Value of Small ad to Large ad = 0.55:1

<table>
<thead>
<tr>
<th>Service Level</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>372654</td>
<td>311766</td>
<td>83.66%</td>
<td>366942</td>
<td>308720</td>
<td>84.13%</td>
<td>362189</td>
<td>305520</td>
<td>83.80%</td>
<td>360137</td>
<td>298844</td>
<td>82.99%</td>
</tr>
<tr>
<td>OP</td>
<td>331519</td>
<td>293129</td>
<td>83.31%</td>
<td>327327</td>
<td>301177</td>
<td>92.01%</td>
<td>324896</td>
<td>297660</td>
<td>91.62%</td>
<td>318249</td>
<td>289513</td>
<td>90.97%</td>
</tr>
<tr>
<td>G</td>
<td>297895</td>
<td>258347</td>
<td>97.41%</td>
<td>293535</td>
<td>286524</td>
<td>97.01%</td>
<td>289347</td>
<td>279810</td>
<td>96.39%</td>
<td>287060</td>
<td>268923</td>
<td>93.68%</td>
</tr>
<tr>
<td>OP</td>
<td>270960</td>
<td>230747</td>
<td>97.73%</td>
<td>267059</td>
<td>265670</td>
<td>99.48%</td>
<td>265620</td>
<td>256289</td>
<td>97.99%</td>
<td>260188</td>
<td>250708</td>
<td>98.79%</td>
</tr>
</tbody>
</table>

#### Value of Small ad to Large ad = 0.50:1

<table>
<thead>
<tr>
<th>Service Level</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>351843</td>
<td>293112</td>
<td>83.13%</td>
<td>350388</td>
<td>292236</td>
<td>83.40%</td>
<td>350378</td>
<td>293066</td>
<td>83.64%</td>
<td>353217</td>
<td>292116</td>
<td>82.70%</td>
</tr>
<tr>
<td>OP</td>
<td>312939</td>
<td>257030</td>
<td>97.61%</td>
<td>311786</td>
<td>286621</td>
<td>91.93%</td>
<td>313862</td>
<td>287436</td>
<td>91.71%</td>
<td>312619</td>
<td>281689</td>
<td>90.11%</td>
</tr>
<tr>
<td>G</td>
<td>281754</td>
<td>235060</td>
<td>97.61%</td>
<td>281371</td>
<td>274845</td>
<td>97.68%</td>
<td>280336</td>
<td>273236</td>
<td>97.50%</td>
<td>279660</td>
<td>268788</td>
<td>95.14%</td>
</tr>
<tr>
<td>OP</td>
<td>256159</td>
<td>205300</td>
<td>99.66%</td>
<td>254232</td>
<td>253509</td>
<td>99.77%</td>
<td>256459</td>
<td>255609</td>
<td>99.69%</td>
<td>255425</td>
<td>251969</td>
<td>98.65%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service Level</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
<th>Sold</th>
<th>Earned</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>351843</td>
<td>293112</td>
<td>83.13%</td>
<td>350388</td>
<td>292236</td>
<td>83.40%</td>
<td>350378</td>
<td>293066</td>
<td>83.64%</td>
<td>353217</td>
<td>292116</td>
<td>82.70%</td>
</tr>
<tr>
<td>OP</td>
<td>312939</td>
<td>257030</td>
<td>97.61%</td>
<td>311786</td>
<td>286621</td>
<td>91.93%</td>
<td>313862</td>
<td>287436</td>
<td>91.71%</td>
<td>312619</td>
<td>281689</td>
<td>90.11%</td>
</tr>
<tr>
<td>G</td>
<td>281754</td>
<td>235060</td>
<td>97.61%</td>
<td>281371</td>
<td>274845</td>
<td>97.68%</td>
<td>280336</td>
<td>273236</td>
<td>97.50%</td>
<td>279660</td>
<td>268788</td>
<td>95.14%</td>
</tr>
<tr>
<td>OP</td>
<td>256159</td>
<td>205300</td>
<td>99.66%</td>
<td>254232</td>
<td>253509</td>
<td>99.77%</td>
<td>256459</td>
<td>255609</td>
<td>99.69%</td>
<td>255425</td>
<td>251969</td>
<td>98.65%</td>
</tr>
</tbody>
</table>
### Value of Small ad to Large ad = 0.45:1

<table>
<thead>
<tr>
<th>Ad Distribution</th>
<th>3/1</th>
<th>3/1</th>
<th>3/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>270436</td>
<td>99.62%</td>
<td>269509</td>
<td>99.62%</td>
<td>267497</td>
<td>99.62%</td>
<td>260923</td>
<td>99.62%</td>
<td>257497</td>
<td>99.62%</td>
<td>257497</td>
<td>99.62%</td>
</tr>
<tr>
<td>OP</td>
<td>260721</td>
<td>84.10%</td>
<td>260721</td>
<td>84.10%</td>
<td>260721</td>
<td>84.10%</td>
<td>260721</td>
<td>84.10%</td>
<td>260721</td>
<td>84.10%</td>
<td>260721</td>
<td>84.10%</td>
</tr>
<tr>
<td>G</td>
<td>244552</td>
<td>99.73%</td>
<td>244552</td>
<td>99.73%</td>
<td>244552</td>
<td>99.73%</td>
<td>244552</td>
<td>99.73%</td>
<td>244552</td>
<td>99.73%</td>
<td>244552</td>
<td>99.73%</td>
</tr>
<tr>
<td>OP</td>
<td>240346</td>
<td>99.73%</td>
<td>240346</td>
<td>99.73%</td>
<td>240346</td>
<td>99.73%</td>
<td>240346</td>
<td>99.73%</td>
<td>240346</td>
<td>99.73%</td>
<td>240346</td>
<td>99.73%</td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = 0.40:1

<table>
<thead>
<tr>
<th>Ad Distribution</th>
<th>3/1</th>
<th>3/1</th>
<th>3/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>310024</td>
<td>84.10%</td>
<td>310024</td>
<td>84.10%</td>
<td>310024</td>
<td>84.10%</td>
<td>310024</td>
<td>84.10%</td>
<td>310024</td>
<td>84.10%</td>
<td>310024</td>
<td>84.10%</td>
</tr>
<tr>
<td>OP</td>
<td>260722</td>
<td>84.10%</td>
<td>260722</td>
<td>84.10%</td>
<td>260722</td>
<td>84.10%</td>
<td>260722</td>
<td>84.10%</td>
<td>260722</td>
<td>84.10%</td>
<td>260722</td>
<td>84.10%</td>
</tr>
<tr>
<td>G</td>
<td>242459</td>
<td>99.69%</td>
<td>242459</td>
<td>99.69%</td>
<td>242459</td>
<td>99.69%</td>
<td>242459</td>
<td>99.69%</td>
<td>242459</td>
<td>99.69%</td>
<td>242459</td>
<td>99.69%</td>
</tr>
<tr>
<td>OP</td>
<td>225236</td>
<td>99.69%</td>
<td>225236</td>
<td>99.69%</td>
<td>225236</td>
<td>99.69%</td>
<td>225236</td>
<td>99.69%</td>
<td>225236</td>
<td>99.69%</td>
<td>225236</td>
<td>99.69%</td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = 0.35:1

<table>
<thead>
<tr>
<th>Ad Distribution</th>
<th>3/1</th>
<th>3/1</th>
<th>3/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>288319</td>
<td>85.20%</td>
<td>288319</td>
<td>85.20%</td>
<td>288319</td>
<td>85.20%</td>
<td>288319</td>
<td>85.20%</td>
<td>288319</td>
<td>85.20%</td>
<td>288319</td>
<td>85.20%</td>
</tr>
<tr>
<td>OP</td>
<td>257673</td>
<td>92.07%</td>
<td>257673</td>
<td>92.07%</td>
<td>257673</td>
<td>92.07%</td>
<td>257673</td>
<td>92.07%</td>
<td>257673</td>
<td>92.07%</td>
<td>257673</td>
<td>92.07%</td>
</tr>
<tr>
<td>G</td>
<td>220753</td>
<td>98.09%</td>
<td>220753</td>
<td>98.09%</td>
<td>220753</td>
<td>98.09%</td>
<td>220753</td>
<td>98.09%</td>
<td>220753</td>
<td>98.09%</td>
<td>220753</td>
<td>98.09%</td>
</tr>
<tr>
<td>OP</td>
<td>210459</td>
<td>99.76%</td>
<td>210459</td>
<td>99.76%</td>
<td>210459</td>
<td>99.76%</td>
<td>210459</td>
<td>99.76%</td>
<td>210459</td>
<td>99.76%</td>
<td>210459</td>
<td>99.76%</td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = 0.30:1

<table>
<thead>
<tr>
<th>Ad Distribution</th>
<th>3/1</th>
<th>3/1</th>
<th>3/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>1/1</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>206692</td>
<td>86.38%</td>
<td>206692</td>
<td>86.38%</td>
<td>206692</td>
<td>86.38%</td>
<td>206692</td>
<td>86.38%</td>
<td>206692</td>
<td>86.38%</td>
<td>206692</td>
<td>86.38%</td>
</tr>
<tr>
<td>OP</td>
<td>238707</td>
<td>85.21%</td>
<td>238707</td>
<td>85.21%</td>
<td>238707</td>
<td>85.21%</td>
<td>238707</td>
<td>85.21%</td>
<td>238707</td>
<td>85.21%</td>
<td>238707</td>
<td>85.21%</td>
</tr>
<tr>
<td>G</td>
<td>213766</td>
<td>98.05%</td>
<td>213766</td>
<td>98.05%</td>
<td>213766</td>
<td>98.05%</td>
<td>213766</td>
<td>98.05%</td>
<td>213766</td>
<td>98.05%</td>
<td>213766</td>
<td>98.05%</td>
</tr>
<tr>
<td>OP</td>
<td>194713</td>
<td>99.70%</td>
<td>194713</td>
<td>99.70%</td>
<td>194713</td>
<td>99.70%</td>
<td>194713</td>
<td>99.70%</td>
<td>194713</td>
<td>99.70%</td>
<td>194713</td>
<td>99.70%</td>
</tr>
</tbody>
</table>
### B.2 Service Levels

Value of Small ad to Large ad = 1:1

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>30</td>
<td>41%</td>
<td>49%</td>
<td>57%</td>
<td>60%</td>
<td>62%</td>
<td>0.8</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>30</td>
<td>51%</td>
<td>61%</td>
<td>71%</td>
<td>75%</td>
<td>74%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>30</td>
<td>61%</td>
<td>66%</td>
<td>69%</td>
<td>71%</td>
<td>72%</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>30</td>
<td>75%</td>
<td>81%</td>
<td>85%</td>
<td>87%</td>
<td>85%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>30</td>
<td>80%</td>
<td>81%</td>
<td>80%</td>
<td>81%</td>
<td>82%</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>30</td>
<td>93%</td>
<td>95%</td>
<td>96%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>30</td>
<td>93%</td>
<td>93%</td>
<td>90%</td>
<td>90%</td>
<td>89%</td>
<td>1.1</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>30</td>
<td>99%</td>
<td>100%</td>
<td>99%</td>
<td>98%</td>
<td>96%</td>
<td></td>
</tr>
</tbody>
</table>
### Value of Small ad to Large ad = .55:1

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>88%</td>
<td>90%</td>
<td>94%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>68%</td>
<td>71%</td>
<td>73%</td>
<td>72%</td>
<td>71%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>88%</td>
<td>90%</td>
<td>92%</td>
<td>95%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>69%</td>
<td>71%</td>
<td>75%</td>
<td>76%</td>
<td>74%</td>
<td>0.8</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>96%</td>
<td>97%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>80%</td>
<td>82%</td>
<td>84%</td>
<td>82%</td>
<td>80%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>96%</td>
<td>97%</td>
<td>97%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>81%</td>
<td>84%</td>
<td>86%</td>
<td>87%</td>
<td>84%</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>93%</td>
<td>93%</td>
<td>94%</td>
<td>90%</td>
<td>89%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>95%</td>
<td>95%</td>
<td>96%</td>
<td>95%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>99%</td>
<td>99%</td>
<td>98%</td>
<td>96%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>98%</td>
<td>96%</td>
<td>1.1</td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = .50:1

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>80%</td>
<td>81%</td>
<td>82%</td>
<td>92%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>80%</td>
<td>80%</td>
<td>79%</td>
<td>76%</td>
<td>74%</td>
<td>0.8</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>80%</td>
<td>80%</td>
<td>80%</td>
<td>88%</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>80%</td>
<td>80%</td>
<td>80%</td>
<td>78%</td>
<td>75%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>90%</td>
<td>90%</td>
<td>92%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>90%</td>
<td>90%</td>
<td>89%</td>
<td>86%</td>
<td>83%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>90%</td>
<td>90%</td>
<td>90%</td>
<td>96%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>90%</td>
<td>90%</td>
<td>90%</td>
<td>88%</td>
<td>84%</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>97%</td>
<td>97%</td>
<td>98%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>93%</td>
<td>91%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>95%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>98%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>96%</td>
<td>1.1</td>
</tr>
</tbody>
</table>
### Value of Small ad to Large ad = \(\frac{.45}{1}\)

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>72%</td>
<td>71%</td>
<td>69%</td>
<td>84%</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>92%</td>
<td>90%</td>
<td>87%</td>
<td>79%</td>
<td>75%</td>
<td>0.8</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>72%</td>
<td>71%</td>
<td>68%</td>
<td>83%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>92%</td>
<td>91%</td>
<td>87%</td>
<td>79%</td>
<td>75%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>85%</td>
<td>83%</td>
<td>80%</td>
<td>93%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>98%</td>
<td>97%</td>
<td>94%</td>
<td>88%</td>
<td>84%</td>
<td>0.9</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>85%</td>
<td>83%</td>
<td>79%</td>
<td>92%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>98%</td>
<td>97%</td>
<td>95%</td>
<td>89%</td>
<td>84%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>95%</td>
<td>95%</td>
<td>95%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>95%</td>
<td>95%</td>
<td>94%</td>
<td>99%</td>
<td>100%</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>95%</td>
<td>94%</td>
<td>93%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>95%</td>
<td>94%</td>
<td>93%</td>
<td>99%</td>
<td>100%</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = \(\frac{.40}{1}\)

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>66%</td>
<td>63%</td>
<td>58%</td>
<td>81%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>99%</td>
<td>98%</td>
<td>92%</td>
<td>80%</td>
<td>75%</td>
<td>0.8</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>66%</td>
<td>62%</td>
<td>57%</td>
<td>81%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>99%</td>
<td>98%</td>
<td>92%</td>
<td>80%</td>
<td>75%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>82%</td>
<td>80%</td>
<td>74%</td>
<td>92%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>89%</td>
<td>84%</td>
<td>0.9</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>82%</td>
<td>80%</td>
<td>74%</td>
<td>92%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>89%</td>
<td>84%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>95%</td>
<td>94%</td>
<td>93%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>95%</td>
<td>94%</td>
<td>93%</td>
<td>99%</td>
<td>100%</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>98%</td>
<td>96%</td>
<td>1.1</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>96%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Value of Small ad to Large ad = .35:1

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>66%</td>
<td>61%</td>
<td>57%</td>
<td>81%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>93%</td>
<td>80%</td>
<td>75%</td>
<td>0.8</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>66%</td>
<td>61%</td>
<td>57%</td>
<td>81%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>93%</td>
<td>80%</td>
<td>75%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>82%</td>
<td>80%</td>
<td>74%</td>
<td>92%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>89%</td>
<td>84%</td>
<td>0.9</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>82%</td>
<td>80%</td>
<td>74%</td>
<td>92%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>89%</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>95%</td>
<td>94%</td>
<td>93%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>95%</td>
<td>94%</td>
<td>93%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>96%</td>
<td>1.1</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>96%</td>
<td></td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = .30:1

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Ad length</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>15</td>
<td>66%</td>
<td>62%</td>
<td>55%</td>
<td>82%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>93%</td>
<td>79%</td>
<td>75%</td>
<td>0.8</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>66%</td>
<td>62%</td>
<td>55%</td>
<td>82%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>93%</td>
<td>79%</td>
<td>75%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>82%</td>
<td>80%</td>
<td>74%</td>
<td>92%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>89%</td>
<td>84%</td>
<td>0.9</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>82%</td>
<td>80%</td>
<td>74%</td>
<td>92%</td>
<td>99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>89%</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>95%</td>
<td>94%</td>
<td>93%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>95%</td>
<td>94%</td>
<td>93%</td>
<td>99%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>96%</td>
<td>1.1</td>
</tr>
<tr>
<td>OP</td>
<td>15</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>96%</td>
<td></td>
</tr>
</tbody>
</table>
### B.3 Utilization of breaks

#### Value of Small ad to Large ad = 1:1

<table>
<thead>
<tr>
<th>Ad dist:</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>96%</td>
<td>93%</td>
<td>89%</td>
<td>85%</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>0.8</td>
</tr>
<tr>
<td>G</td>
<td>94%</td>
<td>92%</td>
<td>88%</td>
<td>85%</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>92%</td>
<td>90%</td>
<td>87%</td>
<td>84%</td>
<td>83%</td>
<td>1</td>
</tr>
<tr>
<td>OP</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>88%</td>
<td>87%</td>
<td>85%</td>
<td>83%</td>
<td>82%</td>
<td>1.1</td>
</tr>
<tr>
<td>OP</td>
<td>91%</td>
<td>90%</td>
<td>91%</td>
<td>89%</td>
<td>88%</td>
<td></td>
</tr>
</tbody>
</table>

#### Value of Small ad to Large ad = .55:1

<table>
<thead>
<tr>
<th>Ad dist:</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>93%</td>
<td>0.8</td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>100%</td>
<td>99%</td>
<td>99%</td>
<td>95%</td>
<td>91%</td>
<td>0.9</td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>97%</td>
<td>97%</td>
<td>95%</td>
<td>93%</td>
<td>90%</td>
<td>1</td>
</tr>
<tr>
<td>OP</td>
<td>97%</td>
<td>98%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>91%</td>
<td>90%</td>
<td>90%</td>
<td>88%</td>
<td>87%</td>
<td>1.1</td>
</tr>
<tr>
<td>OP</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td></td>
</tr>
</tbody>
</table>

#### Value of Small ad to Large ad = .50:1

<table>
<thead>
<tr>
<th>Ad dist:</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>96%</td>
<td>0.8</td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
<td>94%</td>
<td>0.9</td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>95%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>OP</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td>1.1</td>
</tr>
<tr>
<td>OP</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td></td>
</tr>
</tbody>
</table>
### Value of Small ad to Large ad = .45:1

<table>
<thead>
<tr>
<th>Ad dist:</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>0.8</td>
</tr>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>98%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>98%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>91%</td>
<td>90%</td>
<td>91%</td>
<td>89%</td>
<td>88%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>91%</td>
<td>90%</td>
<td>91%</td>
<td>89%</td>
<td>88%</td>
<td>1.1</td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = .40:1

<table>
<thead>
<tr>
<th>Ad dist:</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>0.8</td>
</tr>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>91%</td>
<td>90%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>91%</td>
<td>90%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td>1.1</td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = .35:1

<table>
<thead>
<tr>
<th>Ad dist:</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>0.8</td>
</tr>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>97%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td>1.1</td>
</tr>
</tbody>
</table>

### Value of Small ad to Large ad = .30:1

<table>
<thead>
<tr>
<th>Ad dist:</th>
<th>3:1</th>
<th>2:1</th>
<th>1:1</th>
<th>1:2</th>
<th>1:3</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
<td>0.8</td>
</tr>
<tr>
<td>G</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>99%</td>
<td>95%</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>98%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>98%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
<td>92%</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
<td>1.1</td>
</tr>
</tbody>
</table>