Revenue Maximization with Dynamic Advertisement Scheduling in Breaks of Random Lengths

Ajay S.A. / ajaysa@smu.edu.sg

October 4, 2011

Elevator Pitch

When scheduling ads for breaks of unknown lengths,

- be flexible (greedy $>$ knapsack)
- with an eye to the future (forward looking $>$ myopic)

Outline

Introduction

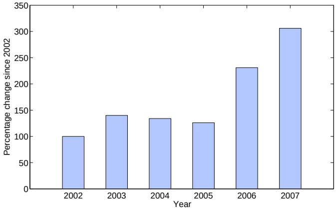


Figure: Growth in investment in cricket broadcasts over 2002 - 2007

Cricket is big business:

- Ad volumes in cricket show a growth of more than three times from 2002 - 2007 (TAM Media research)
- Cricket tournaments such as IPL T-20 earn approx. US $150M for the broadcaster, with per-second rate approx. US $1100 (WSJ 2010)

Introduction

Figure: A game of cricket

Challenge in live broadcasting: stochastic break lengths, mainly:

- Breaks between overs
- Breaks after a wicket has fallen

Literature Review

Random Yield / Capacity

Ciarallo et al. (1994), Yano and Lee(1995), Wang and Gerchak (1996)

Stochastic Knapsack

Kleywegt and Papastavrou (1996), Witchakul, Ayudhya, Charnsethikul(2008)

Stochastic Cutting Stock / Recourse

Gilmore and Gomory (1961), Scull (1981), Birge and Louveaux (1997)

Advertisement Scheduling

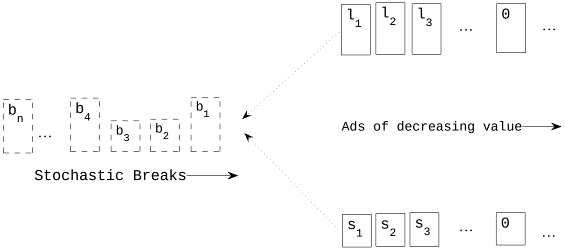
Bollapragada and Garbiras(2003), Kimms and Muller-Bungart(2007)

Approaches

Two approaches:

1. Analytical approach: Find the optimal solution with minimal constraints
2. Heuristic approach: Find a near optimal solution with constraints

Model



Assumptions

- Breaks $\{b_1, b_2, \ldots, b_n\}$ occur sequentially, $1 \leq n \leq N$
- Ads of two types, of size S and $L = 2S$, of infinite number and decreasing value
- Ads that are not fully aired do not earn any revenue

Cases

Schedulers either know the duration of the current break, or they do not.

We discuss Optimal Policy for each scenario under four cases:

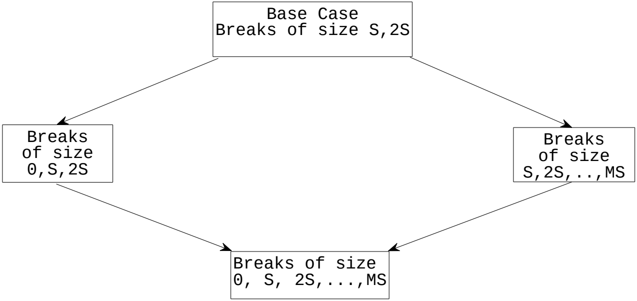


Figure: Four cases

Known Break Size

- The scheduler knows the size of the break which he is currently scheduling, but the sizes of the subsequent breaks are unknown
- Corresponds to cases where the on-field director can predict the duration of the break

Base Case : Known Break Size

If the $b_1 = S$, then the Optimal Policy is to schedule s_1.

Theorem

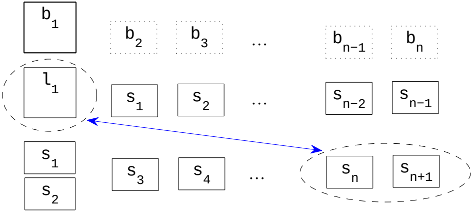
When n breaks remain, and $b_1 = L$, the optimal policy is to:

- *select l_1 if $l_1 \geq s_n + s_{n+1}$*
- *select (s_1, s_2) otherwise*

Base Case : Known Break Size

Intuition The Optimal Policy decides on a minimum threshold for l_1 by minimizing the loss incurred if breaks $b_2, \ldots, b_n$ are all small, since

$$l_1 + \sum_{i=1}^{n-1} s_i \geq \sum_{i=1}^{n+1} s_i$$



Stochastic number of breaks : Known Break Size

- Breaks of size zero can occur with probability p_0
- Breaks of size S and L occur with probabilities p_1 and p_2 respectively
- $p_0 + p_1 + p_2 = 1$

Theorem

When n breaks remain, $p_0 > 0$, and $b_1 = L$, the optimal policy is to:

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

Stochastic number of breaks : Known Break Size

Intuition

- $\left[\binom{n-1}{i} p_0^{(n-1)-i} p_1^i\right]$ is the probability that of the remaining $n-1$ breaks, i breaks are of size S and the others are of size zero.
- The Optimal Policy shifts the threshold for l_1 based on the probability distribution; as p_0 increases, l_1 is compared to a value closer to $(s_1 + s_2)$ than $(s_n + s_{n+1})$

Multiple break sizes : Known Break Size

- Maximum break size possible is MS, and size of b_1 is mS, $1 \leq m \leq M$
- Breaks of size i can occur with probability p_i, $\sum_{i=1}^{M} p_i = 1$

Theorem

If $b_1 = mS$ and n breaks remain, then the Optimal Policy is to select $(l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda})$, where λ is the largest index such that $2\lambda \leq m$ and:

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

Multiple break sizes : Known Break Size

Intuition

- As index i decreases, l_i becomes more likely to be selected in the $\lfloor \frac{m}{2} \rfloor$ breaks, since its value increases and the threshold $s_{m-2i+n} + s_{m-2i+n+1}$ decreases.
- Compared to the Greedy Policy, which looks at all possible combination of ads to select the highest earning combination, the Optimal Policy is less complex and scales well

Stochastic number of breaks of multiple sizes : Known Break Size

- Maximum break size possible is MS, and size of b_1 is mS, $1 \leq m \leq M$
- Breaks of size i can occur with probability p_i, $\sum_{i=0}^{M} p_i = 1$

Theorem

If $b_1 = mS$, $p_0 > 0$, and n breaks remain, then the Optimal Policy is to select $(l_1, \ldots, l_\lambda, s_1, \ldots, s_{m-2\lambda})$, where λ is the largest index such that $2\lambda \leq m$ and:

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

Stochastic number of breaks of multiple sizes : Known Break Size

Intuition The Optimal Policy for stochastic number of breaks of multiple sizes is a combination of policies for stochastic number of breaks and breaks of multiple sizes.

Unknown Break Size

- Each break begins without the schedulers knowing its duration
- Schedulers only use the break size distribution to decide on ad selection

Base Case : Unknown Break Size

- Breaks can be of size S with probability p, and size L with probability $1-p$

Theorem

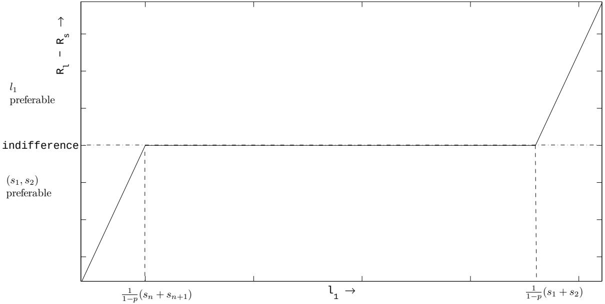
When n breaks remain, and the break size is unknown, the Optimal Policy is to:

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*

Base Case : Unknown Break Size

Intuition

- The threshold for selecting l_1 is based on myopic comparison of expected values since each break is IID
- Region of indifference (Case 3) exists because both l_1 and (s_1, s_2) are guaranteed to be scheduled at some point during the match



Stochastic number of breaks : Unknown Break Size

- Breaks of size zero can occur with probability p_0
- Breaks of size S and L occur with probabilities p_1 and p_2 respectively
- $p_0 + p_1 + p_2 = 1$

Theorem

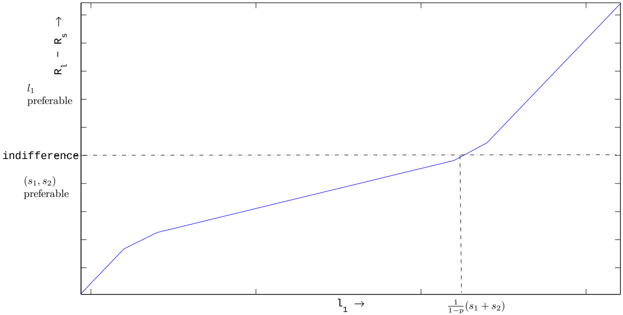
When $k = n$, $p_0 \geq 0$, and break sizes are unknown, the Optimal Policy is to:

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*

Stochastic number of breaks : Unknown Break Size

Intuition

- Region of indifference no longer exists - there is no guarantee that both l_1 and (s_1, s_2) will be scheduled since subsequent breaks need not be of size S or L



Larger breaks of unknown size

- With larger breaks of unknown size, the problem becomes intractable!
- The optimal policy should give the ideal *permutation* of ads to be scheduled
- Number of permutations exponentially increases with break size

M	Number of ad schedules
4	5
5	8
6	13
7	21
$\vdots$	$\vdots$

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

Heuristics

Consider:

- three ad lengths
- multiple break lengths
- break length known when scheduling
- need to satisfy service level guarantees, or be penalized

Parameters

Parameter	Value
Ad lengths	10, 20, 30
Number of ads of each type	$\sim \mathcal{U}(1, 3)$
Break lengths	$\sim \mathcal{U}(10, 70)$
Mean Break Length	40
Number of advertisers	20
Number of breaks / match	50
Number of matches (trials)	100
Revenue / Second / Advertiser	$\sim \mathcal{U}(3500, 5500)$
Target service level	80%
Service level penalty	$1000 per sec

Description of heuristics

Greedy Algorithm (foolish)

For each break, while there is time remaining,

- Find max ad length that can be fit in time remaining
- Find ad with max revenue of that length s.t:
  - It is available in the inventory
  - It has not been scheduled in this break
- If no ad found of this length, go to lower length and repeat until one is found or none available

Certainty Equivalent

- We solve an IP and generate buckets of ads before the start of the match
- Bundle size based on expected break size
- Includes penalty for not meeting service level

Description of heuristics

Modified Certainty Equivalent

- IP same as for CE, but bucket size distribution based on break size distribution
- Fit the scheduled bins of closest matching into the breaks once break length is known

Dynamic Modified Certainty Equivalent

- Buckets are generated dynamically, whenever we run out of buckets
- IP same as for MCE, but number of buckets matches number of breaks left

Perfect Information

Buckets are generated at the start but break sizes are known at the time of generation

Performance

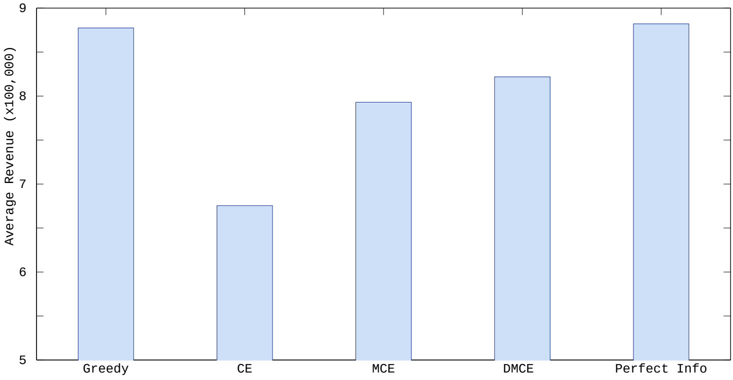


Figure: Performance of heuristics

- Performance of greedy is better than bundling strategies!
- Adaptive sub-modular systems (Golovin / Krause (2010))
  - Greedy within $1 - \frac{1}{e}$ of perfect information!

Numerical Results

Aims

- Compare the Optimal Policy to the Greedy Policy, to find conditions where the Optimal Policy most outperforms the Greedy Policy.
- Study the impact of service level commitments and the impact of uncertainty on the performance of the Optimal Policy.

Deterministic number of breaks

Parameters

- Number of breaks: 50
- Break size (sec) : $\sim \mathcal{B}(10, 20)$
- Number of large ads : 50
- Number of small ads : 100
- Large ads values (\$) : $\sim \mathcal{U}(1000, 1200)$
- Small ads values (\$) : $\sim \mathcal{U}(200, 1000)$

Explanation

- T-20 cricket has 50 breaks on average: 40 overs + 10 wickets
- Values of large and small ads reflect real world data
- Air time sold is equally split between small and large ads

Revenues and Service Levels

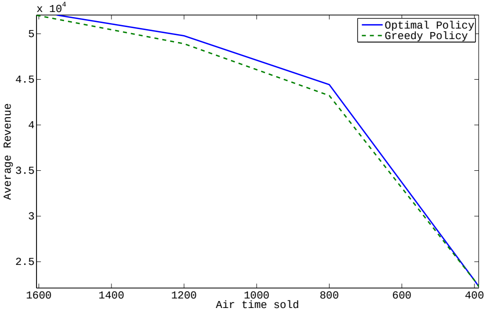
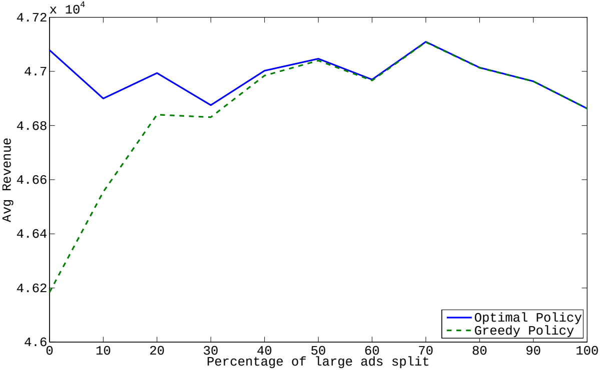


Figure: Average total revenue versus air time sold

- Greedy Policy prefers to schedule small ads more than Optimal Policy
- At $\approx$ 800s (service level is $\sim$ 80%), the Greedy Policy lets almost one short break unused since it runs out of short ads to schedule

Value of Flexibility



- Randomly selected large ads are split into two equally valued small ads, and performance is observed as the ratio of large and small ads changes
- Optimal Policy earns more than Greedy Policy when ratio is ≈ 1:1
- Greedy Policy approaches Optimal Policy as ratio of small ads to large ads increases; Optimal Policy does not show significant deviation

Impact of Uncertainty

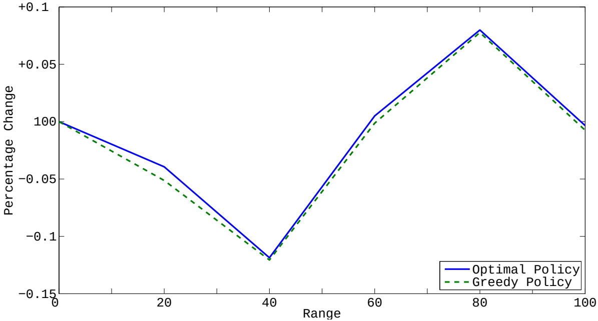
Parameters

- Break lengths : $\sim \mathcal{U}(\mu - \delta, \mu + \delta)$ where $\mu = 60$, $0 \leq \delta \leq 50$
- Number of large ads : 100
- Number of small ads : 200

Explanation

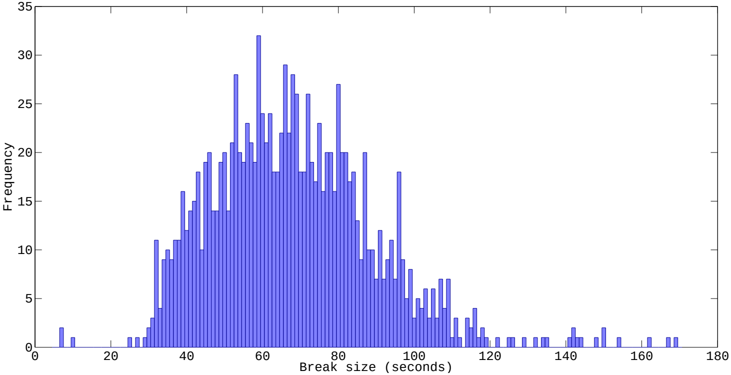
- The performance of the Optimal Policy is observed when break sizes have a Uniform Distribution with mean 60 and the support is varied
- Expected air time has increased $\implies$ increase in the size of our inventory

Impact of Uncertainty



- Neither Optimal Policy nor Greedy Policy are significantly affected by uncertainty, since large breaks and small breaks occur equally on average

Break Sizes



Mean	68.4 sec
Mode	69 sec
Standard Deviation	23 sec
Count	1231 breaks

Ad Sales

- Linear revenues:
  - Mean = \$4,120
  - Standard deviation : \$533
- Ad format: 10, 15, 20 and 30 secs