High Dimensional Revenue Management

by

Dragos Florin Ciocan

B.A., Applied Mathematics, Harvard College (2007)

Submitted to the Alfred P. Sloan School of Management in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Management

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

Signature redacted

Author ....

Alfred P. Sloan School of Management
August 8, 2014

Signature redacted

Certified by..

Vivek F. Farias
Associate Professor and Robert N. Noyce Career Development Professor
Thesis Supervisor

Signature redacted

Accepted by ....

Ezra W. Zuckerman
Director, Sloan School of Management PhD Program

High Dimensional Revenue Management

by

Dragos Florin Ciocan

Submitted to the Alfred P. Sloan School of Management
on August 8, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Management

Abstract

We present potential solutions to several problems that arise in making revenue management (RM) practical for online advertising and related modern applications. Principally, RM solutions for these problems must contend with (i) highly volatile demand processes that are hard to forecast, and (ii) massive scale that makes even basic optimization problems challenging. Our solutions to these problems are interesting in their own right in the areas of stochastic optimization, high dimensional learning and distributed optimization.

In the first part of the thesis, we propose a model predictive control approach to combat volatile demand. This approach is conceptually simple, uses available demand data in a natural way, and, most importantly, can be shown to generate significant revenue advantages on real-world data from ad networks. Under mild restrictions, we prove that our algorithm achieves uniform relative performance guarantees vis-a-vis a clairvoyant in the face of arbitrary volatility, while simultaneously being optimal in the event that volatility is negligible. This is the first result of its kind for model predictive control.

While our approach above is effective at hedging demand shocks that occur over "large" time horizons, it relies on the ability to estimate snapshots of the prevailing demand distribution over "short" time horizons. The second part of the thesis deals with learning the extremely high dimensional demand distributions that are typical in display advertising applications. This work exploits the special structure of the display advertising version of the NRM problem to achieve a sample complexity that scales gracefully in the dimensions of the problem.

The third part of the thesis focuses on the problem of solving terabyte sized LPs on an hourly basis given a distributed computational infrastructure; solving these massive LPs is the computational primitive required to make our model predictive control approach practical. Here we design a linear optimization algorithm that fits a paradigm for distributed computation referred to as 'Map-Reduce'. An implementation of our solver in a shared memory environment where we can benchmark against solvers such as CPLEX shows that the algorithm outperforms those solvers on the types of LPs that an ad network would have to solve in practice.

Thesis Supervisor: Vivek F. Farias
Title: Associate Professor and Robert N. Noyce Career Development Professor

Acknowledgments

As I wrap up my PhD studies at MIT and submit this thesis, there are a few people who have supported me in getting to this point that I would like to express my gratitude to. Firstly, there is my advisor Vivek Farias. It has been a privilege to work with him over the years and I have always found his intellectual curiosity and seemingly boundless energy inspirational. Vivek has been very supportive throughout the highs and lows of my time at MIT, and has set a high bar as an advisor for when I will take on students of my own.

I also owe a great deal of gratitude to the other member of my thesis committee, Professors Steve Graves and Devavrat Shah. It was a pleasure to have them on my committee and to be able to get their feedback and advice. I have also had the opportunity to learn and share ideas with some of the extraordinary faculty at MIT, particularly those in the OM and OR groups at Sloan.

Parts of this thesis stem from work done in collaboration with MohammadHossein Bateni and Vahab Mirrokni while I was an intern at Google Research – they have taught me a lot about the nuances of the online advertising industry and have a opened up many directions for my future research. I would also like to acknowledge Hamid Nazerzadeh who introduced me to some of the problems that I have worked on during my PhD, as well as Mike Barrett for providing the data for some of my computational experiments.

Going back in time a little further, I owe a great deal of gratitude to Professor Salil Vadhan at Harvard, who advised me on my undergraduate thesis. Had he not accepted to supervise me when I walked into his office with an interest in writing a thesis on zero knowledge proofs, I would have never thought about embarking on an academic career in the first place.

One of the greatest things about MIT are the fantastic peers who end up being your friends. Thank you Gonzalo, Andre, Fernanda, Yiwei, Adam, Leon, Nikos, Dan, Srikanth, Guillaume, Spyros, Theja and all the other colleagues I have had at Sloan, the ORC, or LIDS. It has been a pleasure to share a great deal of my time in Boston with such great people and I hope we'll get to see each other in person as often as possible.

Lastly, I have to thank my parents Vladi and Carmen. They have always supported me, and I know it has been difficult to have me far away from Bucharest for so many years. I'll make a non-binding promise to call them more often in the future.

Contents

List of Figures

List of Tables

Introduction

This thesis tackles several challenges that arise in high dimensional network revenue management (NRM) problems. While NRM is a core problem in revenue management and has been extensively researched in the last two decades, there has been relatively less effort towards making classical NRM approaches practical at a scale where the number of customer types and resources run into the billions, as is the case with modern applications such as online advertising. Here, we provide potential solutions to three of the challenges one must contend with at this scale:

1. Absence of demand forecasts: in the long term, demand is highly uncertain and difficult to forecast. A good scheme must "hedge" the impact of this uncertainty.

2. Measuring current demand: even in the short term, customer demand might be difficult to estimate since the number of demand observations an algorithm may have access to will typically be much smaller than the number of demand types.

3. Computational tractability: classical NRM models typically rely on basic optimization tools such as linear programming. However, existing linear optimization packages are not designed to scale to the dimensions encountered in applications such as online advertising. A practical scheme must contend with these scalability issues.

Part I of the thesis develops a model predictive control scheme to mitigate the effect of demand volatility for a broad class of resource allocation problems with uncertain demand, including online advertising and airline yield management. While the classical Deterministic Linear Program (DLP) framework for NRM assumes deterministic known demand arrival rates, we consider a setting where these rates are unknown (or, put another way, impossible to forecast). Under fairly mild assumptions on the nature of the stochastic processes driving these demand rates, we provide an algorithm which achieves constant factor guarantees versus the offline optimal even in the presence of arbitrarily large volatility, while simultaneously being optimal when volatility is low. The approach relies on periodically re-solving an LP requiring only the currently observable arrival rates. In a study on real data from an ad network, we show that the scheme yields nearly optimal performance.

Surprisingly, in Chapter 2 we are also able to extend our scheme to Generalized Second Price ad allocation, which can be interpreted as a variation of the classical network revenue management problem where rewards are endogeneous to the dynamics of the system. This extension makes use of a crucial balancing property of our allocation scheme, which guarantees that resources are exhausted at a constant pace over the time horizon of the problem.

In Part II, we deal with the problem of learning high dimensional customer demand distributions. The model we consider here is one where customers arrive iid from a fixed but unknown multinomial distribution on an exponentially large set of customer types. The question we ask is: can we use a much smaller number of customer arrival observations to learn a control which is $1 - \epsilon$ optimal versus a clairvoyant? In Chapter 3 we identify a class of NRM problems, inspired by display advertising, which have enough structure that this learning problem becomes tractable: for this class of problems we achieve a sample complexity that depends logarithmically on size of the support of the demand distribution. This sampling algorithm ties into our earlier results from Part I – in particular, at the "global" timescale over which customer arrival rates are non-stationary, the prescription is to periodically re-solve an LP in which current rates are plugged in. We build up on this by addressing the issue of how to efficiently learn these current rates over the "local" timescale of one LP re-solve.

Finally, Part III focuses on practically solving linear programs at the terabyte scales that occur in online advertising applications. Here we look for an algorithm which is amenable to implementations in the decentralized computational infrastructures that are typically employed by ad networks. More specifically, our solution fits the Map-Reduce paradigm for distributed computation. Our algorithm proceeds by solving a sequence of relaxations of the original linear program; the key computational step to solving one such relaxation is a large sort, an operation that modern distributed frameworks like Map-Reduce are optimized for. Our implementation in a shared memory environment where we can benchmark against solvers such as CPLEX shows that the algorithm outperforms those solvers on the types of LPs that arise in the online advertising context. Additionally, we show our algorithm outperforms other distributed approaches for solving a broader class of LPs known as packing problems by an order of magnitude on average.

Part I

Model Predictive Control for Dynamic Resource Allocation

Chapter 1

Model Predictive Control

1.1 Introduction.

In this chapter we consider an archetypal dynamic allocation problem that captures a swathe of disparate applications in revenue management and e-commerce. Informally, the class of problems we consider can be described as follows: we are given a bipartite graph consisting of a set of I sources and A sinks, together with E edges from the sources to the sinks. Every source node i receives 'demand' over time. The rate at which demand arrives is described by a general stochastic process. An allocation is an assignment of the demand arriving at each of the source nodes to the sinks they are connected with. This allocation may change over time; hence the dynamic moniker. For every unit of demand allocated along edge e (connecting some source $i(e)$ to some sink $a(e)$), we receive a reward of p_e. In addition, this unit allocation will consume varying quantities of each of K distinct resources described by a vector $A_e \in \mathbb{R}_+^K$. We begin with an initial allocation of each of the K resources and must allocate demand over time in a manner that maximizes revenues while consuming no more than the initial allocation of resources. The key source of uncertainty in this model comes from the stochastic demand rate processes; in practical settings even specifying such a process is a potentially non-trivial task.

The abstract allocation model we have described above can capture a number of applications of broad interest. Two examples that are of particular interest in revenue management (RM) are:

- The Network Revenue Management (NRM) problem: This is a generic high-dimensional allocation problem encountered in industries ranging from the hospitality industry to the airline industry and is effectively a cornerstone RM model. For concreteness, consider the problem faced by an airline that sells a variety of itineraries over a network of cities over time. Each itinerary requires seats on different legs of the network and generates different revenues. The airline must sell itineraries over time in a manner that respects seat capacity and maximizes revenues. Volatility in demand and the high-dimensional nature of the allocation problem together make it challenging.

- Online Ad Display problems: An ad network serves as an intermediary between publishers (sources of traffic or demand) and advertisers. Via a variety of contractual agreements it agrees to display ads from specific advertisers to compatible traffic from publishers (where the notion of compatible might correspond to the publisher's entity among other things). Advertising contracts might specify that a given ad is displayed up to a certain number of times to compatible traffic. Alternatively, by specifying a dollar rate for the display of a specific ad to a specific type of demand, the advertiser might commit to spending up to a certain budget on advertising. The ad exchange may, in turn, seek to exhaust as much of this budget as possible while maximizing revenues. Again, volatility in traffic across publishers and over time, combined with the sheer number of potential alternatives for the allocation of a given unit of traffic, makes this a non-trivial allocation problem.

We defer a rigorous problem definition and a precise explanation of how the above problems and others fit into our framework to Section 3.2. For now, we simply note that the above problems are considered difficult primarily due to uncertainty in demand and the high-dimensional nature of the resource allocation problem at hand. As such, there are distinct bodies of literature devoted to the above problems and variants thereof. Rigorous algorithmic approaches appear to fall roughly into two categories at diametrically opposite ends of the spectrum on demand assumptions. On the one hand, by assuming that the demand processes admit only 'small' shocks so that uncertainty in total demand at each source is small relative to its magnitude[1], one may resort to solving simple 'offline' versions of the allocation problem at hand. These solutions yield, via an appeal to a law of large numbers argument, an essentially optimal solution to the actual allocation problem. This has been the dominant modeling approach in the majority of RM applications. On the other hand, the assumption of 'small' demand shocks is clearly an idealization, so that a distinct approach to such problems has been to assume that demand is adversarial. One then seeks to design online allocation schemes that compete effectively with the adversary generating demand. This has been the dominant modeling approach for many of the e-commerce related allocation problems alluded to above. While such online schemes are typically quite simple, their design and analysis is typically fairly brittle to model assumptions.

In reality one typically faces a world that is somewhere in between the assumptions above: whereas assuming that demand is effectively deterministic (by assuming small shocks) is potentially unrealistic, the assumption of adversarial demand is itself potentially conservative and unrealistic. In particular, in many instances of the applications discussed above, one generates copious amounts of historical data on demand over time that in principle might be used to construct useful forecast models. Unfortunately, such forecast models are far from easy to construct in practice – challenges include judging the historical relevance of data, learning factors that serve as useful predictors, and of course, problem scale. Moreover, even assuming access to such a forecast model, the resulting dynamic optimization problem remains high-dimensional and intractable. The present paper tackles this middle ground on demand assumptions and provides solutions that attempt to address these challenges.

[1]This is analogous to assuming a deterministic demand rate process in the general model we will consider.

1.1.1 Contributions.

The present work posits a simple new approach to the general class of allocation problems above that relies on a combination of re-optimization and 'robust' forecasting; importantly, the approach does not require that the demand rate process be specified. The approach is pragmatic and at the same time admits attractive theoretical performance guarantees. In greater detail, we make the following contributions:

1. A Simple Allocation Scheme: We design an allocation scheme that relies on frequent re-optimization using suitably updated forecasts. In particular, at discrete points in time, one uses demand realized up to that point in time to construct a forecast for future demand in a precisely specified fashion. Assuming these forecasts to be exact, one then solves a simple linear optimization problem that prescribes an allocation of demand from sources to sinks. This allocation rule is followed until the next opportunity to re-optimize. The scheme is entirely mechanical in that it can be applied to any dynamic allocation problem within our framework without any instance specific analysis.

2. Uniform Worst Case Guarantees: Assuming that the demand rate process faced lies in a certain broad class of stochastic processes [2], we show that our allocation scheme yields expected revenues that are within a constant factor of expected revenues under a certain super-optimal policy. The value of this constant is either 0.342 or 0.2 depending on the specific assumptions we make on the family of demand processes. These worst case results are remarkable – they hold for *arbitrarily volatile* demand processes and illustrate that the proposed scheme is robust across a broad class of demand processes while being oblivious to the specification of the process. Our performance analysis overcomes the technical hurdle of analyzing the impact of basis changes in certain math programs that underlie our allocation scheme; this is the primary hurdle in analyzing allocation schemes that rely on re-optimization.

[2] We essentially allow for multi-variate Gaussian processes with continuous sample paths. We require that the volatility of any variate be a concave increasing function of time; we show that these requirements may be viewed as natural in the context of any stochastic forecast model.

3. Parametric Guarantees: In addition to uniform worst case guarantees, we present performance guarantees that reveal that as the volatility of the underlying demand process shrinks, our allocation scheme approaches optimality. Together with the worst case guarantees, this allows the following interpretation: the scheme we propose is essentially optimal if available forecasts are accurate, but otherwise robust to forecast inaccuracies.

4. Computational Evidence: We present computational experiments on both synthetic problem instances as well as a real-world example of an Ad Display problem using demand data from a mobile ad-network. In both the synthetic and real-world instances, the proposed scheme is seen to provide performance levels that are typically well within 90% of an upper bound constructed by assuming that demand realizations were available a-priori.

While our literature review will extensively place these contributions in the context of the extant work on dynamic allocation problems, we end this discussion with a brief statement of the relative merits: The nature of the assumptions made in traditional RM modeling (that of 'small' demand shocks) translates in our model to a demand rate process that is deterministic; the assumption of a *stochastic* demand process with no restrictions on volatility allows one to model (and address) large shocks in demand. Conversely, if one were to adopt an adversarial view of demand within our model, the nature of an optimal online scheme varies considerably across specializations of our model as do the corresponding competitive ratios. Loosely speaking, optimal competitive ratios range from a constant (in certain variants of the Ad Display problem) to scaling inversely with the log of the number of demand types (in the case of a general packing problem such as the NRM problem); the corresponding allocation schemes are quite distinct. Moreover, one typically loses the notion of what it means to have an 'accurate' forecast model in such a setting. In contrast, we present a unified approach based on re-optimization and forecast updates to what is apparently a broad range of problems and establish that it performs well in an expected sense assuming a broad generative family of demand processes.

1.1.2 Literature Review.

By virtue of its generality, our dynamic allocation model bears comparison to a very broad class of models. As such we organize our review of relevant literature around generic algorithmic approaches to comparable models. Given the diversity of comparable dynamic allocation models, this review is by necessity incomplete and biased.

Online Algorithms: The dynamic allocation problems we study are quite similar in spirit to (multi-dimensional) online packing problems. A powerful tool in the design of schemes for such problems is the primal-dual schema. Under an assumption of entirely adversarial demand it is possible to use this schema to design online algorithms that bear a competitive ratio on the order of the logarithm of the number of item types (i.e. $O(\log I)$ in the context of our model); see Buchbinder and Naor (2009). Under the so-called random permutation model, where an adversary selects the number of arriving items but nature then permutes these items uniformly at random, substantially stronger guarantees are possible – in particular, it is possible to provide online algorithms that constitute polynomial time approximation schemes under this model for a variety of packing problems in appropriate regimes. In particular, this was pointed out by Kleinberg (2005) in the context of a secretary problem. More recently, Agrawal et al. (2014) developed an online PTAS for a general multi-dimensional packing problem under this model. The exchangeable nature of the demand distribution under this adversarial model is what effectively drives these results; unfortunately this sort of model does not appear appropriate for demand processes that are inherently non-stationary.

There has also been a good amount of work on online algorithms for some of the specific applications we consider. In particular, for the so-called AdWords problems (a specialization of the Ad Display problem we discuss), Mehta et al. (2005), design an online $1 - 1/e$-competitive algorithm. Mahdian et al. (2009) work with the same Adwords problem, but allow for the existence of forecasts of demand (keyword arrivals). They not only obtain a constant factor guarantee versus worst case (adversarial) inputs, but also a constant factor guarantee versus the revenues that would be achieved if the forecast was perfect. The analysis in both papers assumes individual bids are small compared to the total budgets. In the case of the NRM problem, Ball and Queyranne (2009) consider a simplified version of the NRM problem, namely a setting in which the airline operates multiple itineraries on a *single* leg, i.e. $K = 1$ and show that a competitive ratio of 2 is achievable and optimal.

Many of the online schemes we have discussed are inherently conservative which leads to hesitation in their adoption in practice.

Approximate Dynamic Programming (ADP): Given a model of demand, one may in principle solve the sort of dynamic allocation problems we study via dynamic programming. Of course, this is untenable in practice due to the cure of dimensionality and one approach of contending with this issue is the design of appropriate ADP schemes. For instance, Bertsimas and Demir (2002) solve integer multidimensional knapsack problems via ADP. ADP heuristics have also been developed for the NRM problem. In particular, see Adelman (2007), Farias and Van Roy (2007) and Zhang and Adelman (2009). While with careful tailoring to the problem in question, these heuristics often exhibit excellent practical performance, (absolute) theoretical performance guarantees are difficult to come by in general. Moreover, the 'tailoring' required is frequently non-trivial – one needs to provide these algorithms with good 'approximation architectures' for the problem at hand.

Fluid Models: Fluid models arise essentially from considering allocation problems with their (stochastic) demand processes replaced by 'fluid' arrival processes with deterministic rates matching those of the stochastic demand process; of course, these rates have to be available a-priori. Solving these models is typically substantially easier than the original stochastic problem and the resulting solutions can work well in the original stochastic problem. Briefly, one can expect good performance provided the deviation in cumulative demand relative to its mean is 'small' and this typically requires a suitable scaling of the original problem. The model we study is, in contrast, a *stochastic* fluid model where the rates of the fluids in question are stochastic processes as opposed to being deterministic and known a-priori. Said another way, our work can be viewed as taking the fluid model approach *without* a-priori information on the demand rate process which is itself stochastic.

A typical area of application is in the analysis of control policies for queuing networks; see for instance Bramson (1998). A second area that has found applications for these tools is revenue management. For instance, the seminal work of Gallego and van Ryzin (1997) essentially posits a fluid model for the NRM problem and then proceeds to show that solutions derived from this model work well assuming the scale of demand and capacity grow large simultaneously. It is worth re-iterating that this again requires one know the *rate* of the underlying demand process a-priori and in that sense such a model is unable to capture 'large' shocks in demand. There is a vast literature preceding this paper, two of which do make an attempt to capture large shocks in demand – the first is a paper by Akan and Ata (2009) that considers a stochastic fluid model for NRM. Their model is very closely related to the one studied here. The authors provide a remarkable characterization of optimal policies and show how to compute these optimal policies assuming the demand process is described by a diffusion. Unfortunately, this still requires that one specify the diffusions a-priori. A second paper by Chen and Farias (2013) that is perhaps most closely related to the present work studies a *one* dimensional allocation problem (with a somewhat distinct control/ reward mechanism) under similar assumptions as those studied here. We present performance guarantees relative to a multivariate version of the family of demand processes studied in that work.

Re-Optimization/ Model Predictive Control: The area of model predictive control essentially prescribes solving hard stochastic control problems by posing fluid models and then *re-solving* these fluid models with suitable updates as uncertainty reveals itself. Our approach falls squarely within this philosophy. For a survey of model predictive control literature, see Bemporad (2006).

The MPC approach has been used (under different names) in a variety of settings. One prime example is in scheduling for queuing networks; for instance, the seminal work of Chen and Yao (1993) can be viewed in this light as can the celebrated Max Weight scheduling policy (see Shah and Wischik (2010)). More recently, this philosophy has also found application in revenue management, and in particular, for the NRM problem. In that context, Maglaras and Meissner (2006) pointed out that repeatedly re-solving the fluid model linear program prescribed by Gallego and van Ryzin (1997) yields an optimal allocation in the fluid scale. Subsequent work has shown that this re-optimization can play a substantial role in accelerating the rate at which such allocations approach optimality as the problem is scaled. In particular, Reiman and Wang (2008) show that a single re-optimization at a carefully chosen point can result in an additive performance loss that grows like the square-root of the problem scale. Recently, Jasin and Kumar (2012) provided an extremely elegant demonstration of the fact that with repeated re-optimization (that can be at uniformly spaced intervals) the additive performance loss is *independent* of the problem scale. This work nicely illustrates the impact of re-optimization in combating 'small' shocks in demand (the demand rate is *known* in all of the above papers). The present paper can be seen as complementing that work by showing that re-optimization *with* appropriate forecast updates is beneficial when the demand rate itself is unknown and stochastic, and thereby aids in combating 'large' shocks.

1.2 Model.

This section will describe our model rigorously. We then present examples of three problems that are captured within our model. The first two are the NRM and Ad Display problems described loosely earlier. The third concerns revenue management in a multi-class processing network with stochastic arrival rates.

System: Consider a bi-partite graph with I sources indexed by i and S sinks indexed by a. The edge set of this graph has size E, and a generic edge will be denoted by e. We understand by $i(e)$ the source node for edge e and by $a(e)$ its sink node. Given this graph, that will underlie a general allocation process over time from sources to sinks, we next describe several key model primitives:

1. Demand: We associate each source with a non-negative real valued stochastic process, $\{\Lambda_{i,t}(\omega)\}$, with continuous sample paths [3]. Each of these I processes

[3] we will suppress the dependence on ω if this is clear from context

are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The total demand at source i over the horizon $[0, T]$ will be understood as $\int_0^T \Lambda_{i,t} dt$. We denote by $\mathcal{F}_t$, the sigma algebra generated by the sample paths of the I demand processes up to time t, i.e. $\mathcal{F}_t = \sigma(\{\Lambda_s(\omega) : 0 \leq s \leq \hat{t}\}, \hat{t} \leq t, \omega \in \Omega)$.

2. Resources and Resource consumption: We are given a set of K distinct resources indexed by k. The available quantity of these resources at time t is given by a vector $x_t \in \mathbb{R}_+^K$. A unit allocation of demand along edge e will consume resources. The amount of resource k consumed is given by $A_{k,e}$; the vector of resources consumed by a unit allocation on edge e is thus the column vector $A_{\cdot,e} \triangleq A_e$. Denote by $A \in \mathbb{R}_+^{K \times E}$ the matrix whose eth column is A_e.

3. Prices/ Revenues: Allocating a unit of demand along edge e generates revenue p_e. Denote by $p \in \mathbb{R}_+^E$ the column vector whose eth component is p_e.

Control: Although our system evolves in continuous time, control is exerted at discrete times, $\{0, T/N, 2T/N, \ldots, T\} \triangleq \mathcal{T}_N$. The control chosen at time iT/N remains in effect over the interval $[iT/N, (i+1)T/N)$, and is specified by a vector $z \in \mathcal{Z}$ where the set of feasible controls at time t, $\mathcal{Z}$, is defined as:

$$\mathcal{Z} = \left\{ z \in R_+^E : \sum_{e:i(e)=i} z_e \leq 1 \ \forall i \right\}.$$

The control chosen at time iT/N determines the allocation of demand across edges over the subsequent interval. An *admissible control policy* is a $\mathcal{Z}$-valued process adapted to the filtration $\mathcal{F}_t$ which we will denote $\{z_t\}$. Denote the set of admissible control policies by Π_N. We also denote by $\{z_{e,t}\}$ the e-th component of $\{z_t\}$, i.e. the process describing the allocations across a particular edge e.

Dynamics: The state of the system at time t is effectively the history of the exogenously evolving demand processes up to that point and the quantities x_t of the

K resources that remain. The evolution of x_t is specified by the differential equation:

$$\frac{d}{dt}x_{k,t} = \begin{cases} -\sum_e A_{k,e}\Lambda_{i(e),t}z_{e,d(t)} & \text{if } x_{k,t} > 0 \\ 0 & \text{otherwise} \end{cases}$$

for all k. Here $d(t) = \max_{\{i:iN/T \leq t\}} iN/T$.

The Problem: Define the event $I_{e,t} = \{x_{k,t} > 0 \ \forall \ k \text{ s.t. } A_{k,e} > 0\}$. We are tasked with finding an admissible control policy $\{z_t\}$ that solves the following optimization problem:

$$\max_{\{z_t\} \in \Pi_N} \mathsf{E}\left[\int_0^T \sum_e p_e \Lambda_{i(e),t} z_{e,d(t)} \mathbf{1}_{\{I_{e,t}\}} dt\right] \tag{1.1}$$

We denote the optimal value to this optimal control problem by $J^{*,N}(x_0)$.[4]

The model we consider is best thought of as a *stochastic* fluid model. In particular, in a real system demand is potentially best captured as a multivariate point process with rate Λ_t; in our model we ignore the fluctuations of this counting process from its mean focusing instead on shocks in the rate process itself. In most applications the fluctuations of the counting process about its mean are 'small' in that their effect can be shown to be negligible in a regime where x_0 and Λ_t are scaled simultaneously by some scale factor that grows large.

1.2.1 A Family of Stochastic Demand Models.

While the algorithms we present will apply to general non-negative rate processes, it is hopeless to expect any single algorithm to perform well across a class this broad. As such our *analysis* will be limited to studying the performance of our prescriptions across a more limited family of processes. We consider this family here and then present examples of processes within this family:

Assumption 1.2.1 ***Structure of*** $\{\Lambda_t\}$***:***

1. $\Lambda_{i,t} = \left(\overline{\Lambda}_{i,t}\right)^+$, *where* $\overline{\Lambda}_t$ *is a Gaussian process with continuous sample paths.*

[4]We will often omit a reference to N when it is clear from context

2. $\mathsf{E}\left[\overline{\Lambda}_{i,t}\right] \triangleq \lambda_{i,t}$ *is positive.*

3. The variance of the random variable $\overline{\Lambda}_{i,t}$*,* $\sigma_{i,t}^2$*, is non-decreasing as a function of* t *and concave.*

Note that we do not make any assumptions on the correlation structure for this multi-variate stochastic process. While restricting attention to rate processes within some class of stochastic processes is certainly restrictive relative to, say, an adversarial model of rate processes, the family of processes permitted by Assumption 1.2.1 is ubiquitous in applications:

Ubiquity of Assumption 1.2.1.

We now demonstrate a family of processes that is both ubiquitous in applications and satisfies the requirements of the Assumption 1.2.1. Consider processes $\{\overline{\Lambda}_t\}$ defined according to:

$$\overline{\Lambda}_{i,t} = \lambda_{i,t} + \int_0^t \phi_i(t-s)dZ_{i,s}$$

where $\lambda_{i,t}$ is some Lipschitz-continuous, non-negative function of t for all i, $\phi_i(\cdot)$ is a Lipschitz continuous function of t such that $|\phi_i(\cdot)|$ is non-increasing, and Z_s is I-dimensional Brownian motion. For reasons that will become apparent shortly, we will refer to these as moving average processes. Moving average processes satisfy the requirements of Assumption 1.2.1.

To begin, we note that the class of moving average processes include a fair number of common continuous time processes used in stochastic modeling. For instance, setting $\phi_i = 1$ yields the Wiener process, while setting $\phi(s) = \exp(-s)$ yields the well known Ornstein-Uhlenbeck (or Langevin) process. The ubiquity of this class is perhaps more obvious if one considers evaluating the process at discrete points is time.

In particular, for any $\Delta > 0$, we have (via Ito isometry) that:

$$\overline{\Lambda}_{i,n\Delta} = \lambda_{i,n\Delta} + \sum_{k=0}^{n-1} \theta_{n-k}\epsilon_k$$

where ϵ_k are independent standard normal random variables, and $\theta_j = \sqrt{\int_{(j-1)\Delta}^{j\Delta} \phi^2(s)ds}$. This is nothing but a moving average model which finds wide application in time series modeling. In particular, Assumption 1.2.1 can thus be seen to permit moving average models that are

1. Of arbitrary order.

2. Not necessarily stationary.

3. Satisfying the property that the weights $\{\theta_i\}$ be non-increasing so that past shocks have smaller influence on the process than more recent shocks do.

This is evidently a broad class of models.

In subsequent performance analyses we will heavily exploit the natural *symmetry* in the marginals of this class of processes, as is made precise by the following lemma. The fact that it is the symmetry of the marginals of the demand process that drive our performance results is something of a unique insight:[5]

Lemma 1.2.2 *Let $f : \mathbb{R}_+ \to \mathbb{R}_+$ be continuous, nondecreasing, and concave with $f(0) = 0$. Then, provided the process $\{\Lambda_t\}$ satisfies Assumption 1.2.1 and further, $\lambda_{i,t} = \lambda_i, \forall i \in \mathcal{I}, t \in [0,T]$, we must have:*

$$\frac{\mathsf{E}\left[\frac{1}{T}\int_0^T f(\Lambda_{i,t})dt\right]}{f\left(\mathsf{E}\left[\frac{1}{T}\int_0^T \Lambda_{i,t}dt\right]\right)} \geq 0.342$$

for all i.

Remark 1.2.3 *(Low Volatility) The above result demonstrates an outcome of the symmetry of the marginals of our stochastic process and the resultant uniform bound holds irrespective of the magnitude of σ_t. Of course, one may hope that, if σ_t is small, then one might expect a tighter bound. In fact, Lemma 7 of Chen and Farias (2013)*

[5]In fact, analogues of Lemma 1.2.2 for non-Gaussian marginals drive analogous performance guarantees to those we shall derive in subsequent sections and we will provide one such example in Section 1.3.5.

establishes that if $\sigma_{i,t}/\lambda_i \leq \sqrt{2\pi}B$ for all t, then one has:

$$\frac{\mathsf{E}\left[\frac{1}{T}\int_0^T f(\Lambda_{i,t})dt\right]}{f\left(\mathsf{E}\left[\frac{1}{T}\int_0^T \Lambda_{i,t}dt\right]\right)} \geq \frac{1}{1+B} - \frac{B}{1+B}\left(\exp(1/4\pi B^2) + 0.853\right).$$

The first allocation scheme we consider **will not require** any knowledge of the specification of the process Λ_t. The second scheme we consider will use information on the drift of the process $\overline{\Lambda}_t$; in particular, this scheme will know $\{\lambda_{i,t}\}$ for all i. Note that even in the latter case, the information utilized is mild – it is trivial to construct processes with identical $\{\lambda_{i,t}\}$ but drastically different behavior; for instance, an OU process and a Wiener process.

1.2.2 Examples.

The generic allocation problem described above is quite rich and encapsulates several important classes of problems described below. The first two classes collectively drive billions of dollars in commerce. We describe these problems next:

Network Revenue Management: Here each of the source nodes corresponds to an arriving customer class so that $\Lambda_{i,t}$ captures the arrival rate of that class. Each of the sink nodes corresponds either to an itinerary or an 'offer set' depending on the precise NRM model we wish to capture. Every source has an edge to every sink. K corresponds to the number of legs on the network so that x_k is the residual capacity on leg k. We consider two distinct NRM models and interpret the control $\{z_t\}$ in each:

1. Separable Demand: Here each customer class (i.e. source node) is connected to exactly one itinerary (i.e. sink). $A_{k,e}$ corresponds to the number of seats consumed on leg k by itinerary $a(e)$ and p_e is the price of this itinerary. The demand at a given source node is consequently interpreted as demand for a given itinerary-price pair. The control $\{z_t\}$ is interpreted as follows: $z_{t,e}$ is the probability that the itinerary-price pair corresponding to edge e is made available at time t. This is the pre-dominant model for NRM problems.

2. Customer Choice: Here a given customer class may be connected to multiple sinks. A given sink is associated with an 'offer set' i.e. a set of itineraries the customer might be offered from which she will pick an option desirable to her. $A_{k,e}$ corresponds to the expected number of seats consumed on leg k when customers of class $i(e)$ are presented the offer set $a(e)$; p_e is the expected price of the itinerary selected from the offer set. The control $\{z_t\}$ is then interpreted as follows: $z_{t,e}$ is the probability that the offer set corresponding to edge e is shown to an arriving customer of class $i(e)$ at time t.

What is of note in our model is the ability to capture the fact that the demand rate for a given customer class $\Lambda_{i,t}$ is *stochastic* as opposed to being deterministic (as is assumed typically). This allows us to capture 'large' shocks in demand. Anticipating our scheme for allocation we will be able to do this without explicitly modeling the underlying demand process.

Online Ad Display Allocation: Here each of the source nodes corresponds to an arriving impression type [6], and each of the sink nodes corresponds to an advertiser. Every source has an edge to every sink. K is equal to the number of sinks; x_k can have several interpretations depending on the nature of the contract with the advertiser. For instance, the contract may specify that the advertiser will pay no more than a certain budget in which case x_k is the amount of this budget that remains. Alternatively, the advertiser may agree to having its ad displayed no more than a certain number of times over some period in which case x_k corresponds to the number of times advertiser k's ad can be displayed. p_e is interpreted as the revenues garnered in allocating an impression of type $i(e)$ to the advertiser $a(e)$. In the context of budget based contracts $A_{k,e} = p_e$; in the case of contracts based on the number of impressions served with an ad, $A_{k,e} = 1$ if $i(e)$ is an acceptable ad to advertiser $a(e)$ and 0 otherwise.

It is interesting to note that past work on the above problem in the adversarial setting also effectively considers a fluid model by assuming that the unit of allocation

[6] An impression can be thought of as a unit of web-traffic satisfying certain pre-assigned criteria, such as say gender, age, website etc.

is small relative to the budget, i.e. $A_{k,e} \ll x_k$.

Revenue Management for a Multi-class Processing Network: Consider a fluid processing network wherein I types of fluids are processed. Fluid i is processed at precisely one of K processing stations at rate $1/\mu_i$. Upon being processed, a unit of fluid i is transformed into $P_{i,i'}$ units of fluid i'. Let $P \in R_+^{I \times I}$ be the matrix whose i,i'th entry is $P_{i,i'}$ and assume that $I - P$ is invertible. Fluid i arrives to the system at a rate given by the stochastic process $\{\Lambda_{i,t}\}$ and the revenue from processing a unit of arriving fluid of type i (assuming all fluid generated in subsequent processing steps is also processed) is given by p_i. The goal is to design an admission policy that maximizes revenues from processing fluid over some finite horizon [7]. This may be cast within our framework as follows: we consider a problem with I sources corresponding to each fluid type and a single sink node. For every edge e we have $p_e = p_{i(e)}$. We set x_t to be the vector of uncommitted processing time available over the remainder of the horizon at each of the K stations at time t; thus $x_0 = T\mathbf{1}$. We set $A_e = \mu \odot (I-P)^{-1}u_{i(e)}$ where u_j is the jth unit vector [8]. Note that the quantity $v = (I-P)^{-1}u_j$ solves the Poisson equation $v = u_j + Pv$ and represents the effective amount of fluid (of all types) that an inflow of 1 unit of fluid j will introduce into the system. In this setting we interpret $z_{t,e}$ as the fraction of (non-reentrant) fluid of type $i(e)$ entering the system at time t that is admitted.

It is interesting to note how this processing model departs from typical processing network models: First, the arrival rate of a fluid is stochastic; the process driving this rate is allowed to be fairly general (as discussed in the preceding section). Second, we *maximize rewards* associated with processing fluid as opposed to minimizing some cost associated with backlog. Finally, we note that we do not allow a backlog at time T. A richer formulation would allow for us to optimize some combination of rewards associated with processing fluid and costs associated with backlog at time T; unfortunately our model does not allow this.

[7] Note that *no* revenues are generated from processing re-entrant fluid.
[8] For $a, b \in \mathbb{R}^n$, $a \odot b = [a_1 \cdot b_1, \ldots, a_n \cdot b_n]^\top$

1.3 A Re-Optimization Based Heuristic.

Imagine our information structure were such that $\{\Lambda_t, t \geq 0\} \in \mathcal{F}_0$. If this were the case, the control problem at hand reduces to a deterministic optimization problem; in particular, one simply employs an allocation rule given by any optimal solution to the program

$$\begin{aligned} \max \quad & \sum_e p_e z_e \int_0^T \Lambda_{i(e),t} dt \\ \text{subject to} \quad & \sum_e A_{k,e} z_e \int_0^T \Lambda_{i(e),t} dt \leq x_{k,0} \quad \forall\, k, \\ & z \in \mathcal{Z}. \end{aligned} \tag{1.2}$$

Here we define an extremely simple control scheme for the problem we face. The scheme we propose will resolve a similar linear program as the one above at the times in $\mathcal{T}_N$ with a certain 'projected demand' based on conditions at the time of re-optimization.

1.3.1 The Re-Optimization Scheme.

For $(t, \Lambda, x) \in \mathbb{R}_+ \times \mathbb{R}_+^I \times \mathbb{R}^K$, define the linear program $\mathrm{LP}(t, \Lambda, x)$ according to

$$\begin{aligned} \max \quad & \sum_e p_e z_e \Lambda_{i(e)} \cdot (T - t) \\ \text{subject to} \quad & \sum_e A_{k,e} z_e \Lambda_{i(e)} \cdot (T - t) \leq x_k^+ \quad \forall\, k, \\ & z \in \mathcal{Z}. \end{aligned}$$

Abusing notation, we will also denote the optimal value to this program by $\mathrm{LP}(t, \Lambda, x)$. We will consider a re-optimization based heuristic control policy $\{z_t^{\mathrm{R}}\}$ defined so that

$$z_t^{\mathrm{R}} = z_{d(t)}^{\mathrm{R}},$$

where $z^{\mathrm{R}}_{iT/N}$ is any optimal solution to the linear program LP $\left(iT/N, \Lambda_{iT/N}, x_{iT/N}\right)$. In words, this scheme assumes, at every point of time in $\hat{t} \in \mathcal{T}_N$ that the demand rate over the remaining time horizon will remain unchanged from $\Lambda_{\hat{t}}$, and employs the allocation rule that is optimal for such a scenario over the interval of time until the next re-solve. This procedure is summarized below

Re-optimization Heuristic
At each re-optimization interval $i = 0, \ldots, N-1$ 1. Measure demand rate $\Lambda_{iT/N}$ 2. Obtain fractional allocation $z^{\mathrm{R}}_{iT/N} \in \arg\max \mathrm{LP}(iT/N, \Lambda_{iT/N}, x_{iT/N})$ 3. Over the interval $[iT/N, (i+1)T/N)$, allocate the demand $\int_{iT/N}^{(i+1)T/N} \Lambda_t dt$ according to $z^{\mathrm{R}}_{iT/N}$

Now define $J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)$ as the revenues under the re-optimization heuristic under a specific sample path of demand, $\{\Lambda_t\}$, starting with inventory x_0. In particular,

$$J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0) = \int_0^T \sum_e p_e \Lambda_{i(e),t} z^{\mathrm{R}}_{e,t} \mathbf{1}_{\{I_{e,t}\}} dt$$

where, as before, $I_{e,t} = \{x_{k,t} > 0 \ \forall \ k \text{ s.t. } A_{k,e} = 1\}$. We denote the total expected reward under the re-optimization policy assuming a starting inventory level x_0 by $J^{\mathrm{R}}(x_0)$; $J^{\mathrm{R}}(x_0) = \mathsf{E}\left[J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)\right]$.

1.3.2 An Upper Bound.

Define $J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)$ as the optimal value of the (offline) optimization problem (1.2). That problem provides a useful upper bound on $J^{*,N}(x_0)$. In particular, we have:

Proposition 1.3.1 $\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right] \geq J^{*,N}(x_0)$ *for all* N.

The proof of this result can be found in the Appendix – the result is natural; knowing realized demand a-priori is in essence the best we can hope for.

1.3.3 Sample-path Properties of the Re-optimization Policy and a Lower Bound.

This section concerns two sample-path properties of the re-optimization policy. The first is simply a representation of $J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)$ in terms of the optimal value of the linear programs solved along a sample path. The second simple but crucial property is a statement of 'balanced' inventory consumption along a sample path.

Define $\Lambda^{\min}_{d(t)} \in \mathbb{R}^I_+$ so that $\Lambda^{\min}_{i,d(t)} = \min\{\Lambda_{i,t} : d(t) \le t < d(t) + N/T\}$. Similarly define $\Lambda^{\max}_{d(t)}$. We then have the following lemma:

Lemma 1.3.2

$$J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0) \ge \frac{T}{N}\sum_{j=0}^{N-1}\left(\frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_{jT/N}\right)}{T - jT/N} - C\left(\sum_e p_e\right)\sum_e\left(\Lambda^{\max}_{i(e),jT/N} - \Lambda^{\min}_{i(e),jT/N}\right)\right)$$

where C is some constant dependent only on the quantities $A_{k,e}$, $k \in 1,\ldots,K, e \in E$.

The proof of this lemma is relatively routine and can be found in the Appendix. The continuity of the sample paths of $\{\Lambda_t\}$ yield as an easy corollary the following result, whose proof is also in the Appendix.

Corollary 1.3.3

$$\liminf_N J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0) \ge \liminf_N \frac{T}{N}\sum_{j=0}^{N-1}\frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_{jT/N}\right)}{T - jT/N}$$

We next demonstrate a crucial sample path property that will eventually allow us to relate the linear programs solved at each opportunity to re-optimize to the initial (offline) LP solved at $t = 0$. The property is natural and simple to derive, but quite powerful in its application.

Lemma 1.3.4 *(Balancing) For every sample path of Λ_t,*

$$x_{k,nT/N} \ge x_{k,0}\frac{N-n}{N} + \sum_{j=0}^{n-1}\sum_e \frac{T}{N}A_{k,e}\left(\Lambda_{i(e),jT/N} - \Lambda^{\max}_{i(e),jT/N}\right).$$

for all $n < N$.

Proof We proceed by induction. The claim is trivially true for $n = 0$. Assume the claim true for $n = l - 1 < N - 1$ and observe that

$$\begin{aligned}
x_{k,(l+1)T/N} &\geq x_{k,lT/N} - \sum_e A_{k,e}\left(\int_{lT/N}^{(l+1)T/N} \Lambda_{i(e),t}dt\right) z^{\mathrm{R}}_{e,lT/N} \\
&\geq x_{k,lT/N} - \sum_e A_{k,e}\left(\int_{lT/N}^{(l+1)T/N} \Lambda^{\max}_{i(e),lT/N}\right) z^{\mathrm{R}}_{e,lT/N} \\
&= x_{k,lT/N} - \sum_e \frac{T}{N} A_{k,e}\Lambda_{i(e),lT/N} z^{\mathrm{R}}_{e,lT/N} + \sum_e \frac{T}{N} A_{k,e}\left(\Lambda_{i(e),lT/N} - \Lambda^{\max}_{i(e),lT/N}\right) z^{\mathrm{R}}_{e,lT/N} \\
&\geq x_{k,lT/N}\left(1 - \frac{1}{N-l}\right) + \sum_e \frac{T}{N} A_{k,e}\left(\Lambda_{i(e),lT/N} - \Lambda^{\max}_{i(e),lT/N}dt\right) z^{\mathrm{R}}_{e,lT/N}
\end{aligned} \tag{1.3}$$

The final inequality follows from the fact that $z^{\mathrm{R}}_{lT/N}$ is a feasible solution to $\mathrm{LP}\left(lT/N, \Lambda_{lT/N}, x_{lT/N}\right)$. But the induction hypothesis yields

$$x_{k,lT/N} \geq x_{k,0}\frac{N-l}{N} + \sum_{j=0}^{l-1}\sum_e \frac{T}{N} A_{k,e}\left(\Lambda_{i(e),jT/N} - \Lambda^{\max}_{i(e),jT/N}\right)$$

and substituting in the final inequality of (1.3) then yields

$$x_{k,(l+1)T/N} \geq x_{k,0}\frac{N-(l+1)}{N} + \sum_{j=0}^{l}\sum_e \frac{T}{N} A_{k,e}\left(\Lambda_{i(e),jT/N} - \Lambda^{\max}_{i(e),jT/N}\right).$$

Induction on l completes the proof.

Lemma 1.3.4 yields as a corollary the following result whose proof may be found in the Appendix:

Corollary 1.3.5

$$\liminf_N \frac{T}{N}\sum_{j=0}^{N-1} \frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_{jT/N}\right)}{T - jT/N} \geq \liminf_N \frac{T}{N}\sum_{j=0}^{N-1} \frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_0(N-j)/N\right)}{T - jT/N}$$

Corollaries 1.3.3 and 1.3.5 together imply the following lower bound on revenues under the re-optimization heuristic:

Theorem 1.3.6

$$\liminf_N J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0) \geq \liminf_N \frac{T}{N} \sum_{j=0}^{N-1} \frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_0(N-j)/N\right)}{T - jT/N}$$

1.3.4 Properties of the Re-Solved LP and a Decomposition.

Here we briefly develop an 'expansion' of the optimal value $\mathrm{LP}(t, \Lambda, x)$ that is separable in the components of Λ. This expansion will serve us in our performance analysis and, in particular, will be crucial in analyzing an inherently multi-dimensional system via a single-dimensional analysis. We begin with some definitions.

For every $i \in \{1, \ldots, I\}$ and tuples $(t, \Lambda, x) \in \mathbb{R}_+ \times \mathbb{R}_+^I \times \mathbb{R}_+^K$, let $z(t, \Lambda, x) \in \mathbb{R}^E$ denote an optimal solution to $\mathrm{LP}(t, \Lambda, x)$, and define the functions $f_i^{t,\Lambda,x} : \mathbb{R}_+ \to \mathbb{R}_+$ according to

$$f_i^{t,\Lambda,x}(w) = \sum_{e:i(e)=i} p_e \min\{\Lambda_i, w\} z_e(t, \Lambda, x).$$

We catalog a few properties of these functions that will serve us well in the sequel; these properties are proved in the Appendix.

Lemma 1.3.7 *For every $i \in \mathcal{I}$, and $(t, \Lambda, x) \in \mathbb{R}_+ \times \mathbb{R}_+^I \times \mathbb{R}_+^K$, the function $f(w) \triangleq f_i^{t,\Lambda,x}(w)$ satisfies the following properties:*

1. *$f(0) = 0$ and f is continuous and non-decreasing.*

2. *$f(\cdot)$ is concave.*

3. *For $w, v > 0$, $\frac{f(w)}{f(v)} \geq \min\left\{\frac{w}{v}, 1\right\}$.*

4. *$\frac{1}{x} \int_0^x f(w)dw \leq f\left(\frac{x}{2}\right)$.*

The utility of the functions $f_i^{t,\Lambda,x}$ lies in the fact that they allow us to construct useful approximations to $\mathrm{LP}(t, \Lambda, x)$ that are separable in the components of Λ. This is made precise by the following result whose proof is deferred to the Appendix:

Lemma 1.3.8 *For any* $(t,u,x) \in \mathbb{R}_+ \times \mathbb{R}_+^I \times \mathbb{R}_+^K$*, and an arbitrary* $\Lambda \in \mathbb{R}_+^I$*, we have:*

$$\mathrm{LP}(t,u,x) \geq (T-t)\sum_i f_i^{(t,\Lambda,x)}(u_i)$$

with equality for $u = \Lambda$.

1.3.5 Performance Guarantees for the Re-optimization Scheme.

With the lower and upper bounds developed thus far, we are finally in a position to present performance guarantees for our approach for processes where $\lambda_{i,t} = \lambda_i$.

Theorem 1.3.9 *For demand processes* $\{\Lambda_t\}$ *satisfying Assumption 1.2.1, and with* $\lambda_{i,t} = \lambda_i$ *for all* i,t*, we have, assuming an initial inventory of* x_0*:*

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]} \geq 0.342.$$

Proof We know by Fatou's Lemma that

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]} \geq \frac{\mathsf{E}\left[\liminf_N J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]}.$$

We will proceed in turn to bound the numerator and denominator. We then have that:

$$\begin{aligned}
\liminf_N J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0) &\geq \liminf_N \frac{T}{N}\sum_{j=0}^{N-1}\frac{\mathrm{LP}\left(0,\Lambda_{jT/N},x_0\right)}{T} \\
&\geq \liminf_N \frac{T}{N}\sum_{j=0}^{N-1}\sum_i f_i^{(0,\hat{\Lambda},x_0)}(\Lambda_{i,jT/N}) \qquad (1.4)\\
&= \sum_i \int_0^T f_i^{(0,\hat{\Lambda},x_0)}(\Lambda_{i,t})dt
\end{aligned}$$

where $\hat{\Lambda} \in \mathbb{R}_+^I$ is arbitrary. The first inequality is a consequence of Corollaries 1.3.3 and 1.3.5 and the fact that $\mathrm{LP}(t,\Lambda,x(1-t/T)) = \frac{T-t}{T}\mathrm{LP}(0,\Lambda,x)$ for $0 \leq t \leq T$, and arbitrary $\Lambda \in \mathbb{R}_+^I$, $x \in \mathbb{R}_+^K$. The second inequality is Lemma 1.3.8. The final equality follows from the continuity of $f_i^{(0,\hat{\Lambda},x_0)}(\cdot)$ (Lemma 1.3.7, property 1) and the continuity of $\Lambda_{i,t}$ in t.

Next, we have that:

$$\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right] = \mathsf{E}\left[\mathrm{LP}\left(0, \overline{\Lambda}, x_0\right)\right] \leq \mathrm{LP}\left(0, \mathsf{E}\left[\overline{\Lambda}\right], x_0\right) = \sum_i T f_i^{(0,\mathsf{E}[\overline{\Lambda}],x_0)}\left(\mathsf{E}\left[\overline{\Lambda}_i\right]\right) \tag{1.5}$$

where we define $\overline{\Lambda} \in \mathbb{R}_+^I$ according to $\overline{\Lambda}_i = \frac{1}{T}\int_0^T \Lambda_{i,t}dt$. The inequality above is Jensen's inequality (since $\mathrm{LP}(t,u,x)$ is concave in u). The final equality is Lemma 1.3.8. Now, from (1.4) and (1.5), it follows that:

$$\begin{aligned}
\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]} &\geq \frac{\sum_i \mathsf{E}\left[\int_0^T \frac{1}{T} f_i^{(0,\mathsf{E}[\overline{\Lambda}],x_0)}(\Lambda_{i,t})dt\right]}{\sum_i f_i^{(0,\mathsf{E}[\overline{\Lambda}],x_0)}\left(\mathsf{E}\left[\overline{\Lambda}_i\right]\right)} \\
&\geq \min_i \frac{\mathsf{E}\left[\int_0^T \frac{1}{T} f_i^{(0,\mathsf{E}[\overline{\Lambda}],x_0)}(\Lambda_{i,t})dt\right]}{f_i^{(0,\mathsf{E}[\overline{\Lambda}],x_0)}\left(\mathsf{E}\left[\overline{\Lambda}_i\right]\right)} \\
&\geq 0.342
\end{aligned}$$

where the final inequality is the estimate derived in Lemma 1.2.2.

The above guarantee is remarkable in that it is *uniform* over a broad class of demand processes. In particular, the guarantee places no restrictions whatsoever on the volatility of these processes nor on their correlation or autocorrelation structures. Of course, in the event that the volatility of the underlying process were small, one expects even better performance from this very natural algorithm, and indeed a proof essentially identical to the one above yields the following theorem:

Theorem 1.3.10 *Consider demand processes $\{\Lambda_t\}$ satisfying Assumption 1.2.1, with $\lambda_t = \lambda$ for all t. In addition assume that $\sigma_t/\lambda \leq \sqrt{2\pi}B$. We then have, assuming an* *initial inventory of x_0:*

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]} \geq \max\left\{0.342, \frac{1}{1+B} - \frac{B}{1+B}\left(\exp(-1/4\pi B^2) + 0.853\right)\right\}$$

The proof of the above theorem proceeds identically to that of Theorem 1.3.9 with the exception that, in the very final inequality of that proof, we use the general bound given in Remark 1.2.3.

Theorems 1.3.9 and 1.3.10 together establish a strong statement about the robustness of our re-optimization heuristic. In particular, these theorems establish that with frequent re-optimization, this natural heuristic attains the ability to compete with a clairvoyant with perfect knowledge of the sample paths of the demand process, irrespective of the volatility of that process. Simultaneously, in the event that the underlying process were not volatile at all the same scheme is essentially optimal.

Essential Properties of the Rate Process and Potential Generalizations.

The essence of Theorem 1.3.9 is that it reduces the performance analysis of our re-optimization scheme to the analysis of *simple* properties of *marginals* of the the rate process. As it happens, these properties can be tractably quantified for moving average processes (which as we established earlier are a particularly interesting family of processes). However, it is possible that such guarantees might be established for other classes of processes, and to this end we explicitly isolate the property of the rate process that drives the bound of Theorem 1.3.9 and then present an example of a process that is not a moving average process but admits a constant factor guarantee.

Notice that the proof of Theorem 1.3.9 yields

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]} \geq \min_i \frac{\mathsf{E}\left[\frac{1}{T}\int_0^T f_i(\Lambda_{i,t})dt\right]}{f_i\left(\mathsf{E}\left[\frac{1}{T}\int_0^T \Lambda_{i,t}dt\right]\right)}$$

for some set of functions f_i, each satisfying Lemma 1.3.7, and *any* rate process with continuous sample paths. We were able to come up with a uniform lower bound to this quantity for arbitrary, non-stationary moving average processes.

The analysis of the above quantity is substantially simplified in the case of stationary processes. In particular, consider processes that are *stationary* and whose marginals have finite expectation. If this were the case, we have:

$$\begin{aligned}
\frac{\mathsf{E}\left[\frac{1}{T}\int_0^T f_i(\Lambda_{i,t})dt\right]}{f_i\left(\mathsf{E}\left[\frac{1}{T}\int_0^T \Lambda_{i,t}dt\right]\right)} &\geq \mathsf{E}\left[\frac{1}{T}\int_0^T \min\left\{\frac{\Lambda_{i,t}}{\mathsf{E}\left[\frac{1}{T}\int_0^T \Lambda_{i,t}dt\right]}, 1\right\} dt\right] \\
&= \mathsf{E}\left[\frac{1}{T}\int_0^T \min\left\{\frac{\Lambda_{i,t}}{\mathsf{E}\left[\Lambda_{i,t}\right]}, 1\right\} dt\right] \\
&= \mathsf{E}\left[\min\left\{\frac{\Lambda_{i,0}}{\mathsf{E}\left[\Lambda_{i,0}\right]}, 1\right\}\right]
\end{aligned}$$

where we have used Lemma 1.3.7 for the first inequality, and the stationarity of the process with Fubini's theorem for the next two equalities. Notice that this bound has a natural interpretation. It measures, in a sense, the asymmetry of the marginals of the process under consideration. In general, this quantity can be arbitrarily small (consider, for instance, a suitable two point distribution). However, as witnessed by the moving average family of processes, there are potentially *large* families of stochastic processes for which this quantity is uniformly bounded for all processes within the family.

Here we give another example. Imagine that each of the I marginals are described by a *Cox-Ingersoll-Ross* (CIR) process. A CIR process is a non-negative, mean-reverting process and is perhaps the best known example of an *affine* process. It is typically used to model the behavior of non-negative quantities such interest rates or (more recently) arrival rates to a queueing system (see Besbes and Maglaras (2009)). The process is defined as the solution of the stochastic differential equation:

$$d\Lambda_{i,t} = \theta_i(\lambda_i - \Lambda_{i,t})dt + \sigma_i\sqrt{\Lambda_{i,t}}dZ_{it} \text{ for } \theta_{\mathrm{i}}, \lambda_{\mathrm{i}}, \sigma_{\mathrm{i}} > 0.$$

When $2\theta_i\lambda_i > \sigma_i^2$ (the regime typically considered in any modeling with this process), the CIR process is strictly positive and has a stationary distribution. This stationary distribution is Gamma (as opposed to Gaussian for moving average processes) with shape parameter $a \triangleq 2\theta_i \lambda_i / \sigma_i^2$ and scale parameter $\sigma_i^2/2\theta_i$. Consequently, for *stationary* CIR processes we may compute (see Farias and Van Roy (2010)),

$$\mathsf{E}\left[\min\left\{\frac{\Lambda_{i,0}}{\mathsf{E}\left[\Lambda_{i,0}\right]}, 1\right\}\right] = 1 - \frac{\Gamma(a+1,a)}{\Gamma(a+1)} + \frac{\Gamma(a,a)}{\Gamma(a)} \geq 1 - \frac{1}{e}$$

for $a \geq 1$. This yields the following Theorem

Theorem 1.3.11 *For demand processes whose marginals are stationary (but otherwise arbitrary) Cox-Ingersoll-Ross processes, we have assuming an initial inventory of x_0:*

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]} \geq 1 - \frac{1}{e}$$

Of course, the class of processes here, while allowing for arbitrary volatility, is nowhere as large or perhaps interesting as the moving average processes we have focused on, but the result provides a sense of the generalizability of the analysis.

1.3.6 The Impact of Re-Optimization Frequency.

Our performance guarantees thus far call for frequent re-optimization (i.e. large N). At this juncture we ask two questions:

1. Is a large value of N – and the implicit demand forecast updating – truly necessary for good performance of our scheme?

2. If large values of N are indeed necessary, what impact does a finite number of re-optimizations have on the performance guarantee of Theorem 1.3.9?

We will answer the first question in the affirmative by providing a sequence of problems where infrequent re-optimization results in poor (in fact, arbitrarily poor) performance. As for the second question, we will use a well known result on the modulus of continuity for the sample paths of Brownian motion, to characterize the error due to finite re-optimization.

Throughout this section, we make the dependence of the revenue on the number of re-optimization intervals explicit through the notation $J^{\mathrm{R},N}_{\{\Lambda_t\}}$ (in place of simply $J^{\mathrm{R}}_{\{\Lambda_t\}}$). To answer the first question and demonstrate the importance of re-solving, we will describe a sequence of problems, indexed by T, and show that if we choose to not re-optimize (and thereby not adjust forecasts) under our scheme, such a choice can grow arbitrarily sub-optimal as T grows large. This will show that re-optimization plays a dramatic role in the actual performance of our scheme. We next describe this sequence of problems:

Example Consider an allocation problem with two sources and a single sink. We have one resource type so that $K = 1$, and set $A_{1,(1,1)} = A_{1,(2,1)} = 1$, while we set $x_0 = T$. Our time horizon is T, and we set $p_{(1,1)} = 1$ and $p_{(2,1)} = T^{1/3+\epsilon}$ where $\epsilon > 0$ is some constant. The demand process is described as follows: $\Lambda_{1,t} = (\frac{1}{2} + \sqrt{2\pi} W_t)^+$, where W_t is standard Brownian motion and $\Lambda_{2,t} = \frac{1}{2}$. Notice that this demand process satisfies Assumption 1.2.1.

We then have the following result whose proof may be found in the Appendix:

Proposition 1.3.12 *For the problem described in Example 1.3.6, we have:*

$$\frac{J^{\mathrm{R},1}(x_0)}{J^{\mathrm{UB}}(x_0)} = O(T^{-1/3}).$$

Now, in contrast, Theorem 1.3.9 implies that

$$\liminf_{N} \frac{J^{\mathrm{R},N}(x_0)}{J^{\mathrm{UB}}(x_0)} = \Theta(1).$$

Contrasting these two results demonstrates the importance of frequent re-optimization in the context of using our allocation scheme.

In light of the above result, we can move on to answering the next logical question, which asks us to establish the effect of a finite number of re-optimizations, or more specifically a 'rate' for the limit infimum in Theorem 1.3.9. We will establish such a rate for moving average demand rate processes. Doing this will require the following principal technical tool which is a *global* modulus of continuity for sample paths of moving average processes (see, for instance Karatzas and Shreve (1991)):

Theorem 1.3.13 *(Levy's modulus of continuity) Let $\overline{\Lambda}_t$ be a moving average process. Then, almost surely,*

$$\limsup_{h\to 0} \sup_{0\le t\le T-h} \frac{|\overline{\Lambda}_{t+h} - \overline{\Lambda}_t|}{\sqrt{2h\log 1/h}} \le \phi(0).$$

Note that Levy's theorem is typically stated for standard Brownian motion where the limit above can be shown to exist, and is equal to 1. The above result is, in fact, a simple corollary to that theorem. Roughly, the theorem can be interpreted as stating that $\sup_{0\le t\le T-h} |\overline{\Lambda}_t - \overline{\Lambda}_{t+h}| = O(\sqrt{h\log 1/h})$. We will now employ this theorem to prove a performance guarantee for the re-optimization scheme allowing only a finite number of re-optimizations.

Theorem 1.3.14 *Assume the demand rate process $\{\Lambda_t\}$ satisfies $\Lambda_{i,t} = \left(\overline{\Lambda}_{i,t}\right)^+$ where $\overline{\Lambda}_t$ is a multi-variate moving average process. Define $\sigma_i \triangleq \phi_i(0)$. Then,*

$$\mathsf{E}\left[J^{\mathrm{R},N}_{\{\Lambda_t\}}(x_0)\right] \ge 0.342\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right] - \Delta(N),$$

where $\Delta(N)$ satisfies

$$\limsup_N \frac{\Delta(N)}{\sqrt{\log N/N}} \le C'\sum_i \sigma_i.$$

The constant C' can be specified independent of the demand process.

The constant C' above is derived explicitly in the proof of the theorem (which can be found in the Appendix). It depends solely on the quantities p, A, K, T and I. As such, the above result can loosely be interpreted as stating that a finite number of re-optimizations introduces an *additive* error that behaves roughly like $O\left(\sum_i \sigma_i\sqrt{\log N/N}\right)$. This has an interesting interpretation: the additive error component grows at most linearly with the volatility of the process. However, with sufficiently frequent re-optimization (as specified by the rate in the theorem), this additive error can be made arbitrarily small. Put another way, frequent re-optimization is particularly valuable when the demand rate process in question is highly volatile.

1.4 A Heuristic That Uses Forecasts: β-Re-optimization.

In this Section, we develop and analyze a heuristic that has access to a deterministic forecast of demand evolution. In the context of our assumed model of stochastic demand processes, we will assume knowledge of λ_t. The heuristic we develop is closely related to the re-optimization heuristic, and will be competitive for processes where λ_t is not necessarily constant.

Imagine a scenario wherein the demand process had no noise – in particular, $\sigma_t = 0$. In this event, any optimal solution to $\text{LP}(0, \int_0^T \lambda_t dt, x_0)$ constitutes an optimal (and static) allocation rule. Of course, if $\sigma_t > 0$, this is not the case, and so the policy we propose will entail a careful convex combination of this 'deterministic' policy with a re-optimization policy analogous to that studied in the previous Section.

Informally speaking, our heuristic will approximately simulate the following allocation over time:

1. Split the total demand $\{\Lambda_t\}$ and resource capacity x_0 into two systems. System 1 (the 're-optimization' system) sees the demand process $\{(1-\beta)\Lambda_t\}$ and begins with inventory $(1-\beta)x_0$. System 2 (the 'deterministic' system) sees the demand process $\{\beta\Lambda_t\}$ and begins with inventory βx_0.

2. Apply the re-optimization policy (of the last Section) $\{z_t^{\text{R}}\}$ to the re-optimization system.

3. Apply the deterministic policy (to be defined momentarily) $\{z_t^{\text{D}}\}$ to the deterministic system.

This Section will establish uniform performance guarantees for the policy described (loosely) above; these guarantees will implicitly identify an oblivious choice of β that is 'good'. In addition, we will provide a sketch of potential improvements to the policy and a guideline on how to tune β given further information about the demand process. In broad steps, the analysis will proceed along the lines of the following roadmap:

1. Establish a performance guarantee for the re-optimization policy with respect to an upper bound on revenues for the 're-optimization system'. This will utilize our previous analysis. See Lemma 1.4.4.

2. Establish a performance guarantee for the deterministic policy with respect to an upper bound on revenues for the 'deterministic system'. See Lemma 1.4.5.

3. Establish a relationship between optimal revenues for the re-optimization and deterministic systems with optimal revenues for the original problem. See Lemma 1.4.6.

4. Using the above three steps, compute a performance guarantee for the overall scheme. This guarantee will be a function of β, which we may then optimize. See Theorem 1.4.7.

We next define our policy formally.

1.4.1 The β-Re-optimization Scheme.

We first construct a few auxiliary processes. In particular, define the 're-optimization' inventory process $\hat{x}^{\mathrm{R}}_t$ according to:

$$\hat{x}^{\mathrm{R}}_{k,t} = \hat{x}^{\mathrm{R}}_{k,0} - \int_0^t \mathbf{1}_{\hat{x}^{\mathrm{R}}_{k,t}>0} \sum_e \Lambda_{i(e),t} A_{k,e} \hat{z}^{\mathrm{R}}_{e,d(t)} dt$$

where $\hat{x}^{\mathrm{R}}_0 = x_0$ and $\hat{z}^{\mathrm{R}}_{iT/N}$ is any optimal solution to LP $\left(iT/N, \Lambda_{iT/N}, \hat{x}^{\mathrm{R}}_{iT/N}\right)$. In words, $\hat{x}^{\mathrm{R}}_t$ is the inventory process obtained if one employs the allocation policy $z_t = \hat{z}^{\mathrm{R}}_{d(t)}$.

Next, let us denote by z^{D}, an optimal solution to LP$(0, \int_0^T \lambda_t dt, x_0)$. Define the 'deterministic' policy $\{\hat{z}^{\mathrm{D}}_t\}$ according to $\hat{z}^{\mathrm{D}}_t \triangleq \left(\frac{\lambda_{d(t)}}{\Lambda_{d(t)}} \wedge 1\right) z^{\mathrm{D}}$. Note that $\{\hat{z}^{\mathrm{D}}_t\}$ is an optimal allocation policy in the event $\sigma_t = 0$.

The control we propose, denoted $\left\{z_t^{\text{R}-\beta}\right\}$ is defined according to

$$z_t^{\text{R}-\beta} = (1-\beta)\hat{z}^{\text{R}}_{d(t)} + \beta\hat{z}^{\text{D}}_{d(t)}.$$

β-Re-optimization Heuristic

1. Compute z^D
2. At each re-optimization interval $i = 0, \ldots, N-1$
   a. Measure demand rate $\Lambda_{iT/N}$
   b. Obtain allocation $\hat{z}^{\text{R}}_{iT/N} \in \arg\max \text{LP}(iT/N, \Lambda_{iT/N}, \hat{x}^{\text{R}}_{iT/N})$
   c. Over the interval $[iT/N, (i-1)T/N)$, allocate the demand according to $(1-\beta)\hat{z}^{\text{R}}_{iT/N} + \beta\hat{z}^{\text{D}}_{iT/N}$

1.4.2 Preliminary Sample Path Properties.

Here we identify several sample path properties for the β-re-optimized policy. Define the 're-optimized' revenue under the β-re-optimized policy according to:

$$J^{\text{R}}_{S,\{\Lambda_t\}} = \int_0^T \sum_e p_e \hat{z}^{\text{R}}_{e,d(t)} \Lambda_{i(e),t} \mathbb{I}_{\{\hat{I}_{e,t}\}} dt$$

where $\hat{I}_{e,t} = \{\hat{x}^{\text{R}}_{k,t} > 0 \ \forall k \text{ s.t. } A_{k,e} > 0\}$. Similarly define the 'deterministic' revenue under the β-re-optimized policy according to:

$$J^{\text{R}}_{D,\{\Lambda_t\}} = \int_0^T \sum_e p_e \hat{z}^{\text{D}}_{e,d(t)} \Lambda_{i(e),t} dt.$$

Given the definitions of $\hat{x}^{\text{R}}_t$ and $\hat{z}^{\text{R}}_{jT/N}$, Theorem 1.3.6 immediately implies:

$$\liminf_N J^{\text{R}}_{S,\{\Lambda_t\}}(x_0) \geq \liminf_N \frac{T}{N} \sum_{j=0}^{N-1} \frac{\text{LP}\left(jT/N, \Lambda_{jT/N}, \hat{x}^{\text{R}}_0(N-j)/N\right)}{T - jT/N} \tag{1.6}$$

Moreover, we have:

$$
\begin{aligned}
J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0) = \sum_e p_e \int_0^T \Lambda_{i(e),t}\hat{z}^{\mathrm{D}}_{e,t}dt &\geq \frac{T}{N}\sum_{j=0}^{N-1}\sum_e p_e \Lambda^{\min}_{i(e),jT/N}\hat{z}^{\mathrm{D}}_{e,jT/N} \\
&= \frac{T}{N}\sum_{j=0}^{N-1}\Bigg(\sum_e p_e\Lambda_{i(e),jT/N}\hat{z}^{\mathrm{D}}_{e,jT/N} - \\
&\qquad \sum_e p_e(\Lambda_{i(e),jT/N} - \Lambda^{\min}_{i(e),jT/N})\Bigg)
\end{aligned}
$$

so that

$$
\liminf_N J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0) \geq \liminf_N \frac{T}{N}\sum_{j=0}^{N-1}\sum_e p_e\Lambda_{i(e),jT/N}\hat{z}^{\mathrm{D}}_{e,jT/N} \tag{1.7}
$$

Finally, we have the following result that decomposes the revenues under the β−re-optimized policy into the above 're-optimized' and 'deterministic' revenue components; the proof may be found in the Appendix.

Lemma 1.4.1

$$
\liminf_N \mathsf{E}\left[J^{\mathrm{R}-\beta}_{\{\Lambda_t\}}(x_0)\right] \geq \liminf_N (1-\beta)\mathsf{E}\left[J^{\mathrm{R}}_{S,\{\Lambda_t\}}(x_0)\right] + \liminf_N \beta\mathsf{E}\left[J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0)\right]
$$

Remark 1.4.2 *We note that although we will not need this fact, the inequality established in the result above also holds on a sample path basis; In particular, we can show:*

$$
\liminf_N J^{\mathrm{R}-\beta}_{\{\Lambda_t\}}(x_0) \geq \liminf_N (1-\beta)J^{\mathrm{R}}_{S,\{\Lambda_t\}}(x_0) + \liminf_N \beta J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0).
$$

1.4.3 Performance Analysis.

This Section establishes uniform performance guarantees on the performance of the β-re-optimization scheme. Using the decomposition arrived at it the previous Section, this guarantee is arrived at by deriving appropriate uniform guarantees on the re-optimized and deterministic revenue terms. The former guarantee is essentially obtained via Theorem 1.3.9 while the latter requires a new argument. Our arguments will require one extra technical assumption on the demand process $\{\Lambda_t\}$:

Assumption 1.4.3 *For all i, $\mathsf{E}\left[\max_{t\in[0,T]} \Lambda_{i,t}\right] < \infty$.*

We begin with an analysis of the 're-optimized' revenue component, $J^{\mathrm{R}}_{S,\{\Lambda_t\}}$. The proof is essentially a corollary to Theorem 1.3.9 and deferred to the Appendix.

Lemma 1.4.4

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{S,\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{(\Lambda_t-\lambda_t)^+\}}(x_0)\right]} \geq 0.342$$

Our next result provides an analysis of the 'deterministic' component of revenues under the β-re-optimization scheme, $J^{\mathrm{R}}_{D,\{\Lambda_t\}}$.

Lemma 1.4.5

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0)\right]}{J^{\mathrm{UB}}_{\{\lambda_t\}}(x_0)} \geq \frac{1}{2}$$

Proof Now, we have:

$$\begin{aligned}
\liminf_N \mathsf{E}\left[J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0)\right] &\geq \mathsf{E}\left[\liminf_N J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0)\right] \\
&\geq \mathsf{E}\left[\liminf_N \frac{T}{N}\sum_{j=0}^{N-1}\sum_e p_e \Lambda_{i(e),jT/N}\hat{z}^{\mathrm{D}}_{e,jT/N}\right] \\
&= \lim_N \mathsf{E}\left[\frac{T}{N}\sum_{j=0}^{N-1}\sum_e p_e \Lambda_{i(e),jT/N}\hat{z}^{\mathrm{D}}_{e,jT/N}\right] \\
&\geq \liminf_N \mathsf{E}\left[\frac{T}{N}\sum_{j=0}^{N-1}\sum_e p_e \mathbb{I}_{\{\Lambda_{i(e),jT/N}\geq\lambda_{i(e),jT/N}\}}\lambda_{i(e),jT/N} z^{\mathrm{D}}_e\right] \\
&= \lim_N \frac{T}{N}\sum_{j=0}^{N-1}\sum_e p_e \frac{1}{2}\lambda_{i(e),jT/N} z^{\mathrm{D}}_e \\
&= \frac{1}{2} J^{\mathrm{UB}}_{\{\lambda_t\}}(x_0).
\end{aligned}$$

The first inequality is Fatou's Lemma while the second follows from (1.7). The second equality follows from the Dominated convergence theorem: in particular, observe that

$$\lim_N \frac{T}{N}\sum_{j=0}^{N-1}\sum_e p_e \Lambda_{i(e),jT/N}\hat{z}^{\mathrm{D}}_{e,jT/N}$$

exists by the definition of $\hat{z}^{\mathrm{D}}_{e,jT/N}$ and the continuity of $\Lambda_{i,t}$ and $\lambda_{i,t}$. Further,

$$\sum_{j=0}^{N-1}\sum_{e} p_e \Lambda_{i(e),jT/N}\hat{z}^{\mathrm{D}}_{e,jT/N} \leq E\sum_{i}\max_{t\in[0,T]}\Lambda_{i,t}$$

which was assumed finite by Assumption 1.4.3.

Before moving on to our approximation guarantee, we establish one last fact (the proof is in the Appendix):

Lemma 1.4.6

$$J^{\mathrm{UB}}_{\{\lambda_t\}}(x_0) + \mathsf{E}\left[J^{\mathrm{UB}}_{\{(\overline{\Lambda}_t-\lambda_t)^+\}}(x_0)\right] \geq \mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]$$

We are now in a position to provide a uniform performance guarantee for the β-re-optimized scheme. In particular, we have:

Theorem 1.4.7

$$\liminf_{N} \frac{\mathsf{E}\left[J^{\mathrm{R}-\beta}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]} \geq (1-\beta)0.342 \wedge \beta 0.5$$

Proof We have

$$\begin{aligned}
\liminf_{N} \frac{\mathsf{E}\left[J^{\mathrm{R}-\beta}_{\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]} \geq & \frac{\liminf_N (1-\beta)\mathsf{E}\left[J^{\mathrm{R}}_{S,\{\Lambda_t\}}(x_0)\right] + \liminf_N \beta\mathsf{E}\left[J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{(\overline{\Lambda}_t-\lambda_t)^+\}}(x_0)\right] + J^{\mathrm{UB}}_{\{\lambda_t\}}(x_0)} \\
\geq & \frac{(1-\beta)0.342\mathsf{E}\left[J^{\mathrm{UB}}_{\{(\overline{\Lambda}_t-\lambda_t)^+\}}(x_0)\right] + \beta 0.5 J^{\mathrm{UB}}_{\{\lambda_t\}}(x_0)}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{(\overline{\Lambda}_t-\lambda_t)^+\}}(x_0)\right] + J^{\mathrm{UB}}_{\{\lambda_t\}}(x_0)} \\
= & (1-\beta)0.342 \wedge \beta 0.5.
\end{aligned}$$

The first inequality follows from Lemmas 1.4.6 and 1.4.1. The second inequality follows from Lemmas 1.4.5 and 1.4.4.

Before closing this Section, we remark on the implications of the above result and several issues related to implementing the β-re-optimized scheme in practice:

1. Choosing β: Optimizing the bound in Theorem 1.4.7 suggests setting $\beta \sim 0.4$. This is an oblivious choice of β that results in a uniform performance guarantee.

2. With further knowledge about the demand process one might be able to do better. For instance, an estimate of the relative values of $J^{\text{UB}}_{\{\lambda_t\}}(x_0)$ and $\mathsf{E}\left[J^{\text{UB}}_{\{(\bar{\Lambda}_t-\lambda_t)^+\}}(x_0)\right]$ (which can both be computed easily if the model for Λ_t were known) suggests a better selection rule: set $\beta = 1$ if $J^{\text{UB}}_{\{\lambda_t\}}(x_0) > 0.684\mathsf{E}\left[J^{\text{UB}}_{\{(\bar{\Lambda}_t-\lambda_t)^+\}}(x_0)\right]$ and set $\beta = 0$ otherwise. A practical guideline is to set $\beta = 1$ when we believe there is little to no volatility in the demand process, and set $\beta = 0$ otherwise. Our numerical experiments (described in the following section) seem to suggest that, for natural problem instances, volatility quickly drowns out the value of forecasts, and hence $\beta = 1$ tends to perform best in most instances where demand uncertainty is present.

3. A practical improvement to the algorithm that does not alter the guarantees moves inventory made available due to 'under-utilization' by the deterministic system to the stochastic system. This compensates for the lack of inventory 'sharing' between the re-optimized and deterministic systems. In particular, while leaving all of the details of the algorithm unchanged, we define the dynamic for $\hat{x}^{\text{R}}_t$ according to

$$\hat{x}^{\text{R}}_{k,t} = \hat{x}^{\text{R}}_{k,0} - \int_0^t \mathbf{1}_{\hat{x}^{\text{R}}_{k,t}>0} \sum_e \Lambda_{i(e),t} A_{k,e} \hat{z}^{\text{R}}_{e,d(t)} dt + \int_0^t \sum_e (\lambda_{i(e),t} z^{\text{D}}_e - \Lambda_{i(e),t} \hat{z}^{\text{D}}_{e,t}) dt.$$

1.5 Experiments.

We focus our experiments on instances of the Ad Display problem. We present results for two sets of experiments. The first set consists of synthetic instances; the purpose of this set of experiments is to gauge performance across a variety of parameter regimes. The second set of experiments is derived from an actual allocation problem and (real) traffic from an ad network.

1.5.1 A Generative Family of Instances.

We characterize problem instances along two dimensions, namely:

1. **Load Factor:** We define load factor as the quantity

$$\mathrm{LF} \triangleq \frac{\mathsf{E}[\sum_i \int_0^T \overline{\Lambda}_{i,t} dt]}{\sum_k x_{k,0}}.$$

   This is a natural measure of the scarcity (or abundance) of a resource relative to demand.

2. **Coefficient of Variation:** We measure the relative volatility in demand via the quantity

$$\mathrm{CV} \triangleq \frac{\sqrt{\mathrm{Var}[\sum_i \int_0^T \overline{\Lambda}_{i,t} dt]}}{\mathsf{E}[\sum_i \int_0^T \overline{\Lambda}_{i,t} dt]}$$

We consider the following generative family of instances:

1. **Topology:** We set $I = 30$ and $A = 30$. An edge connecting a given source i to a given sink a exists independently with probability 0.1. In essence, this prevents a scenario where a given unit of traffic can be used by essentially all sinks.

2. **Resources, Prices and Horizon:** We set $x_{k,0} = 100$. The price associated with a given edge is generated according to an independent uniform distribution on $[0, 100]$. We set $T = 1$.

3. **Demand:** We use an Ornstein-Uhlenbeck process to generate Λ_t. In particular, we set

$$\overline{\Lambda}_t = \lambda_t + \sigma \int_0^t e^{(s-t)} dZ_s$$

   where Z_s is standard I dimensional Brownian motion.

   We generate λ_t as follows. First, we draw a vector λ uniformly from $[0, 100]^I$. Depending on whether the experiment tests the no-forecast algorithm or not, we then either set $\lambda_{i,t} = M\lambda_i$ (for the no forecast case) or we set $\lambda_{i,t} = M\lambda_i(1 + 2t)$

with probability 1/2 and $\lambda_{i,t} = M\lambda_i(1-t/2)$ with the remaining probability (in the case where we test the algorithm incorporating forecasts). In both cases, M is selected so that the load factor LF takes the appropriate value for the instance we wish to generate.

We set σ so that CV takes the appropriate value; we use bisection to find the appropriate value of σ here.

Finally, we generate the stochastic process via the natural discretization of the continuous time process defined above; in particular, we use the recursion

$$\overline{\Lambda}_{n\Delta} = \lambda_{n\Delta} + (1-\Delta)(\overline{\Lambda}_{n-1\Delta} - \lambda_{n-1\Delta}) + \sigma\epsilon_n$$

where $\Delta = 1/100$ and ϵ_n is a zero mean normal with variance Δ.

1.5.2 Results for Instances from Generative Family.

We first consider problems generated from the family described above with λ set to a constant. We consider 30 ensembles of instances. Each ensemble differs in the parameters (LF, CV) and itself contains 30 individual problem instances. We employ the re-optimization scheme designed for scenarios where no forecast is available, taking the re-optimization frequency, N, to be 100. We use as our upper bound the quantity $\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]$.

The results of these experiments are summarized in Table 1.1. The 95% confidence intervals for reported figures are within $+/-5\%$. Here we make several observations: First, performance relevant to the clairvoyant upper bound is consistently good; it is at least within 80% of this upper bound, and, frequently, well within 90%. We observe some performance degradation in regimes of extremely high volatility. Also, problems with low load factors (i.e. where demand is scarce) appear to be more challenging for the scheme. This is somewhat intuitive if seen from the perspective that in such a regime, one will not be able to consume each component of x_0 with its 'optimal' impression.

Next, we consider a similar ensemble of problems, but with λ_t allowed to be time varying (in the manner described in the previous section). We employ the re-optimization scheme that utilizes forecasts. This scheme requires a tuning parameter β. We consider two sets of experiments; the first uses the 'robust' choice of β ($\beta = 0.406$) identified via Theorem 1.4.7; the results for this scheme are described in Table 1.2. We then allow the algorithm designer to choose β from the set $\{0, 0.2, \ldots, 1.0\}$, and report performance for the best of these values in Table 1.3. Again, we set $N = 100$ and use $\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right]$ for upper bound comparisons in all experiments. In addition, we note that we employ the 'inventory sharing' improvement described following Theorem 1.4.7.

The results for these experiments are described in Tables 1.2 and 1.3 respectively. While it is not displayed here, the optimal β in essentially all cases for Table 1.3 where demand was volatile was $\beta = 0$; i.e. the simple re-optimization scheme that ignores forecasts altogether. We see qualitatively similar results to the case where λ is a constant. Moreover, when one allows the user to optimize β, performance is essentially as good as the constant λ case.

Lastly, we test the no-drift algorithm by varying N in the set $\{1, 2, 5, 10, 50, 100\}$, while keeping the load factor constant at 1. As expected, the impact of discretization increases with volatility. However, performance is often satisfactory even for $N = 1, 2$ or 5. Increasing N from 1 to 5 does not seem to have a reliable effect, however increasing N from these lower ranges to 100 monotonically and markedly improves performance. The results of the experiments are summarized in Table 1.4.

Table 1.1: No-drift algorithm performance vs upper bound. The load factor is along the vertical axis and the CV is along the horizontal axis.

Load Factor/CV	0	0.5	1	2.5	5	10
0.1	100.00%	99.97%	99.00%	92.04%	79.06%	81.00%
0.5	100.00%	98.75%	97.03%	90.32%	82.31%	82.63%
1	100.00%	99.15%	96.86%	91.08%	86.51%	84.54%
2	100.00%	99.91%	99.54%	97.34%	91.98%	87.88%
5	100.00%	99.92%	99.68%	98.85%	97.23%	94.26%

Table 1.2: Drift algorithm performance vs upper bound for $\beta = 0.406$. The load factor is along the vertical axis and the CV is along the horizontal axis.

Load Factor/CV	0	0.5	1	2.5	5	10
0.1	100.00%	74.45%	67.80%	62.79%	62.54%	74.35%
0.5	99.91%	91.66%	84.59%	73.45%	69.35%	74.80%
1	99.74%	95.46%	91.03%	81.54%	76.95%	78.08%
2	99.86%	97.63%	95.44%	91.13%	86.34%	84.37%
5	99.95%	99.06%	98.46%	97.23%	95.15%	92.43%

Table 1.3: Drift algorithm performance vs upper bound for optimized β. The load factor is along the vertical axis and the CV is along the horizontal axis.

Load Factor/CV	0	0.5	1	2.5	5	10
0.1	100.00%	99.95%	98.90%	92.18%	79.17%	80.98%
0.5	100.00%	99.31%	97.87%	90.80%	82.34%	82.55%
1	100.00%	98.95%	97.23%	90.70%	85.81%	84.28%
2	100.00%	99.17%	97.84%	94.60%	90.10%	87.21%
5	100.00%	99.80%	99.56%	98.71%	96.57%	93.22%

Table 1.4: No-drift algorithm performance vs upper bound for various re-solving frequencies. The number of re-optimization intervals is along the vertical axis and the CV is along the horizontal axis.

N/CV	0	0.5	1	2.5	5	10
1	100.00%	99.99%	98.59%	94.49%	83.95%	74.46%
2	100.00%	99.94%	98.09%	94.09%	83.37%	74.79%
5	100.00%	99.96%	98.03%	93.71%	83.54%	75.96%
10	100.00%	99.99%	98.39%	94.74%	85.31%	78.91%
50	100.00%	99.99%	98.85%	96.13%	89.5%	85.15%
100	100.00%	99.99%	99.15%	96.86%	91.08%	86.51%

1.5.3 A Real Instance From An Ad Platform.

We also consider experiments on a real instance of the Ad Display problem described here. In particular, we have data from a mobile ad platform for a single day of traffic. On this day, the platform served 240 distinct advertisers using impressions from a highly heterogeneous pool. Each campaign can only be served by a subset of this traffic based on a set of parameters. Payments are uniform across compatible traffic for a given advertiser; i.e. $p_{(i,a)} \in \{0, p_a\}$. We aggregated impressions based on their originating website/ mobile application resulting in a total of about 40 traffic sources. The number of arrivals during the 24 hour interval varies from 4 million for the largest inventory type, to 10 thousand for the smallest. The coefficient of variation varies from 0.4 to 1.5 and there are noticeable intra-day, cyclical trends in arrival rates (for example, arrivals have larger intensity in the morning and evening, versus late night). The average in-degree of the implied bipartite graph from inventory to campaigns is 4.5, the average out-degree is 18 and the load factor of user requests to campaign capacities is 1.5.

We chose to use the re-optimization scheme without forecast inputs and set the re-optimization frequency to $N = 24$ (i.e. once every hour). Solutions to the LP were interpreted probabilistically, and we used as our benchmark the clairvoyant bound (which in this case corresponds to assigning the day's traffic after it has been realized). Our scheme earns revenues that are within 99.3% of this benchmark which is encouraging. Moreover, performance within 99% of the benchmark was maintained under two stress tests, namely scaling down the load factor to 0.75, and introducing heterogeneous prices (where we multiplied prices by a standard uniform random variable).

1.6 Conclusion.

Our main contribution has been to develop a simple, easy to interpret algorithm that can efficiently solve a large class of dynamic allocation problems. Our method is robust (as witnessed by worst case guarantees) and in the event that demand volatility (or, equivalently, deviations of demand from its forecast) is not large, the scheme is simultaneously optimal. Practical experiments have shown that the approach is promising both in terms of performance and practicality. At a somewhat more abstract level, we believe this work contributes to the (theoretically) poorly understood area of model predictive control, and as such, we believe the simple analytical tools developed here (the balancing property) may provide value in other contexts.

Our scheme also motivates some of the research questions that constitute the rest of this thesis. In particular:

1. Chapter 2 deals with a non-standard NRM model where instead of an exogenous reward p_e on each edge of the graph, the mechanism governing the reward makes p_e depend dynamically on the prevailing landscape of resource consumption. Such a mechanism in encountered in the sponsored search varieties of online advertising, where the amounts the advertisers are charged are determined through a Generalized Second Price mechanism.

2. Our scheme relies on the ability to measure the instantaneous demand Λ_t at the start of each reoptimization interval. This may in itself be a difficult estimation task. In a highly volatile regime where the reoptimization interval becomes small, the number of samples available to estimate the I different instantaneous rates is substantially smaller than I. Chapter 3 deals with how to accomplish this learning task efficiently.

3. The scheme also relies on the ability to quickly resolve a linear program. Such LP based techniques have not been adopted to this point in online advertising due to the prohibitively large size of the LPs that must be solved in practice. Chapter 4 tackles this computational challenge.

There are many directions that merit further attention beyond the scope of this thesis:

1. Dynamic Prices: The NRM models that we consider in this chapter and the next capture contract based variants of online advertising where advertisers commit

to their bids for the entirety of the contract lifetime. The recent emergence of ad exchanges calls for NRM models that incorporate the strategic behavior of advertisers, which may be dynamically changing their bids over the lifetime of the contract.

2. Fairness: In practice, ad networks are also concerned in meeting certain quality and fairness metrics for their advertisers, since advertisers who do not receive an appropriate level of service are unlikely to purchase contracts in the future. It would be interesting to explore extensions of our scheme where fairness constraints would be incorporated into the model.

3. Inventory Balancing: In addition to being a potentially valuable theoretical tool in other contexts, the inventory balancing property of our re-optimization scheme permits some interesting (un-intended) applications. From a practical perspective, in the case of the Ad Display problem, it automatically results in allocations that satisfy what are commonly called 'pacing' constraints where resources (eg. budgets) may be consumed at most at some pre-specified uniform rate over time. From a theoretical perspective, a future direction of research is to see whether simpler heuristics (compared to resolving LPs) that preserve the inventory balancing property could yield comparable performance.

Chapter 2

Optimal Allocation for Generalized Second Price AdWords

2.1 The Generalized Second Price AdWords Problem.

The focus of Chapter 1 was a "first price" model of network revenue management. For online advertising, this corresponds to a contract based system where the advertiser declares a bid for an impression and upon the allocation of this impression, the advertiser is charged their own bid. This is typically the pricing scheme that characterizes display advertising systems, but not the only pricing mechanism encountered in the online advertising industry. In contrast to display ads, most sponsored search systems use an alternative mechanism referred to as a Generalized Second Price (GSP) scheme where the amount an advertiser is charged is a function of the bids of other competing advertisers, as well as their remaining budgets. The rationale behind this pricing scheme, which was first introduced with Google AdWords sponsored search system, is that it generalizes the idea of a Vickrey auction (Krishna (2002)) and hence hopefully preserves its incentive computability properties, thus incentivizing advertisers to declare their true valuations for an impressions to the ad network. Unfortunately, it has since been shown in Varian (2007); Edelman et al. (2005) that GSP is non-truthful.

From a revenue management perspective, it is perhaps most natural to regard GSP as an *endogenous price* NRM problem. Algorithmically, handling this endogeneity is significantly more challenging when compared to the exogeneous, "first price "variant described in Chapter 1. Hence not surprisingly, both the computer science and RM communities have previously focused on the first price approximation to this allocation problem which ignores the GSP pricing mechanism. This approximation, which also fits the NRM model we considered in the previous chapter, is referred to in the literature as AdWords. Accordingly, we will henceforth refer to the realistic, endogenous version as *Generalized Second Price Adwords* and note that, a priori, it does not fit the model from Chapter 1. This chapter will exploit the balancing properties of our model predictive control scheme described in Lemma 1.3.4 to yield an algorithm that extends our old revenue guarantees to GSP AdWords.

2.1.1 Literature Review.

Much of our literature review from the previous chapter is relevant for the simplified AdWords version mentioned above. Of particular note is the adversarial arrival primal dual schema due to Mehta et al. (2005) and Buchbinder and Naor (2009), as well as the bid price learning schema of Devanur and Hayes (2009), which assumes a random order model of keyword arrivals. There been fewer results that deal with the realistic GSP AdWords version. Abrams et al. (2007) formulate the offline version of problem as a linear program, and propose column generation methods to solve this program, but do not provide an algorithm for the online version with uncertain impression arrivals. The only result for this online version that we are aware of is due to Goel et al. (2010), who propose a greedy heuristic with a 1/3 worst case competitive ratio, but no asymptotic guarantees versus the forecasts.

While GSP AdWords have not been extensively studied from the above online algorithms and dynamic allocation perspective, there is a second, so far separate and much more intensely studied, stream of literature focusing on the game theoretic properties of the GSP mechanism. While we study a model where we need to allocate a highly heterogenous pool of impressions to another highly heterogenous pool of advertisers, but abstract away any game theoretic complexities regarding advertising bidding behavior, this stream of literature abstracts the allocation problem by focusing on simplified models where a single impression type is auctioned off, but tries to gain game-theoretic insight into the advertiser bidding behavior. In particular, Varian (2007) and Edelman et al. (2005) point out the non-truthfulness of the GSP mechanism even in a single shot setting, characterize an envy-free family of its bidding equilibria, and show this family contains an equilibrium with revenues equivalent to the VCG outcome. The more recent works of Iyer et al. (2012) and Balseiro et al. (2012) obtain mean field equilibria characterizations of the bidding landscape in repeated auctions with budgeted advertisers – this still abstracts away the heterogeneity in impression types and is thus disparate form the setting that we are concerned with.

2.1.2 Model.

The Generalized Second Price AdWords problem differs from the Ad-Display problem described in Section 1.2.2 in several key features:

1. Multiple slots per impression: for every impression there are $k \geq 1$ "slots" to be assigned to advertisers. Each slot has a non-increasing quality factor θ_l, with $\theta_1 = 1$. Advertiser a declares a bid $b_{i,a}$ for the top slot for impression i, and its bids for the lower slots are scaled down by the quality factor. (This captures the empirical observation that the ad shown in the top-most slot is more likely to be clicked on than the same ad placed in a lower slot.)

2. Advertiser budgets: for AdWords problems, budgets are specified in dollar terms rather than impression counts. We denote the starting budget vector of advertisers by B_0.

3. Generalized Second Price allocation mechanism: this describes how much advertiser budget the ad network charges from assigning an advertiser's ad in slot l for impression i. The mechanism is the following – upon the arrival of an

impression of type i, the ad network selects a subset of size at most $k+1$ advertisers. We call this subset a "slate". The selected advertisers are assigned slots beginning from the top-most downwards to the k-th. The advertiser with the l-th largest bid is assigned slot l, but they are charged the $(l+1$-th highest bid (hence the Second Price name); the $(k+1)$-th advertiser does not receive a slot and is selected simply to determine the amount paid by the advertiser in the k-th slot.

There are two variants of the GSP mechanism that have been proposed:

The non-strict model: at time t, the ad network is allowed to pick and allocate impressions to a slate composed of advertisers whose remaining budgets $B_t = 0$. This is a model in which advertisers who are effectively inactive due to having exhausted their budgets can still be used by the ad network to set prices for active advertisers – for this reason, this is generally not an accepted GSP model. However, it is attractive in that it corresponds to an exogenous price NRM problem, as prices are set at time 0 and do not depend on the evolution of the budgets in the system. Consequently, the non-strict version fits the model described in Section 3.2 and therefore the theorems regarding the performance of the model predictive control policy transfer directly.

For non-strict GSP, the mapping to our resource allocation graph from Section 3.2 is as follows: while the source nodes still correspond to impressions, the sink nodes now correspond to slates; the number of slates is $\binom{A}{k+1}$, where A is the number of advertisers. We remind the reader that we use the notation $j(e)$ to denote the slate that edge e is incident to. Additionally, we define the notation $a(j,l)$ to denote the advertiser that has the l-th highest bid in slate j. For edge e, we set

$$p_e = \sum_{l=1}^{k} \theta_l b_{i(e),a(j,l+1)}$$

and

$$c_{a,e} = \theta_l b_{i(e),a} \text{ if } a = a(j(e),l) \text{ and } 0 \text{ otherwise.}$$

The strict model: this is a model where slates may dynamically become disallowed, since the ad network may only permit advertisers with positive remaining budgets can participate in the auction. This is the model that has garnered adoption in the industry, due to it being perceived as fairer than the non-strict version. It is a model with endogenous prices, since the prices associated with a particular slate depend on whether all advertisers within the slate have remaining budget. In particular, consider an edge e from an impression type i to a slate j and define the event $I_{e,t} = \{B_{a,t} > 0, \forall a \in j(e)\}$. At time $t \in [0, T]$, the rewards are:

$$p_e^{\mathrm{GSP}}(t) = \sum_{l=1}^{k} \theta_l b_{i,a(j(e),l+1)} 1(I_{e,t}) \tag{2.1}$$

where $j = \{a_1, \ldots, a_{k+1}\}$ and we assume without loss of generality that the advertisers are indexed in order of decreasing bids. Similarly, the budget consumption rates are:

$$c_{a,e}^{\mathrm{GSP}}(t) = \begin{cases} \theta_l b_{i(e),a(j(e),l+1)} 1(I_{e,t}) & \text{if } a = a(j(e), l), 0 \le l \le k \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Note that the dependence on B_t is non-linear, which in fact may suggest our previous model predictive control scheme cannot generalize to this setting.

We mention that the practice of choosing a slate of advertisers out of the possibly much larger set of advertisers who have positive bids for an impression is known in the industry as "throttling". Sponsored search systems use throttling as a means to give the network freedom to optimize the allocation process. If no throttling was done and all interested advertisers were allowed to participate auction, the trajectory of the allocations would be completely set by the GSP rule and the ad network would have no way of controlling the allocation process.

2.2 An Extension of Model Predictive Control to Generalized Second Price AdWords.

In this section, we extend the re-optimization scheme from Chapter 1 to online Generalized Second Price AdWords in the strict model. As stated before, the main barrier to directly applying the result lies in the endogeneity of prices in the strict GSP setting. However, note that endogeneity is not an issue in the offline version of the problem for either the strict or non-strict models. Indeed, if impression frequencies are known ahead of time, an LP formulation achieves uniform advertiser budget consumption over time, ensuring that $I_{e,t} = 1$ for all t and hence every slate remains valid until the end of the time horizon. However, for the online version of the allocation problem, this task is much harder – as the the frequencies of impression arrivals vary over time, it is not clear the system can be controlled in a way that ensures smooth budget consumption. This suggests that (i) the rate at which budget is consumed can be crucial in analyzing the performance of a policy in this system, and (ii) that the model predictive control policy from the previous chapter, which admits precise characterizations of this rate, might be a tractable policy to analyze in this case.

Indeed, our algorithm does exploit the balanced budget consumption property of the model predictive control scheme in the following way. First, we look at the system in the non-strict model where the dependence of the GSP prices on B_t has been relaxed; we apply the model predictive control scheme to this hypothetical system to yield a control which we use in the real system. Secondly, we make use of the Balancing Lemma to show that following the prescriptions given by our relaxation does not lead to any revenue loss when we apply the strict GSP constraints ex post.

For the sake of completeness, we define a model for strict GSP that is analogous to the one from Section 3.2 except in the fact that the rewards and resource consumptions are allowed to vary dynamically with time as defined in (2.1) and (2.2).

Control and Dynamics. As before, our continuous control x is reevaluated at N discrete time intervals $\{0, T/N, 2T/N, \ldots, T\}$. The control that is calculated at time iT/N remains in effect over $[iT/N, (i+1)T/N]$ and is specified by a vector $x \in \mathcal{X}$, where $\mathcal{X} = \left\{x \in R_+^E : \sum_{e:i(e)=i} x_e \leq 1 \ \forall i\right\}$.

The state of the system at time t is given by the level B_t of advertiser budgets that remain at t. The evolution of B_t is specified by the differential equation:

$$\frac{d}{dt} B_{a,t} = -\sum_e c_{a,e}^{\mathrm{GSP}}(t) x_{e,d(t)}$$

for all a. Here $d(t) = \max_{\{i: iN/T \leq t\}} iN/T$.

Optimum value. The optimization problem is to find an admissible control policy $\{x_t\}$ that maximizes the overall revenues:

$$\max_{\{x_t\}} \quad \mathsf{E}\left[\int_0^T \sum_e p_e^{\mathrm{GSP}}(t) \Lambda_{i(e),t} x_{e,d(t)} dt\right] \tag{2.3}$$

We denote the optimal value to this optimal control problem by $J_{\mathrm{GSP}}^{*,N}(B_0)$.

For our proof, we will consider a hypothetical, non-strict correspondent of this system, where $p_e^{\text{non-strict-GSP}}(t) = p_e^{\mathrm{GSP}}(0)$ and $c_{a,e}^{\text{non-strict-GSP}}(t) = c_{a,e}^{\mathrm{GSP}}(0)$ for all t. Let us define a policy which simply applies the model predictive control resolve technique from Section 1.3 in this hypothetical system:

$$\begin{aligned} \max \quad & \sum_e p_e^{\mathrm{GSP}}(0) x_e \Lambda_{i(e)} \cdot (T-t) \\ \text{subject to} \quad & \sum_e c_{a,e}^{\mathrm{GSP}}(0) x_e \Lambda_{i(e)} \cdot (T-t) \leq B_a^+ \quad \forall\, a, \\ & x \in \mathcal{X}. \end{aligned} \tag{2.4}$$

We define the revenues garnered by this heuristic in the non-strict GSP system

$$\begin{aligned} J_{\text{non-strict-GSP}}^{\mathrm{R},N}(B_0) &= \int_0^T \sum_e p_e^{\text{non-strict-GSP}}(t) \Lambda_{i,t} x_{e,t}^{\mathrm{R}} dt \\ &= \int_0^T \sum_e p_e^{\mathrm{GSP}}(0) \Lambda_{i,t} x_{e,t}^{\mathrm{R}} dt. \end{aligned}$$

Moreover, the linear program above also gives an offline upper bound on $J_{\mathrm{GSP}}^{*,N}(B_0)$

equal to

$$J^{\mathrm{UB}}_{\mathrm{GSP}}(B_0) = \mathrm{LP}\left(0, \frac{1}{T}\int_0^T \Lambda_t dt, B_0\right).$$

This is due to the fact that the above quantity is an upper bound on $J^{*,N}_{\text{non-strict-GSP}}$ by Proposition 1.3.1 and, also, on $J^{*,N}_{\mathrm{GSP}}$ since the prices p^{GSP} are by definition non-increasing in t.

The following proposition applies Theorem 1.3.10 to the non-strict system:

Theorem 2.2.1 *Consider demand processes $\{\Lambda_t\}$ satisfying Assumption 1.2.1, with $\lambda_t = \lambda$ for all t. In addition assume that $\sigma_t/\lambda \leq \sqrt{2\pi}\nu$. We then have, assuming an initial inventory of x_0:*

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}}_{\textit{non-strict-GSP}}(B_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\textit{GSP}}(B_0)\right]} \geq \max\left\{0.342, \frac{1}{1+\nu} - \frac{B}{1+\nu}\left(\exp(-1/4\pi\nu^2) + 0.853\right)\right\}.$$

Let us now return to the strict GSP AdWords model; in the following, we make use of Lemma 1.3.4 to show that the control computed assuming a non-strict model will never exhaust advertiser budgets mid-way through the time horizon of the problem. Hence, across all sample paths of impression arrivals, all slates will remain active, implying equivalence between the strict and non-strict versions of the problem.

Lastly, define

$$J^{\mathrm{R},N}_{\mathrm{GSP}}(B_0) = \int_0^T \sum_e p_e^{\mathrm{GSP}}(t)\Lambda_{i,t} x^{\mathrm{R}}_{e,t}\mathbf{1}(I_{e,t})dt,$$

the revenues garnered in the real system using the sequence of re-optimized controls x^{R} for the non-strict GSP problem. In the following we will show that

$$\liminf_N J^{\mathrm{R},N}_{\mathrm{GSP}}(B_0) = \liminf_N J^{\mathrm{R},N}_{\text{non-strict-GSP}}(B_0).$$

Intuitively, this is due to the fact that the Balancing Lemma guarantees that

$$\liminf_N p_e^{\mathrm{GSP}}(t) = p_e^{\text{non-strict-GSP}}(0), \forall e, t$$
$$\liminf_N c_{a,e}^{\mathrm{GSP}}(t) = C_{a,e}^{\text{non-strict-GSP}}(0), \forall a, e, t.$$

We are now ready to state and prove the validity of our algorithm in the strict model.

Theorem 2.2.2 *Consider demand processes $\{\Lambda_t\}$ satisfying Assumption 1.2.1, with $\lambda_t = \lambda$ for all t. In addition assume that $\sigma_t/\lambda \leq \sqrt{2\pi}\nu$. We then have, assuming an initial inventory of x_0:*

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R},N}_{GSP}(B_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{GSP}(B_0)\right]} \geq \max\left\{0.342, \frac{1}{1+\nu} - \frac{\nu}{1+\nu}\left(\exp(-1/4\pi\nu^2) + 0.853\right)\right\}$$

Proof We have

$$\begin{aligned}
\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R},N}_{\mathrm{GSP}}(B_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\mathrm{GSP}}(x_0)\right]} &\geq \liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R},N}_{\text{non-strict-GSP}}(B_0)|I_{e,t}>0,\ \forall t,e\right]\mathbb{P}\left[I_{e,t}>0,\ \forall t\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\text{non-strict-GSP}}(x_0)\right]} \\
&\quad + \liminf_N \frac{\left(\sum_a B_{a,0}\right)\left(1-\mathbb{P}\left[I_{e,t}>0,\ \forall t,e\right]\right)}{\mathsf{E}\left[J^{\mathrm{UB}}_{\text{non-strict-GSP}}(x_0)\right]} \\
&\geq \liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R},N}_{\text{non-strict-GSP}}(B_0)|I_{e,t}>0,\ \forall t,e\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\text{non-strict-GSP}}(x_0)\right]} \liminf_N \mathbb{P}\left[I_{e,t}>0,\ \forall t,e\right] \\
&= \liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R},N}_{\text{non-strict-GSP}}(B_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\mathrm{GSP}}(x_0)\right]} \\
&\geq \max\left\{0.342, \frac{1}{1+B} - \frac{B}{1+B}\left(\exp(-1/4\pi B^2) + 0.853\right)\right\},
\end{aligned}$$

where the first inequality follows by taking conditional expectations and noticing that the total revenues are upper bound by the sum of advertiser budgets, the second inequality follows from the fact that $\liminf a_n b_n \geq \liminf a_n \liminf b_n$ and the fact that the Balancing Lemma implies that $\liminf_N \mathbb{P}\left[I_{e,t}>0,\ \forall t,e\right] = 1$, and the last inequality follows from Proposition 2.2.1.

We end this section by noting several features of our result. Firstly, for simplicity here, we have only given a proof of validity for processes with $\lambda_t = \lambda$ for all t. As we have done in the previous chapter, using β reoptimization would allow us to relax this assumption at the expense of a worse constant factor. Secondly, the algorithm and its analysis is applicable to any pricing mechanism (beyond GSP) where the price endogeneity originates from the restriction that only advertisers with un-exhausted budgets are permitted to set prices, i.e. mechanisms where $I_{e,t}$ is a sufficient statistic for $p_e(t)$ and $c_{a,e}(t)$.

2.3 Conclusion.

In this chapter of the thesis, we have extended our model predictive control scheme to the Generalized Second Price AdWords setting. Compared to first price versions of the AdWords problem, our model captures the realistic GSP price mechanism used by most sponsored search systems. Quite interestingly, the balancing property of our scheme lies at the heart of our reduction to the first price case.

In terms of future directions, an inconvenience of our scheme lies in the exponential dependence in the number of slots. It would be quite interesting to see whether some dimension reduction in the number of slates that need to be considered. One possible direction could to be to use a bid-price control approximation that could be calculated from a linear program of reduced dimension.

Part II

High Dimensional Learning

Chapter 3

Demand Learning for Display Advertising

3.1 Introduction

The explosion in recent years in web and mobile online advertising volume has brought to the fore a host of dynamic allocation problems in the same genre as the network revenue management (NRM). A modern version of an NRM problem which we will refer to as the Ad Display problem occurs in online display advertising markets.

Informally, the Ad Display problem is formulated as follows: an ad network receives an online sequence of user arrivals called impressions, each associated with a vector of features in some d-dimensional space $\mathcal{X}$. Each vector $x \in \mathcal{X}$ identifies an impression type. Upon the arrival of an impression, the network must decide whether to allocate it to one of m competing advertisers. Typically, the network and the advertisers strike contracts for ad campaigns where the advertiser commits to paying some fixed amount r_a ("bid") for any impression belonging to a set of compatible types $\mathcal{X}_a \subseteq \mathcal{X}$ for up to a total amount of impression deliveries (the "budget"). The ad network's goal is to maximize its overall revenues from allocating impressions to advertisers over the finite time horizon of the various campaigns, subject to advertiser budget constraints.

We emphasize the fact that we describe a problem with the special structure that advertisers pay a *fixed* bid for any compatible impression type; this problem is also referred in the literature as *vertex-weighted matching* (Aggarwal et al. (2011)) and differs from the more general network revenue management structures considered in the previous chapters where advertisers could potentially place different bids for the various types in $\mathcal{X}_a$. This special structure mirrors the contracts that occur in display advertising, where the contracts written between advertisers and the ad network do indeed specify a single fixed bid for an entire set of acceptable types. In practice, advertisers who may want to express heterogenous impressions valuations may contract the network to concurrently run several campaigns, such as a campaign that targets "low quality", cheap impression types and another campaign that targets "high quality" types for which the bid may be significantly higher.

The revenue management literature has extensively studied network problems of this flavor where arriving "customer types" (impression types in our setting) must be matched to "products" (ads) which consume a set of limited "resources" (advertiser budgets) in a way that maximizes the seller's overall revenues. The universal prescription is the following: with *a priori* knowledge of the distribution of customer demand, one can solve a linear optimization problem known as the Deterministic Linear Program (DLP) and obtain a control known as a *bid price* policy Gallego and van Ryzin (1997); Talluri and Ryzin (1998), which assigns a customer to the product with maximum revenue discounted by the total economic opportunity cost of the resources required to "build" the product. This bid price policy is known to be optimal under mild assumptions on the instance structure; more recently it has been discovered that, even in the case of changing demand, the bid price policy is optimal as long as the DLP is resolved at roughly the same timescale as the scale of demand changes Ciocan and Farias (2012).

In typical applications, the assumption that customer demand is known a priori is fairly innocuous since the relatively small number of customer types makes it simple to learn the demand distribution quickly – in a "high volume" regime where the number of samples is on the same scale as the number of types, an empirical demand estimator performs extremely well essentially via a Strong Law of Large Numbers argument. In contrast, the present paper is concerned with settings where the demand distribution is *extremely heterogenous and high-dimensional.* This is the case with Ad Display where an impression type is described by a detailed information vector including user location, demographics, past browsing behavior and various other user attributes. To give the reader a sense of the dimension of the problem in practice, we note that modern platforms such as Facebook or Google AdSense currently use $d = 30$ to 70 such attributes; the resulting demand type space $\mathcal{X}$ has dimension $n = \exp(d) = 2^{70}$.

The reason this regime is fundamentally different from the first is the following: in order to be practical at realistic scales, an ad serving algorithm needs to calibrate the demand distribution at time scales on the order of 10 minutes. Over a 10 minute interval, there is not enough volume of observed impression arrivals is not enough to even cover $\mathcal{X}$, which would be necessary to learn via a direct application of the Strong Law of Large Numbers. At first glance, it seems impossible to learn a multinomial distribution with support n using significantly fewer samples than n and it is not a priori clear that a better scaling is achievable. Additionally, this difficulty is supported by what we observe in practice: most advertising systems that are deployed in production in fact eschew the usage of demand forecasts in the interest of tractability and use policies that are agnostic to the demand model even at the expense of leaving revenues on the table. See for example, the primal dual algorithms of Mehta et al. (2005); Buchbinder and Naor (2009); Feldman et al. (2009); it is interesting that these policies also take the form of a (different) bid price policy from the one yielded by the DLP.

In light of the above, the question we ask is the following: is there an algorithm that learns a nearly optimal bid price policy with a more graceful sample complexity dependence in terms of $\mathcal{X}$ than the $O(n)$ that is suggested by the Strong Law of Large Numbers? Our results are the following:

1. We show that Ad Display is a special class of network revenue management problems for which the sample complexity indeed scales gracefully. In particular, we show that one can learn a bid price policy which, with high probability,

captures $1 - \epsilon$ of OPT with

$$N = \frac{m \log(nm)}{\rho^2 \epsilon^2},$$

impression arrival samples, where ρ is an instance-specific quantity that relates the optimal rate of revenue per impression to the maximum advertiser bid. While we cannot arrive at a uniform bound on ρ across all problem instances, we show that reasonably mild regularity conditions imposed on the family of instances yields a ρ that scales like the ratio between the maximum advertiser bid and the advertiser bid. Hence, under these conditions, our bound scales *linearly* of the impression type space dimension $d = \log n$, rather than the expected linear in n dependence.

2. Moreover, the result nearly matches the lower bound for our algorithm, as we show that our analysis is tight up to a factor of $\log(nm)$.

As will become clear in the following section, we achieve these results by building a simple empirical estimator $\hat{\mu}$ of the true demand (i.e. impression) distribution μ which has extremely sparse support versus μ ($\log(n)$ versus n); one could hence view the algorithm as implicitly recovering a latent low dimensional representation of the real distribution.

3.1.1 Literature Review

Clearly, with such a stringent condition on the sample complexity, it is hopeless to arrive at a uniformly sharp resolution over all of $\mathcal{X}$. Instead, our learning goal is to arrive at an estimate over $\mathcal{X}$ that yields an approximately optimal control to the allocation problem; put another way, we require a forecast that we can plug into the DLP and achieve roughly the same objective value we would have garnered with knowledge of the true forecast. One natural candidate scheme to accomplish this is to sample N impressions and feed the resulting empirical distribution into the DLP to obtain estimate bid prices; these estimates can then be used as hopefully accurate controls to drive the allocation decisions over the remaining life of the problem.

This natural sampling based approach has been analyzed in a sequence of papers, Devanur and Hayes (2009); Feldman et al. (2010); Agrawal et al. (2014); Molinaro and Ravi (2012), in the context of the AdWords problem, and subsequently, of a general NRM (or resource allocation) framework. These papers consider a random permutation model of demand arrivals, in which the assumption is that the total number of arriving impressions is known in advance, but the order of the arrivals is uniformly chosen over all possible permutations. We note that, while more general than the i.i.d. model we will consider in our model, the random permutation model is still limited to describing stationary demand distributions.[1] The guarantee that these papers establish is the following: with ϵn samples, the algorithm is $1-\epsilon$ optimal as long as the minimum advertiser budget B_{min} satisfies a certain scaling $g(n, m, \epsilon)$. As we discuss in Section 3.5.1, in order to achieve a polynomial dependence on d and m by appealing to these existing analyses, one needs restrictive conditions on g, leading significantly worse sample complexity.

The algorithm we employ is essentially identical to the one used in the random permutation model literature cited above. However, we restrict our analysis to Ad Display problems rather than the general network revenue management problem. This is a crucial assumption: we leverage the particular pricing structure (a single advertiser bid for all compatible impressions) that is idiosyncratic to Ad Display to achieve our improved result and it is unlikely that our analysis could be generalized beyond the Ad Display model.

3.2 Model and Algorithm

Impressions model: We consider a discrete T-time period model in which, at each time step $1 \leq i \leq T$ exactly one impression arrives to the ad network. Let $\mathcal{X} \subseteq \mathbb{R}^d$

[1] In fact, de Finetti's theorem establishes that the exchangeable distributions of demand in a random permutation model are in fact independent conditioned on a latent variable.

be a discrete feature space with each point in $\mathcal{X}$ describing an impression *type*.[2] We assume that there exists an *unknown* distribution $\mu : \mathcal{X} \to [0,1]$ from which each of the arriving T impressions are sampled i.i.d.; upon arrival, each impression is assigned a type in $\mathcal{X}$ according to μ. Let $X_1, \ldots, X_T$ be the sequence of random variables denoting the types of the arriving impressions.

Advertiser model: There are m advertisers with budgets $B \cdot T \in \mathbb{R}^m_+$. B can be interpreted as the budget the contract specifies per unit of impression. Each advertiser a is endowed with a characteristic set $\mathcal{X}_a \subseteq \mathcal{X}$ such that a's bid for impression type x is $r_a \mathbb{1}(x \in \mathcal{X}_a)$ for some positive r_a. In order to provide exact constants in our bounds, we will assume $m \geq 4$ throughout.

Let $\mathrm{OPT}(T, TB)$ be the maximum revenues achievable in this system. In principle, this optimum could be calculated in the following way. Let $B(t)$ the m-dimensional random variable that describes the remaining budgets of advertisers (with the boundary condition $B(0) = BT$), and define the admissible control set at time t to be:

$$\mathcal{O}^t = \{o : \mathcal{X} \to \{\mathbf{e}_1, \ldots, \mathbf{e}_m\} \text{ s.t. } \mathbb{1}'o \leq 1, o \leq B(t)\},$$

where $\mathbf{e}_a$ is the a-th unit vector in $\mathbb{R}^m$. Then,

$$\mathrm{OPT}(T, TB) = \max_{o^t \in \mathcal{O}^t} \mathsf{E}\left[\sum_{t=1}^{T} \sum_{a} r_a \mathbb{1}(X_t \in \mathcal{X}_a) o_a^t\right].$$

In the analysis of our algorithm, we work with a natural upper bound on this

[2]It is not necessary to constrain $\mathcal{X}$ to be discrete. Alternatively, $\mathcal{X}$ could be any Borel measurable set and d would correspond to the metric entropy of $\mathcal{X}$.

optimum value. Let us define the following unit time optimization problem

$$
\begin{aligned}
\mathrm{LP}_\mu = \max \quad & \sum_a \sum_{x\in\mathcal{X}} r_a \mathbb{1}(x \in \mathcal{X}_a) z(x,a)\mu(x) \\
\text{subject to} \quad & \sum_{x\in\mathcal{X}} z(x,a)\mu(x) \leq B_a \\
& \sum_a z(x,a) \leq 1 \\
& z \geq 0,
\end{aligned}
$$

together with its dual formulation:

$$
\begin{aligned}
\text{D-LP}_\mu = \min \quad & \sum_{x\in\mathcal{X}} \alpha(x) + \sum_a B_a \beta(a) \\
\text{subject to} \quad & \alpha(x) + \mu(x)\beta(a) \geq r_a \mathbb{1}(x \in \mathcal{X}_a)\mu(x) \\
& \alpha, \beta \geq 0.
\end{aligned}
$$

One can interpret LP_μ as the long run unit time revenue a clairvoyant could achieve as $T \to \infty$ with a priori knowledge of μ. More rigorously, LP_μ provides an upper bound on the unit time optimum as stated in the following lemma whose proof is delayed to the Appendix:

Lemma 3.2.1 *$LP_\mu \geq OPT(T, TB)/T$.*

It will be convenient in our analysis to use LP_μ as the benchmark to measure the performance of our learning algorithm.

Furthermore, we make the well-known observation that the dual of LP_μ gives rise to a vector of "bid prices" on advertisers which can be used to calculate a primal control in the following way:

Definition Let $\beta \in \mathbb{R}_+^m$ be a vector of shadow prices for the m advertisers. The bid price control associated to β is a map

$$z^\beta : \mathcal{X} \times [m] \to \{0, 1\}$$

such that

$$z^{\beta}(x,a) = \begin{cases} 1, & \text{if } a \in \arg\max_j \{r_j \mathbb{1}(x \in \mathcal{X}_j) - \beta(j)\} \text{ uniquely} \\ & \text{and } r_a \mathbb{1}(x \in \mathcal{X}_a) - \beta(a) > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Note that our definition potentially throws away many impressions because of ties. We show that the impact of how we deal with ties is negligible in the next section.

We are now ready to describe our learning algorithm. As with previous approaches such as Agrawal et al. (2014), we allow the algorithm a "burn-in" period to observe N training impression arrivals and estimate a control policy. This policy will then be used to decide the allocation for the impressions that arrive over the following T period horizon.

Learning algorithm:

1. Sample N impressions from μ and calculate the empirical distribution $\hat{\mu}_N$

$$\hat{\mu}_N(x) = \frac{1}{N} \sum_{1 \le i \le N} \mathbb{1}(X_i = x).$$

2. Compute an extreme point $\hat{\beta} \in \arg\min \text{D-LP}_{\hat{\mu}_N}$.

3. Use the control $z^{\hat{\beta}}$ to allocate impressions $X_1, \ldots, X_T$.

Computational burden of the learning scheme. The size of D-LP$_{\hat{\mu}_N}$ in step 2 of the algorithm only depends on n artificially. In fact, since at most N points of $\hat{\mu}$ have nonzero density, the impression dimension of D-LP is at most N. Put another way, the benefits of our algorithm having low sample complexity are two-fold - besides learning a control policy with a parsimonious number of samples, the computational complexity of the underlying control problem is also reduced significantly.

We end this section by defining additional notation that we will use throughout the rest of the paper. For some bid price β, let Z_a^{β} be the set of x's which get allocated to advertiser a, i.e.

$$Z_a^\beta = \{x \in \mathcal{X} : z^\beta(x,a) > 0\}.$$

Additionally, let us denote the revenues of using policy β when impressions arrive from measure ν by

$$\begin{aligned}
\mathrm{Rev}_\nu(\beta) &= \sum_a r_a \min\left\{B_a, \sum_{x\in\mathcal{X}} \mathbb{1}(x \in \mathcal{X}_a) z^\beta(x,a)\nu(x)dx\right\} \\
&= \sum_a r_a \min\left\{B_a, \nu\left(Z_a^\beta\right)\right\}.
\end{aligned}$$

Lastly, we define the following quantities which will appear in our sample complexity results:

$$r_{\text{avg}} = \frac{1}{m}\sum_a r_a,\ r_{\max} = \max_a r_a$$

and, correspondingly,

$$\rho_{\text{avg}} = \min\left\{\frac{\mathrm{LP}_\mu}{r_{\text{avg}}}, 1\right\},\ \rho = \rho_{\max} = \frac{\mathrm{LP}_\mu}{r_{\max}}.$$

3.2.1 Bid prices and Optimality

In this section, we describe a generic condition under which using the bid price controls defined above closely approximates the value of using the optimal primal control. Our argument here is similar to that used in Agrawal et al. (2014) and requires that the impression type distribution μ is "granular" enough that no single mis-assigned impression type can contribute a disproportionately large fraction of the optimal revenues. We define this property formally below:

Definition A distribution $\nu : \mathcal{X} \to [0,1]$ is ϵ-good if

$$||\nu||_\infty \leq \frac{\epsilon\rho}{m}.$$

We give some brief intuition for this condition here. While we delay a formal and detailed argument to the Appendix, it turns out that the bid price policy coincides with an optimal primal solution in all but at most m impression types in $\mathcal{X}$ and, by the way we defined z^{β}, does not assign the rest. Hence, the error from the mis-assignment versus the primal optimum is at most $m r_{\max} \max_x \mu(x)$. For an ϵ-good distribution, this is bounded by $\epsilon \mathrm{LP}_{\mu}$.

In order for our analysis to carry through, we need to make the following assumption on the granularity of μ:

Assumption 3.2.2 *The true distribution μ is $\frac{\epsilon}{m}$-good.*

We make two observations: the first is that, while it seems we have imposed an unnecessarily strong condition on μ, we will leverage this to further guarantee that $\hat{\beta}$ is also at most $\epsilon \mathrm{LP}_{\mu}$ away from $\mathrm{LP}_{\hat{\mu}}$, which is a condition our analysis will require. Secondly, this assumption is without loss of generality: if there exist heavy mass impression types that violate this assumption, we can divide them into several artificial types such that each individual point $x \in \mathcal{X}$ has the required condition on its probability mass.

We now state a lemma qualifying the optimality of the bid price $\hat{\beta}$ versus $\mathrm{LP}_{\hat{\mu}}$; the crucial property that achieves this is that the sampled distribution $\hat{\mu}$ is ϵ-good with high probability for $N = \mathrm{poly}(\log n, m)$. This lemma is proved in the Appendix.

Lemma 3.2.3 *Let $X_1, \ldots, X_N$ be N i.i.d. draws from the distribution μ and for all $x \in \mathcal{X}$,*

$$\hat{\mu}_N(x) = \frac{1}{N} \sum_{1 \leq i \leq N} \mathbb{1}(X_i = x).$$

For

$$N \geq \frac{4}{\rho} \frac{m}{\epsilon \log m} \left(\log n + \log \frac{1}{\delta} \right),$$

$\hat{\mu}_N$ is ϵ-good with probability at least $1 - \delta$ as long as μ satisfies Assumption 3.2.2.

Before moving on, we note that, as suggested, there is a loss of a factor of m in the granularity of $\hat{\mu}$ versus μ. We could instead only assume μ is ϵ-good and still get the same guarantee for $\hat{\mu}$ via the Dvoretsky-Kiefer-Wolfovitz inequality, but at a cost of a quadratic dependence on m in the sample complexity.

Lastly, we give the following bound on the total number of possible bid prices which can arise as the solution to the dual linear program over all possible distributions ν. We will later use this lemma in Section 3.4, as proving that $\hat{\beta}$ is approximately optimal with high probability will involve taking a union bound over all possible bid price controls that our algorithm could output.

Lemma 3.2.4 *Let*

$$\mathcal{B} = \{\beta \in \mathbb{R}^m \text{ s.t. } \exists \text{ a distribution } \nu \text{ for which } \beta \text{ is an extreme point of D-LP}_\nu\}.$$

Then

$$|\mathcal{B}| \leq \binom{nm}{m}.$$

Proof For any ν, using the transformation $\tilde{\alpha}(x) = \frac{\alpha(x)}{\nu(x)}$ yields that β is the solution to:

$$\begin{aligned} \min \quad & \sum_{x \in \mathcal{X}} \nu(x)\tilde{\alpha}(x) + \sum_a B_a \beta(a) \\ \text{subject to} \quad & \tilde{\alpha}(x) + \beta(a) \geq r_a \quad \forall a, x \in \mathcal{X}_a \\ & \tilde{\alpha}, \beta \geq 0, \end{aligned}$$

The feasible set of the dual LP under the above change of variable does not depend on ν, so we are only left with counting the total number of βs that can form its extreme points, of which there are at most

$$\binom{\sum_a |\mathcal{X}_a|}{m} \leq \binom{mn}{m}.$$

3.3 A Lower Bound on N

In this section, we exhibit a lower bound on the number of samples our algorithm requires to find a near optimal $\hat{\beta}$. In order to do this, we ask a simpler question: for a *fixed* bid price policy β, how many samples N are needed to estimate the revenues from using that policy (i.e. bound the distance between $\text{Rev}_{\hat{\mu}_N}(\beta)$ and $\text{Rev}_\mu(\beta)$)? In fact, it turns out that there exist instances for which there will always exist a gap between the estimator $\mathrm{Rev}_{\hat{\mu}_N}(\beta)$ and its true value:

Lemma 3.3.1 *Fix any bid price β and consider a family of instances where we set $B_a = \mu\left(Z_a^\beta\right) = \frac{1}{m}, \forall a$. Then,*

$$|\mathsf{E}\left[Rev_{\hat{\mu}_N}(\beta)\right] - Rev_{\mu}(\beta)| \geq \frac{1}{\sqrt{2\pi}} r_{avg,1}\sqrt{\frac{m}{N}} - 3r_{avg,1}\frac{m}{N}.$$

Proof The bound is a consequence of part 2 of Lemma A.10.1 in the Appendix with $Y_N = \hat{\mu}_N\left(Z_a^\beta\right)$:

$$\begin{aligned}
|\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right] - \mathrm{Rev}_{\mu}(\beta)| &= \left|\mathsf{E}\left[\sum_a r_a \min\left\{B_a, \hat{\mu}_N\left(Z_a^\beta\right)\right\}\right] - \sum_a r_a \min\left\{B_a, \mu\left(Z_a^\beta\right)\right\}\right| \\
&= \left|\mathsf{E}\left[\sum_a r_a \left(\hat{\mu}_N\left(Z_a^\beta\right) - \frac{1}{m}\right)^-\right]\right| \\
&= \sum_a r_a \left|\mathsf{E}\left[\left(\hat{\mu}_N\left(Z_a^\beta\right) - \frac{1}{m}\right)^-\right]\right| \\
&\geq \sum_a r_a \left(\frac{1}{\sqrt{2\pi}}\sigma\left(\hat{\mu}_N\left(Z_a^\beta\right)\right) - \frac{3\left(1 - 2\mu_N\left(Z_a^\beta\right)\right)}{N}\right) \\
&\geq \frac{1}{\sqrt{2\pi}} r_{\text{avg}}\sqrt{\frac{m}{N}} - 3r_{\text{avg}}\frac{m}{N}.
\end{aligned}$$

Note that, for the third equality, we have used that all terms within the sum are all negative and, for the last inequality, the fact that for all a, $\sigma\left(\hat{\mu}_N\left(Z_a^\beta\right)\right) = \sqrt{\frac{1}{m}\left(1 - \frac{1}{m}\right)/N}$.

Note that this implies that in order to bound the error of a single β by $\epsilon \mathrm{LP}_\mu$ one needs to draw on the order of $\left(\frac{r_{\text{avg}}}{\mathrm{LP}_\mu}\right)^2 \frac{m}{\epsilon^2}$ samples. Not surprisingly, this turns out to be a bound on the number of samples required by our algorithm to learn a near optimal β over the entire space of bid prices, as evidenced by the following theorem:

Theorem 3.3.2 *The algorithm requires drawing at least*

$$N = \Omega\left(\left(\frac{1}{\rho_{avg}}\right)^2 \frac{m}{\epsilon^2}\right)$$

samples to guarantee that

$$|Rev_\mu(\beta) - \mathsf{E}[Rev_\mu(\hat{\beta})]| \leq \epsilon LP_\mu.$$

Proof We exhibit a simple instance for which a large estimation gap in the value of the optimal bid price policy β^* implies a large optimality gap for the approximate bid price policy $\hat{\beta}$. Let us fix a bid price control β: we construct an instance such that this β is optimal by setting $B_a = \mu(Z_a^\beta) = 1/m$ and additionally, setting the bids such that for every $x \in \mathcal{X}$, there exists a unique advertiser a such that $x \in \mathcal{X}_a$ (in other words, each impression type can only go to one advertiser). For this instance, β achieves the best possible revenues, equal to $r_{\text{avg},1}$. Furthermore $\beta_a = \beta_a^* = 0$, for all a, and $\mathcal{X}_a = Z_a^\beta$. Let us assume that the estimation gap for β is such that:

$$|\text{Rev}_\mu(\beta) - \mathsf{E}\left[\text{Rev}_{\hat{\mu}}(\beta)\right]| \geq \Delta$$

and we shall prove by contradiction that this implies:

$$|\text{Rev}_\mu(\beta) - \mathsf{E}\left[\text{Rev}_\mu(\hat{\beta})\right]| \geq \Delta.$$

Assume the contrary; since $\text{Rev}_\mu(\beta) = \sum r_a B_a = r_{\text{avg},1}$ and thus is greater or equal to both $\mathsf{E}\left[\text{Rev}_{\hat{\mu}}(\beta)\right]$ and $\mathsf{E}\left[\text{Rev}_\mu(\hat{\beta})\right]$, it must then be that

$$\mathsf{E}\left[\text{Rev}_{\hat{\mu}}(\beta)\right] + \Delta \leq \text{Rev}_\mu(\beta) < \mathsf{E}\left[\text{Rev}_\mu(\hat{\beta})\right] + \Delta$$

which implies that $\mathsf{E}\left[\text{Rev}_\mu(\hat{\beta})\right] > \mathsf{E}\left[\text{Rev}_{\hat{\mu}}(\beta)\right]$, or, expanding their expressions

$$\sum_a r_a \mathsf{E}\left[\left(\min\left\{\frac{1}{m}, \mu(Z_a^{\hat{\beta}})\right\} - \min\left\{\frac{1}{m}, \hat{\mu}(Z_a^{\beta})\right\}\right)\right] \geq 0.$$

Since we have constructed the instance such that no impression types can go to two advertisers, it follows that $\hat{\beta}$ can only take two values, namely

$$\hat{\beta}_a = \begin{cases} r_a, & \text{if } \hat{\mu}(\mathcal{X}_a) > B_a = \mu(\mathcal{X}_a) \\ 0, & \text{otherwise} \end{cases}$$

and, since this implies we accept impression type x if and only if $\hat{\mu}(\mathcal{X}_a) > \mu(\mathcal{X}_a)$,

$$\mathsf{E}\left[\min\left\{\frac{1}{m}, \mu(Z_a^{\hat{\beta}})\right\}\right] = \frac{1}{m}\mathbb{P}\left[\hat{\mu}(\mathcal{X}_a) > \mu(\mathcal{X}_a)\right]$$

and

$$\begin{aligned}\mathsf{E}\left[\min\left\{\frac{1}{m}, \hat{\mu}(Z_a^{\beta})\right\}\right] &= \frac{1}{m}\mathbb{P}\left[\hat{\mu}(\mathcal{X}_a) > \mu(\mathcal{X}_a)\right] + \mathsf{E}\left[\hat{\mu}(Z_a^{\beta})\right]\left(1 - \mathbb{P}\left[\hat{\mu}(\mathcal{X}_a) > \mu(\mathcal{X}_a)\right]\right) \\ &= \mathsf{E}\left[\min\left\{\frac{1}{m}, \hat{\mu}(Z_a^{\beta})\right\}\right] + \\ &\quad \underbrace{\mathsf{E}\left[\hat{\mu}(Z_a^{\beta}) | \hat{\mu}(\mathcal{X}_a) > \mu(\mathcal{X}_a)\right]\left(1 - \mathbb{P}\left[\hat{\mu}(\mathcal{X}_a) > \mu(\mathcal{X}_a)\right]\right)}_{>0}.\end{aligned}$$

Hence $\mathsf{E}\left[\left(\min\left\{\frac{1}{m}, \mu(Z_a^{\hat{\beta}})\right\} - \min\left\{\frac{1}{m}, \hat{\mu}(Z_a^{\beta})\right\}\right)\right] < 0$ and summing over all a we get the desired contradiction. To complete the proof, note that to get $\Delta = \Theta\left(r_{\text{avg}}\sqrt{\frac{m}{N}}\right) \leq \epsilon \mathrm{LP}_\mu$, we need to set

$$N = \Omega\left(\left(\frac{1}{\rho_{\text{avg}}}\right)^2 \frac{m}{\epsilon^2}\right).$$

3.4 Sample Complexity

The key step in our sample complexity analysis will be to find a uniform bound on the estimation error of $|\mathrm{Rev}_{\hat{\mu}_N}(\beta) - \mathrm{Rev}_\mu(\beta)|$, over all bid-prices $\beta \in \mathcal{B}$. We state this key lemma below.

Lemma 3.4.1 *For*

$$N = \frac{64}{\rho^2\epsilon^2}\left(m \log(mn) + \log\frac{1}{\delta}\right),$$

$$\mathbb{P}\left[\exists \beta \in \mathcal{B} \ s.t. \ \left|Rev_{\hat{\mu}_N}(\beta) - Rev_{\mu}(\beta)\right| \geq \epsilon LP_{\mu}\right] \leq \delta.$$

In order to prove the above, we proceed in two stages

1. We first bound the estimation error for a fixed $\beta \in \mathcal{B}$. We will break up this error into two components, which we bound in Lemmas 3.4.2 and 3.4.3, respectively.

2. Having bounded the error for a fixed β, we prove the above lemma by taking a union bound over all possible bid-prices, whose cardinality we have upper bounded in Lemma 3.2.4.

As alluded to above, given a particular β, we use the triangle inequality to split the estimation error into two components:

$$|\mathrm{Rev}_{\hat{\mu}_N}(\beta) - \mathrm{Rev}_{\mu}(\beta)| \leq |\mathrm{Rev}_{\hat{\mu}_N}(\beta) - \mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right]| + |\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right] - \mathrm{Rev}_{\mu}(\beta)| \quad (3.1)$$

We bound the two terms in equation 3.1 separately: (a) The first component is probabilistic and we control it using a concentration of measure argument. (b) The second component is precisely the expected bias we lower bounded in Section 3.3; in the following, we provide a uniform matching upper bound on the magnitude of this bias allowing us to calculate the rate at which $\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right]$ approaches $\mathrm{Rev}_{\mu}(\beta)$.

The first term admits the following high probability bound:

Lemma 3.4.2 *For fixed $\beta \in \mathcal{B}$,*

$$\mathbb{P}\left[\left|Rev_{\hat{\mu}_N}(\beta) - \mathsf{E}\left[Rev_{\hat{\mu}_N}(\beta)\right]\right| \geq \epsilon LP_{\mu}\right] \leq 2\exp\left(-\frac{N\epsilon^2\rho^2}{8}\right).$$

Proof Let us view our estimate as a function of the N samples $X = (X_1, \dots, X_N)$ drawn from μ to form the empirical distribution, i.e.

$$\mathrm{Rev}_{\hat{\mu}_N}(\beta) = g(X).$$

We begin by showing that g satisfies a bounded difference property. Consider two particular sequences of observations, $s = (x_1, \dots, x_i, \dots x_N)$ and $s' = (x_1, \dots, x_i', \dots x_N)$

inducing empirical distributions $\hat{\mu}_N$ and, respectively, $\hat{\mu}'_N$. Since there is a single sample on which s and s' differ, it follows that:

$$\hat{\mu}_N(x) = \hat{\mu}'_N(x), \quad \forall x \in \mathcal{X} \setminus \{x_i, x'_i\}$$
$$|\hat{\mu}_N(x) - \hat{\mu}'_N(x)| \leq \frac{1}{N}, \quad \forall x \in \{x_i, x'_i\}$$

and, consequently

$$\begin{aligned}|g(s) - g(s')| &= \left|\sum_a r_a \min\{B_a, \hat{\mu}_N\left(Z_a^\beta\right)\} - \sum_a r_a \min\{B_a, \hat{\mu}'_N\left(Z_a^\beta\right)\}\right| \\ &\leq 2\frac{r_{\max}}{N}.\end{aligned}$$

Using the Bounded Differences Inequality (Proposition A.11.1), it follows that:

$$\mathbb{P}\left[|g(X) - \mathsf{E}\left[g(X)\right]| \geq \epsilon \mathrm{LP}_\mu\right] \leq 2\exp\left(-\frac{N\epsilon^2\rho^2}{8}\right).$$

We now focus on the second term of equation 3.1, $|\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right] - \mathrm{Rev}_\mu(\beta)|$ and prove an upper bound on the magnitude of the error.

Lemma 3.4.3 *For any $\beta \in R^m$ and $N \geq m$,*

$$|\mathsf{E}\left[Rev_{\hat{\mu}_N}(\beta)\right] - Rev_\mu(\beta)| \leq 4r_{\max}\sqrt{\frac{m}{N}}.$$

Proof The expected bias is

$$\begin{aligned}
\left|\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right]-\mathrm{Rev}_{\mu}(\beta)\right| &= \left|\mathsf{E}\left[\sum_a r_a \min\left\{B_a, \hat{\mu}_N\left(Z_a^\beta\right)\right\}\right]-\sum_a r_a \min\left\{B_a, \mu\left(Z_a^\beta\right)\right\}\right| \\
&= \left|\mathsf{E}\left[\sum_a r_a \min\left\{B_a, \hat{\mu}_N\left(Z_a^\beta\right)\right\}\right]-\sum_a r_a \min\left\{B_a, \mathsf{E}\left[\hat{\mu}_N\left(Z_a^\beta\right)\right]\right\}\right| \\
&\leq \sum_a r_a \left|\mathsf{E}\left[\min\left\{B_a, \hat{\mu}_N\left(Z_a^\beta\right)\right\}-\min\left\{B_a, \mathsf{E}\left[\hat{\mu}_N\left(Z_a^\beta\right)\right]\right\}\right]\right| \\
&\leq \frac{1}{\sqrt{2\pi}}\sum_a r_a\left(\sigma\left(\hat{\mu}_N\left(Z_a^\beta\right)\right)+\frac{3\left(1-2\mu\left(Z_a^\beta\right)\right)}{N}\right) \\
&\leq \frac{1}{\sqrt{2\pi}}\sum_a r_a \sigma\left(\hat{\mu}_N\left(Z_a^\beta\right)\right)+3\frac{m}{N}r_{\max}
\end{aligned}$$

where the first equality follows from linearity of expectations, the second from the fact that $\mathsf{E}\left[\hat{\mu}_N\left(Z_a^\beta\right)\right]=\mu\left(Z_a^\beta\right)$, the first inequality is an application of the triangle inequality, the second inequality follows from Lemma A.10.1 with $X_N=\hat{\mu}_N\left(Z_a^\beta\right)$ and in the last inequality we assumed $N \geq m$.

For ease of notation, let us call $p_a = \mathbb{P}\left[\mathbb{1}(x \in Z_a^\beta)\right]$ such that

$$\sigma\left(\hat{\mu}_N\left(Z_a^\beta\right)\right)=\sqrt{\frac{p_a(1-p_a)}{N}}.$$

In order to find a uniform bound on the expected bias $\left|\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right]-\mathrm{Rev}_{\mu}(\beta)\right|$ (up to constants), we can now simply optimize the above the bound over all possible probabilities p:

$$\begin{aligned}
\left|\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right]-\mathrm{Rev}_{\mu}(\beta)\right| &\leq \frac{1}{\sqrt{2\pi}}\max_{p\geq 0, 1^T p \leq 1}\sum_a r_a\sqrt{\frac{p_a(1-p_a)}{N}}+3\frac{m}{N}r_{\max} \\
&\leq \frac{1}{\sqrt{2\pi}}r_{\max}\max_{p\geq 0, 1^T p \leq 1}\sum_a \sqrt{\frac{p_a(1-p_a)}{N}}+3\frac{m}{N}r_{\max}.
\end{aligned}$$

The last optimization problem is maximized when $p_a = \frac{1}{m}, \forall a$, yielding the bound

$$\begin{aligned} \left|\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right] - \mathrm{Rev}_{\mu}(\beta)\right| &\leq \frac{1}{\sqrt{2\pi}} r_{\max}\sqrt{\frac{m}{N}} + 3r_{\max}\frac{m}{N} \\ &\leq 4r_{\max}\sqrt{\frac{m}{N}}. \end{aligned}$$

We note that the lemma above heavily uses the special structure of our problem by simplifying the bias to a sum of truncated random variables; such a simplification would not be possible had we used a more general price structure where advertisers might bid different amounts over the space $\mathcal{X}_a$ of compatible impression types.

Proof of Lemma 3.4.1. We show that, if $N \geq \frac{64}{\rho^2\epsilon^2}\left(m\log(mn) + \log\frac{1}{\delta}\right)$,

$$\mathbb{P}\left[\exists\beta \in \mathcal{B} \text{ s.t. } \left|\mathrm{Rev}_{\hat{\mu}_N}(\beta) - \mathrm{Rev}_{\mu}(\beta)\right| \geq \epsilon \mathrm{LP}_{\mu}\right] \leq \delta.$$

For any $N = N_1 \geq \frac{64m}{\rho^2\epsilon^2}$, Lemma 3.4.3 guarantees that

$$\left|\mathsf{E}[\mathrm{Rev}_{\hat{\mu}_N}(\beta)] - \mathrm{Rev}_{\mu}(\beta)\right| \leq \frac{\epsilon}{2}\mathrm{LP}_{\mu},$$

such that

$$\mathbb{P}\left[\exists\beta \in \mathcal{B} \text{ s.t. } \left|\mathrm{Rev}_{\hat{\mu}_N}(\beta) - \mathrm{Rev}_{\mu}(\beta)\right| \geq \epsilon \mathrm{LP}_{\mu}\right] \leq$$

$$\mathbb{P}\left[\exists\beta \in \mathcal{B} \text{ s.t. } \left|\mathrm{Rev}_{\hat{\mu}_N}(\beta) - \mathsf{E}[\mathrm{Rev}_{\hat{\mu}_N}(\beta)]\right| \geq \frac{\epsilon}{2}\mathrm{LP}_{\mu}\right]$$

as a consequence of equation 3.1. To conclude our proof, we simply employ a union bound over $\beta \in \mathcal{B}$ (Lemma 3.2.4) and use Lemma 3.4.2 to show that

$$\mathbb{P}\left[\exists\beta \in \mathcal{B} \text{ s.t. } \left|\mathrm{Rev}_{\hat{\mu}_N}(\beta) - \mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right]\right| \geq \frac{\epsilon}{2}\mathrm{LP}_{\mu}\right]$$

$$\leq \binom{mn}{m} \mathbb{P}\left[|\mathrm{Rev}_{\hat{\mu}_N}(\beta) - \mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right]| \geq \frac{\epsilon}{2}\mathrm{LP}_\mu\right]$$
$$\leq \binom{mn}{m} 2\exp\left(-\frac{1}{32}\epsilon^2 N\rho^2\right)$$
$$\leq (mn)^m 2\exp\left(-\frac{1}{32}\epsilon^2 N\rho^2\right),$$

The above probability is bounded by δ for $N_2 = \frac{64}{\rho^2\epsilon^2}\left(m\log(mn) + \log\frac{1}{\delta}\right)$. Therefore, taking $N \geq \max\{N_1, N_2\} = N_2$ yields the result.

The following theorem uses the above uniform bound on the estimation error over all bid-prices to show that the sampled problem, in which impressions arrive from $\hat{\mu}_N$, provides a close representation of the original problem.

Theorem 3.4.4 *Let $\beta^* \in \operatorname{argmax} D\text{-}LP_\mu$ and $\hat{\beta} \in \operatorname{argmax} D\text{-}LP_{\hat{\mu}_N}$. With probability at least $1 - 2\delta$,*

$$\left|Rev_\mu(\beta^*) - Rev_{\hat{\mu}_N}\left(\hat{\beta}\right)\right| \leq \epsilon LP_\mu,$$

for

$$N = \frac{256}{\rho^2\epsilon^2}\left(m\log(mn) + \log\frac{1}{\delta}\right).$$

Proof First we take a union bound over the events $\{\hat{\mu}$ is not $\epsilon/2$-good with respect to $\mathrm{LP}_\mu\}$ and $\left\{\exists\beta, |\mathrm{Rev}_\mu(\beta) - \mathrm{Rev}_{\hat{\mu}_N}(\beta)| \leq \frac{\epsilon}{2}\mathrm{LP}_\mu\right\}$ to show that for

$$N = \frac{256}{\rho^2\epsilon^2}\left(m\log(mn) + \log\frac{1}{\delta}\right)$$

the following hold with probability at least $1 - 2\delta$,

$$|\mathrm{LP}_{\hat{\mu}_N} - \mathrm{Rev}_{\hat{\mu}_N}(\beta^*)| \leq \frac{\epsilon}{2}\mathrm{LP}_\mu, \tag{3.2}$$

$$|\mathrm{Rev}_\mu(\beta^*) - \mathrm{Rev}_{\hat{\mu}_N}(\beta^*)| \leq \frac{\epsilon}{2}\mathrm{LP}_\mu, \tag{3.3}$$

and

$$\left|\mathrm{Rev}_\mu(\hat{\beta}) - \mathrm{Rev}_{\hat{\mu}_N}(\hat{\beta})\right| \leq \frac{\epsilon}{2}\mathrm{LP}_\mu \tag{3.4}$$

We have used Lemma 3.2.3 for 3.2 and Lemma 3.4.1 for 3.3 and 3.4.

But then,

$$\begin{aligned}
\mathrm{Rev}_\mu(\beta^*) &\geq \mathrm{Rev}_\mu(\hat{\beta}) - \frac{\epsilon}{4}\mathrm{LP}_\mu \\
&\geq \mathrm{Rev}_{\hat{\mu}_N}(\hat{\beta}) - \frac{3\epsilon}{4}\mathrm{LP}_\mu \\
&\geq \mathrm{Rev}_{\hat{\mu}_N}(\beta^*) - \frac{5\epsilon}{4}\mathrm{LP}_\mu \\
&\geq \mathrm{Rev}_\mu(\beta^*) - \frac{7\epsilon}{4}\mathrm{LP}_\mu,
\end{aligned}$$

where the first inequality follows from applying Assumption 3.2.2 and Lemma A.9.1, the second from equation 3.3, the third from equation 3.2, and the fourth from equation 3.4. It hence follows that

$$\mathrm{Rev}_\mu(\beta^*) + \epsilon\mathrm{LP}_\mu \geq \mathrm{Rev}_{\hat{\mu}_N}(\hat{\beta}) \geq \mathrm{Rev}_\mu(\beta^*) - \epsilon\mathrm{LP}_\mu,$$

or, equivalently,

$$\left|\mathrm{Rev}_\mu\left(\beta^*\right) - \mathrm{Rev}_{\hat{\mu}_N}\left(\hat{\beta}\right)\right| \leq \epsilon\mathrm{LP}_\mu.$$

We are now ready to prove our main result, which is a direct consequence of the theorem above, and proves that the sampled bid-price control $\hat{\beta}$ gives a $1-\epsilon$ approximation to the optimal primal control:

Theorem 3.4.5 *With probability at least* $1-2\delta$*,*

$$Rev_\mu(\hat{\beta}_N) \geq (1-2\epsilon)LP_\mu$$

as long as

$$N \geq \frac{256}{\rho^2\epsilon^2}\left(m\log(mn) + \log\frac{1}{\delta}\right).$$

Proof Using the same union bound over events as for theorem 3.4.4,

$$\begin{aligned}
\mathrm{LP}_\mu - \mathrm{Rev}_\mu(\hat{\beta}) &\leq \mathrm{Rev}_\mu(\beta^*) - \mathrm{Rev}_\mu(\hat{\beta}) + \frac{\epsilon}{4}\mathrm{LP}_\mu \\
&\leq \left|\mathrm{Rev}_\mu(\beta^*) - \mathrm{Rev}_{\hat{\mu}_N}(\hat{\beta})\right| + \left|\mathrm{Rev}_{\hat{\mu}_N}(\hat{\beta}) - \mathrm{Rev}_\mu(\hat{\beta})\right| + \frac{\epsilon}{2}\mathrm{LP}_\mu \\
&\leq 2\epsilon\mathrm{LP}_\mu,
\end{aligned}$$

where we have used Lemma A.9.1 in the first inequality, the triangle inequality in the second, and Theorem 3.4.4 and equation 3.4 in the third inequality.

Finally, we relate our main theorem back to OPT(T,TB). Theorem 3.4.5 is a statement regarding the performance of our algorithm as $T \to \infty$. However, it is straightforward to establish that this result holds for finite T:

Corollary 3.4.6 *Let N satisfy the condition in Theorem 3.4.5. Let $Rev^T(\hat{\beta}_N)$ be the revenues garnered from using the resulting bid-price control on the following T samples. Then, with probability $1-3\delta$,*

$$Rev^T(\hat{\beta}_N) \geq (1-3\epsilon)OPT(T, B^T),$$

for $T \geq \frac{64}{\rho^2\epsilon^2}\left(m\log(mn) + \log\frac{1}{\delta}\right)$.

Proof Note that $\mathrm{Rev}^T(\hat{\beta}_N) = T\mathrm{Rev}_{\hat{\mu}_T}(\hat{\beta}_N)$. By applying a union bound and using Lemma 3.4.1 and Theorem 3.4.5, we can guarantee that, with probability at least $1-3\delta$,

$$|\mathrm{Rev}_{\hat{\mu}_T}(\hat{\beta}_N - \mathrm{Rev}_{\mu}(\hat{\beta}_N)| \leq \epsilon \mathrm{LP}_{\mu}$$

and

$$\mathrm{Rev}_{\mu}(\hat{\beta}_N \geq (1-2\epsilon\mathrm{LP}_{\mu}.$$

Hence,

$$\begin{aligned}
\mathrm{Rev}^T(\hat{\beta}_N) \geq & T(\mathrm{Rev}_{\mu}(\hat{\beta}_N) - \epsilon\mathrm{LP}_{\mu}) \\
\geq & T(1-2\epsilon)\mathrm{LP}_{\mu}) \\
\geq & (1-2\epsilon)\mathrm{OPT}(T, TB),
\end{aligned}$$

where the last inequality follows from Lemma 3.2.1.

Before moving on, we address an issue that our analysis raises. Our sample complexity result stems from Lemma 3.4.3, which establishes that

$$|\mathsf{E}\left[\mathrm{Rev}_{\hat{\mu}_N}(\beta)\right] - \mathrm{Rev}_{\mu}(\beta)| \leq 4r_{\mathrm{avg},2}\sqrt{\frac{m}{N}}.$$

Noticing that $g(\hat{\mu}) = \text{Rev}_{\hat{\mu}}(\beta)$ is a concave function in $\hat{\mu}$, the above error term becomes a Jensen's inequality type bias equal to $g(\mathsf{E}[\hat{\mu}]) - \mathsf{E}[g(\mu)] \geq 0$. The simple structure of g in our case potentially leaves room for us to improve our analysis: the idea would be to correct for this convexity bias by using some other estimator that empirical distribution. While beyond the scope of this chapter, this is an interesting avenue for future research that further leverages the unique structure of the problem examined here.

3.5 How large is ρ?

Our sample complexity results depend on the ratio between LP_μ and the maximum advertiser bid. Clearly, an adversary could choose a problem instance where these ratios were arbitrarily small by simply choosing appropriately small advertiser budgets – for example, choosing B such that $\sum_a B_a = 1/n$ would lead to an large N that scaled one-to-one with the size of $\mathcal{X}$. However, such extreme instances would be unlikely to occur in practice, where typically ad networks negotiate budgets with advertisers such that the number of total impressions that must be delivered (i.e. $T \cdot \sum_a B_a$) is roughly balanced with the inventory that is expected to arrive over the life of the campaigns $(T \cdot \sum_x \mu(x))$.

Hence, one might hope that by imposing certain mild constraints on the instance family, such as $\sum_a B_a = K$ for some constant $K > 0$, one might obtain significantly better scaling of the ρs and, consequently, of N. In this section, we present one such generative family of instances for which the ρ ratios are $\tilde{O}(1)$ on average (where the $\tilde{O}()$ notation hides any logarithmic dependencies on n and m). We define the parameters of our generative model for the instance family as follows:

Generative Model 1 *1. Dimensions: Let n be the set of impression types and m be the number of advertisers and assume that*

$$\frac{m \log m}{n} = o(1).$$

2. *Advertiser budgets: Let the advertiser budget vector B be drawn from a joint distribution f with the following properties:*

   (a) $\mathsf{E}[B_a] = \frac{1}{m}$.

   (b) $\mathbb{P}\left[B_a \geq \frac{1}{m}\right] \geq \alpha$ *for some constant* $\alpha > 0$.

3. *μ distribution: Let the impression type distribution μ be drawn iid from a joint distributions f with the following properties:*

   (a) $\mathsf{E}[B_a] = \frac{1}{m}, \mathsf{E}[\mu(x)] = \frac{1}{n}$.

   (b) *For any* $S \subseteq \mathcal{X}$, $\mathbb{P}\left[\sum_{x \in S} \mu(x) \geq \frac{|S|}{n}\right] \geq \beta$ *for some constant* $\beta > 0$.

4. *Graph topology: For every $x \in \mathcal{X}$, we sample uniformly at random one advertiser $a(x) \in [m]$. We allow any edge set of compatible advertisers that contains $a(x)$.*

5. *Advertiser bids: We allow any bid vector $r \in \mathbb{R}_+^m$.*

Before quantifying the magnitude of the value of LP_μ drawn from this family, we pause to highlight the generality of this generative model for Ad Display instances:

- Our choice for the distributions of B and μ is quite general: properties $(2.a)$ and $(3.a)$ enforce that the "load factor" of the instance is constant, while properties $(2.b)$ and $(3.b)$ enforce that the marginals have constant positive mass to the right of their expectations. We note that these properties allow for distributions with both light tailed and heavy tailed marginals.

- One should interpret our requirement on the instance's edge topology in the following way: we allow any graph where the edge set of each impression type contains at least one advertiser chosen uniformly at random. This in particular places no additional constraints on the size of the edge set or the distribution of the other advertisers in this edge set.

For instances belonging to Generative Model 1, we can show that the expectation of LP_μ (and hence ρ) does not depend on either m or $\mathcal{X}$. In particular,

Theorem 3.5.1 *Consider an instance generated according to Generative Model 1. Then,*

$$\mathsf{E}\left[LP_{\mu}\right] = O\left(r_{avg}\right).$$

3.5.1 Comparison with existing results

For the sake of precision, we compare our result to the one-time learning guarantee from Agrawal et al. (2014) – however, our comparison is also valid with respect to the other papers mentioned in the literature review. We can modify their analysis to allow for a separate fixed approximation ratio ϵ and sampling ratio δ to yield the following result:

Theorem 3.5.2 (Agrawal et al. (2014)) *In the random permutation model, as long as*

$$B_{\min} = \min_{a} B_a = \Omega\left(\frac{m\log(n/\epsilon)}{n\epsilon^2\delta}\right),$$

the competitive ratio of the one time bid learning algorithm with δn samples is $1-O(\epsilon)$.

The number of samples required by Agrawal et al. (2014) becomes:

$$N = \Omega\left(\frac{m\log(n/\epsilon)}{\epsilon^2 B_{\min}}\right).$$

The above condition involving $B_{\min}$ is significantly more punishing on typical instances than our condition involving ρ. For example, consider an instance drawn from our generative model 1, with the budgets drawn from the following distribution:

$$B_a = \frac{Y_a}{\sum_{a'} Y_{a'}}$$

where the $Y_a \sim \text{Exp}(n/m)$ are independent. Then, if $U = \sum_{a'} Y_{a'}$,

$$\begin{aligned}
\mathsf{E}\left[B_{\min}\right] &\leq \mathsf{E}\left[\min_a \frac{Y_a}{U}|U > n - 0.5n\right] \mathbb{P}\left[U \geq n - 0.5n\right] + \mathbb{P}\left[U < n - 0.5n\right] \\
&= \mathsf{E}\left[\min_a \frac{Y_a}{U}|U > n - 0.5n\right] \left(1 - \left(\frac{e^{0.5}}{1.5}\right)^{-m}\right) + \left(\frac{e^{0.5}}{1.5}\right)^{-m} \\
&\leq \mathsf{E}\left[2 \min_a \frac{Y_a}{n}\right] \left(1 - \left(\frac{e^{0.5}}{1.5}\right)^{-m}\right) + \left(\frac{e^{0.5}}{1.5}\right)^{-m} \\
&= \frac{2}{m^2} \left(1 - \left(\frac{e^{0.5}}{1.5}\right)^{-m}\right) + \left(\frac{e^{0.5}}{1.5}\right)^{-m} \\
&= \Theta\left(\frac{1}{m^2}\right),
\end{aligned}$$

where in the first equality we used a Chernoff bound for exponential random variables, and in the second equality we have used the fact that the expectation of the minimum of k independent $\text{Exp}(\lambda)$ random variables is $1/k\lambda$. Assuming all advertiser bids are bounded by a constant, Theorem 3.4.5 gives an average sample complexity that scales like

$$O(1)\frac{m \log(nm) + \log 1/\delta}{\epsilon^2},$$

whereas by comparison, the Agrawal et al. (2014) analysis gives a bound of

$$\frac{m^3 \log n/\epsilon}{\epsilon^2}.$$

More generally, distributions with heavier left tails than those of the exponential could potentially lead to even exponentially small minimum budgets in m, leading to an exponential sample complexity, while our sample complexity would stay the same.

3.6 Experimental Performance

We test the sampling algorithm on a family of synthetic Ad Display instances where we set the number of advertisers $m = 50$ and the number of impressions $T = 5000$.

We define the load factor of an instance to be the ratio

$$\text{LF} = \frac{T}{\sum_a B_a}$$

and try out load factors in LF $\in \{0.5, 0.75, 1, 1.5, 2\}$ by setting advertiser budgets uniformly. For every impression in $[T]$, we uniformly draw a random subset of 100 advertisers - this defines the characteristic sets $\mathcal{X}_a$ for each advertiser and hence the impression types. For each advertiser, we also sample a price r_a for $\mathcal{X}_a$ independently from an Exp(1) distribution.

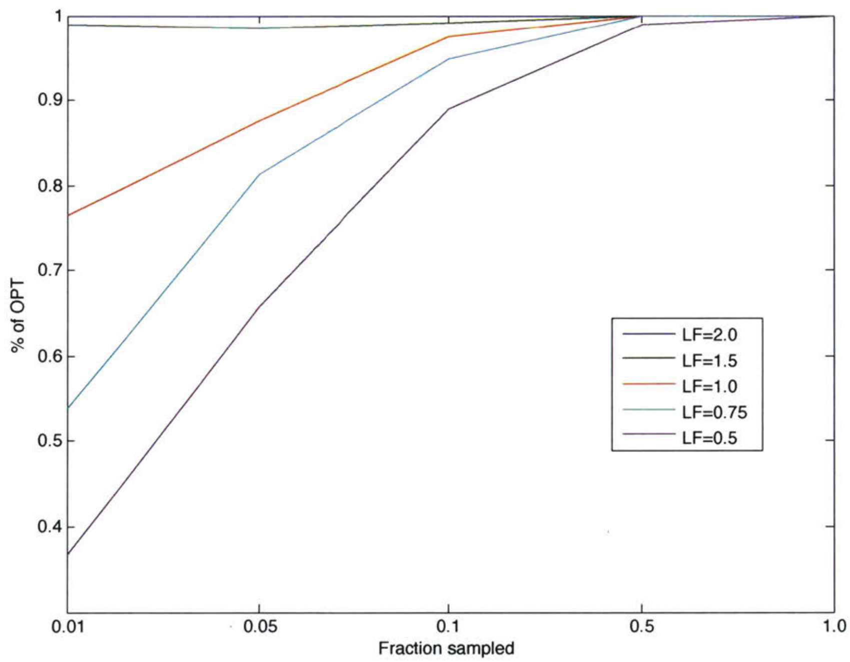
We consider an ensemble of experiments where for 50 randomly drawn instances from the family above, we sample a $100\%, 50\%, 10\%, 5\%$ and 1% fraction of the impressions and we use these sampled impressions to calculate bid prices. We then measure the revenues resulting from these bid prices if they were used on the entire set of T impressions.

Figure 3.6 shows the performance of the sampling algorithms versus the true optimum.We observe that the performance of the algorithm varies substantially based on the load factor of the instance; one might intuit this since t the load factor controls how difficult the instance is. For example, for high load factors one expects the optimal policy to be close to the greedy policy that just allocates all impressions the the highest paying advertiser. Such a policy would be easy to learn, which can be seen in our experiments where for LF $= 1.5, 2$ very few samples essentially yield the optimum. On the other hand, for lower load factors where it is not clear the optimum policy is as simple, the amount of samples has a substantive difference on performance.

3.7 Conclusions

We have analyzed a class of NRM models, specifically Ad-Display allocation problems, in which the sample complexity of learning a high-dimensional demand object scales linearly with its underlying dimension, whereas previous results suggested the best

Figure 3-1: Performance of our algorithm as a function of the fraction of the impressions which are sampled, for a variety of problem instances with different load factors.



dependence was exponential. Moreover, we have established a lower bound on the sample complexity of our estimator.

There are several direction of future research that we find particularly tempting:

1. In this captor we have developed an algorithm which is based on the empirical estimator. A question to ask is whether this estimator is the one that leads to the optimal sample complexity, or whether alternative estimators could work better.

2. One way to interpret our result is that we have made a "low rank" assumption on the problem structure which has resulted in a revenue function that is easier to learn than it would be in more general models. An interesting direction is to ask whether other low rank assumptions can yield similar results for a broader class of allocation problems.

3. A natural extension to our model is to think of advertisers as also arriving i.i.d. from some distribution of features. It would be very interesting to see whether a bid price policy can be built on sampling both advertisers and impressions, and whether such an approach could lead to a better dependence on m in the sample complexity.

Part III

Massive Scale Optimization

Chapter 4

Solving Linear Programs in Map-Reduce

4.1 Introduction.

The "yield management" problem is a central optimization problem that must be solved by ad networks in the process of optimally matching impressions (sessions on web sites/ mobile apps, etc.) to advertisements. The problem is non-trivial since, in addition to many other business constraints, advertisers have finite budgets, the supply of impressions of a given type is limited, and finally, the economic value generated from a single impression can vary widely across ads and advertisers. The net economic value of this problem is large (on the order of tens of billions of dollars in a year).

A first best approach to solving this problem in practice entails solving a certain linear program – the so-called "DLP" – and today forms the basis to solving yield management problems in several large industries (airlines, hospitality, etc.). This approach has not seen wide adoption in advertising applications which instead rely on certain "adaptive greedy" approaches to allocation in spite of a potentially large up-side to using the former approach. One of the primary reasons motivating this choice is efficient computation at scale. In particular, we do not have effective tools for computing the DLP at web-scale and in the sorts of distributed computational environments that are germane to those settings.

This paper presents a candidate algorithm that is amenable to solving a large class of structured linear programs that include the DLP in Map-Reducible environments, and at web-scale. In particular, we make the following contributions:

1. We develop an algorithm for solving structured LPs (such as the DLP) by solving a sequence of "projected" versions of the DLP, a general scheme introduced in seminal work by Plotkins, Shmoys and Tardos two decades ago (Plotkin et al. (1991)). Here, we choose the projection carefully so that solving the projection relies on a fully combinatorial algorithm for the computation of a 2-D convex hull, which in turn relies on a sort. Put another way, the *key large scale computational step in our scheme is a large scale sort*; sorting is an operation that many modern distributed computational frameworks, such as Map-Reduce handle well (see for instance O'Malley (May 2008)).

2. We prove that the number of projections solved by our approach for an $\epsilon-$optimal solution scales like $O\left(\frac{\rho \log A}{\epsilon^2}\right)$, where A is the number of advertisers in the problem and ρ is a certain sparsity parameter. Importantly, the number of rounds (or sorts) is independent of the number of impression types which can be very large – in the case of a naive application of PST, the dependence would be $O(\log(I + A))$. Our algorithm is hence particularly appealing for lop-sided bipartite matching problems where $I \gg A$ (perhaps even exponentially so).

3. Most importantly, we implement our scheme in a large scale shared memory environment where comparisons with commercial, *non-distributed* solvers are possible alongside comparisons with state of the art distributed approaches tailored to solving packing programs. Here we show that we outperform distributed approaches for packing programs by up to an order of magnitude on typical instances. Surprisingly, we also outperform a state-of-the-art commercial solver that was optimized for network structured programs and made full use of all cores for linear algebraic operations; this commercial solver obviously cannot scale beyond a shared memory environment.

4.1.1 Literature Review

Our work lies at the intersection of several streams of research dealing with (a) dynamic resource allocation, (b) distributed algorithms for linear optimization .

Dynamic Resource Allocation: While the model of dynamic resource allocation we consider covers most models relevant to modern yield management, we single out two specific problems relevant to online advertising: *Display Ads Allocation (DA)* and *AdWords (AW)* both of which are concerned with the allocation of impressions to advertisers. Ciocan and Farias (2012) presents a model predictive control approach to solve such problems that requires repeatedly solving a certain LP and presents constant factor relative performance guarantees assuming that impression arrivals are governed by a general class of stochastic processes. A related class of stochastic models assumes that the stream of impressions form an exchangeable sequence. In this setting, Devanur and Hayes (2009); Feldman et al. (2010); Vee et al. (2010); Agrawal et al. (2014); Molinaro and Ravi (2012), all develop near optimal (i.e. $(1-\epsilon)$-approximation) algorithms that rely on "learning" demand from a few early samples, solving a 'sampled' linear program and then deriving an allocation mechanism from the solution of this program. The class of linear programs we consider in this paper subsumes the class of programs considered in this stream of literature. It is worth noting that when the budget or capacity of advertisers is "large", the adversarial online AW and DA problems admit a $(1-\frac{1}{e})$-competitive algorithm by applying a primal-dual technique(Mehta et al. (2005); Feldman et al. (2009)). This primal-dual analysis again rests on analyzing the properties of a linear program of a similar nature as in the literature above and yields the sort of adaptive greedy algorithms that find use today. However, this approach eschews the use of impression traffic statistics taking a "worst-case" approach instead.

Distributable First Order Methods for Linear Programming. While designing a parallel algorithm for solving general linear programs is a P-complete problem, various primal-dual type techniques have been developed for packing and covering LPs – broadly, these techniques are message passing algorithms which rely on passing update messages between the primal and dual variables of the optimization problem. The multiplicative weights framework of Plotkin et al. (1991) is closest to the current work – our algorithm can be interpreted as a multiplicative update, customized to our special "resource allocation" LP structure. Other algorithms that share commonality with multiplicative weights have been studied in Garg and Könemann (2007); Luby and Nisan (1993). Stateless distributed algorithms have been developed for these problems Awerbuch and Khandekar (2008) and have been recently generalized for solving mixed packing-covering LPs Manshadi et al. (2013).

Besides multiplicative update rules, one other message passing paradigm that is of particular interest are max product belief propagation (BP) algorithms. Originally a tool for inference in graphical models, these algorithms have recently been reinterpreted in the context of linear optimization. For example, for maximum weight bipartite matching and subsequently b-matching, which are a subfamily of the network revenue management LP structures we will consider, Bayati et al. (2008a,b); Sanghavi et al. (2009) establish BP's pseudo-polynomial convergence as long as the matching has a unique optimum. Gamarnik et al. (2012) prove a similar result for min-cost network flow, as well as provide a fully polynomial time random approximation scheme which does not require uniqueness.

There has been relatively less focus devoted to understudying these algorithms from a distributed systems perspective – to the best of our knowledge, Manshadi et al. (2013) is the only existing attempt at building a MapReducible LP algorithm. From a theoretical perspective, our work differs from the results described above in that we seek a solution that is optimized for resource allocation LPs, which we define shortly, rather than a generic algorithm for packing/covering.

4.2 Linear Programs for Yield Management.

The linear program we study is associated with the resource allocation structure from Section 3.2. For completeness, we describe this family of linear programs here. Consider a bi-partite graph with I sources indexed by i and J sinks indexed by j.

The edge set of this graph has size E, and a generic edge is denoted by e. We use the notation $i(e)$ for the source node for edge e and $j(e)$ for its sink node. Given this graph, define the following primitives:

Demand: We associate each source i with a deterministic inflow equal to D_i.

Resources and Resource consumption: We are given a set of A distinct resources indexed by a. The available capacity of these resources is given by a vector $B \in \mathbb{R}_+^A$. We assume that allocating a unit of impression (demand) type i to advertiser (product) j (i.e. making an allocation along edge $e \triangleq (i,j)$) consumes $c_{a,e}$ units of resource a. We write $a \in j'$ if $c_{a,e} > 0$ for some edge e incident on j', (i.e. $j(e) = j'$).

Prices/ Revenues: Allocating a unit of demand along edge e generates revenue p_e. Denote by $p \in \mathbb{R}_+^E$ the column vector whose eth component is p_e.

The objective is to maximize total revenues from fractionally allocating demand from sources to sinks:

$$\begin{aligned} &\max_{z \geq 0} \sum_e p_e z_e \\ &\text{s.t.} \ \sum_e c_{a,e} z_e \leq B_a \quad \forall\, a \\ &\qquad \sum_{e:i(e)=i} z_e \leq D_i \quad \forall\, i. \end{aligned} \tag{4.1}$$

We refer to the first set of constraints (indexed by a) as the *resource constraints*, and the second set (indexed by i) as the *source constraints*. We also note that, while our algorithm solves the primal allocation problem, it is quite easy (via complementary slackness) to transform an optimal primal solution to this LP into a dual solution and obtain bid prices similarly to Talluri and Ryzin (1998).

In practice, D_i is obviously not known and must be learned; the recipe that we have proposed in Chapter 1 to handle the online allocation problem is the following: estimate D_i by observing the demand process for a short time; use the LP with the estimated D_i. Re-estimate D_i at reasonable intervals of time and re-solve the LP. Without loss of generality, we will henceforth normalize all D_i to 1.

The family of general resource allocation problems described above is applicable to many settings. However, we focus on modeling several large scale ad allocation problems including Ad Display (Section 1.2.2) and Generalized Second Price AdWords (Section 2.1.2).

Practical Scale For LP: In online ad related applications, the size of the bipartite graph is up to 1 billion impressions and 1 million advertisers, with 100 billion edges between them – in terms of data sizes, representing such a graph sparsely requires on the order of 100Gb. Lastly, we note that in both applications, it is quite natural to think of the number of an impression as scaling exponentially in some feature dimension d; for example, a given impression may be specified by hundreds of features describing the user's demographic information, browsing history and other parameters.

4.3 Map-Reducing The Yield Management LP.

The yield management linear program (4.1) consists of A resource specific constraints and I source type constraints. Our overall approach solves a sequence of relaxations to this problem by 'averaging' the resource constraints. The solution to each such relaxation will be computed using a (distributed) algorithm that exploits strengths of the Map-Reduce paradigm with the key step that uses 'all' the data being a single (large) sort. In greater detail, the following is a schematic view of the algorithm:

1. Begin with the uniform measure w^0 on all resources.

2. At the t-th stage construct a linear program, form Relax(w^t) that relaxes (4.1) by replacing the A resource constraints

$$\sum_e c_{a,e} z_e \leq B_a \quad \forall\ a$$

with a single 'averaged' constraint

$$\sum_a w_a^t \sum_e c_{a,e} z_e \leq \sum_a w_a^t B_a$$

using the measure w^t.

3. Solve Relax(w^t) using a fully combinatorial distributed algorithm that exploits map-reduce; described in detail in Section 4.3.1.

4. Use the solution of the relaxation to compute w^{t+1} (discussed in Section 4.3.2); go to step 2.

In the following two sections, we answer (i) how one can solve Relax(w^t) and (ii) how many relaxations are necessary to guarantee convergence to an optimal solution of (4.1).

4.3.1 Solving Relax(w).

Our goal here is to solve the relaxed program Relax(w) that forms the crux of each iteration of our scheme. We first write this relaxation as

$$\begin{aligned} \max_{z\geq 0} & \sum_e p_e z_e \\ \text{s.t.} & \sum_e c_e z_e \leq B \\ & \sum_{e:i(e)=i} z_e \leq 1 \quad \forall\, i, \end{aligned} \tag{4.2}$$

where $c_e \triangleq \sum_a w_a c_{a,e}$ and $B \triangleq \sum_a w_a B_a$. Let us begin by observing that we can partition all the decision variables z_e of (4.2) by source node (i.e. impression type) i, suggesting I source type specific subproblems whose primal and, respectively, dual are:

$$\begin{aligned} f^i(B^i) = \max_{z\geq 0} \quad & \sum_{e:i(e)=i} p_e z_e \\ \text{s.t.} \quad & \sum_{e:i(e)=i} c_e z_e \leq B^i \\ & \sum_{e:i(e)=i} z_e \leq 1 \end{aligned} \qquad\qquad \begin{aligned} \min_{u,v\geq 0} \quad & uB^i + v \\ \text{s.t.} \quad & uc_e + v \geq p_e \quad \forall e \text{ s.t. } i(e) = i \end{aligned}$$

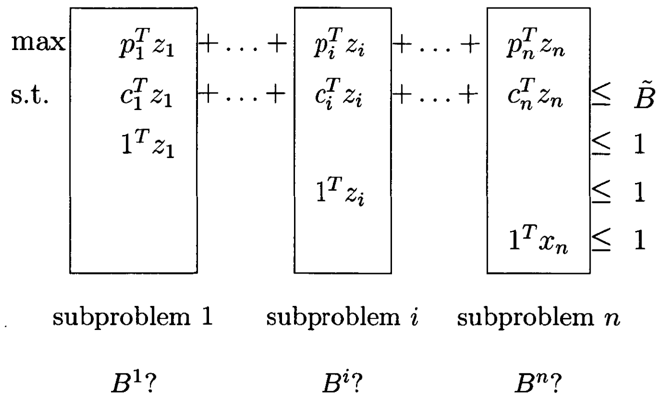


Figure 4-1: Subproblem structure for an Ad-Display type allocation LP.

Figure 4-1 illustrates the nature of this decomposition pictorially. The objective value Relax(w) is then equivalent to that of the following program:

$$\begin{aligned} \max_{B \geq 0} \quad & \sum_i f^i(B^i) \\ \text{s.t.} \quad & \sum_i B^i \leq \tilde{B}, \end{aligned} \tag{4.3}$$

This equivalent program is now easily seen to be a non-linear knapsack problem that we solve as follows:

1. Compute a representation of f^i independently for each impression type i: We can show that f^i is concave and admits the following representation:

$$f^i(B) = \begin{cases} u^1 B + v^1, & \text{if } B \in [0, B_1^i) \\ \dots & \\ u^j B + v^j, & \text{if } B \in [B_{j-1}^i, B_j^i) \\ \dots & \\ u^l B + v^l, & \text{if } B \in [B_{l-1}^i, \infty). \end{cases}$$

The number of pieces is $l \leq d + 1$ where d is the number of non-zero coefficients in the constraint $\sum_{e:i(e)=i} c_e z_e \leq B^i$. Moreover, this representation can be computed in $O(d \log d)$ time. Denote the k-th 'segment' of this representation by the tuple (u^k, v^k, Δ^k) where $\Delta^k = B^i_k - B^i_{k-1}$ is the length of the segment with the convention that $B^i_0 = 0$ and $B^i_l = \infty$. Define u^k as the slope of segment k.

2. Sort segments across all f^i by slope (the key step that cannot be solved independently across all impression types): Construct an ordered list L consisting only of segments with positive slope, with the property that segment slope is non-increasing in the order of the list. This step requires $E \log E$ operations.

3. Build a solution: We must now allocate the budget $\tilde{B}$ in (4.4) across impression types. We do this as follows: Consider segments in the list L in order. For segment (u^k, v^k, Δ^k) allocate the smaller of the remaining budget and Δ_k to that segment and decrement the remaining budget.

4. Construct primal solution: Given the optimal individual budget allocation to each subproblem, the primal solution to each subproblem can be computed using complementary slackness.

Before giving proofs for the validity of the algorithm's steps, we note that:

1. The *key computational step in the procedure above is the sort entailed in constructing the list* L; the remaining procedures are easily solved as (easy) independent sub problems. In particular, our overall algorithm will spend the bulk of its runtime in this step so that such an algorithm will benefit tremendously from a system that is optimized to sort very large sets of numbers.

2. The relaxation provides an upper bound on the true optimum and, consequently, a certificate for the optimality gap.

Lemma 4.3.1 *The optimal value of the i-th subproblem with respect to its assigned* *budget B^i is a piecewise linear function of the form*

$$f^i(B) = \begin{cases} u^1 B + v^1, & \textit{if } B \in [0, B^i_1) \\ \dots & \\ u^k B + v^k, & \textit{if } B \in [B^i_{k-1}, B^i_k) \\ \dots & \\ u^l B + v^l, & \textit{if } B \in [B^i_{l-1}, \infty). \end{cases}$$

Additionally, $l \leq d+1$ where where $d = |\{e : i(e) = i\}|$, and the coefficients specifying f^i can be computed in $O(d \log d)$ time.

Proof Consider the feasible region of the dual as illustrated in Figure 4.3.1. Although there are $(d+2)^2$ possible intersections between the inequalities defining this feasible region, each constraint cannot create two intersections points with other constraints that lie on the envelope of the feasible region. Hence, the feasible region has at most $d+1$ extreme points $(u^1, v^1), \dots, (u^l, v^l)$ with $l \leq d+1$, and since the feasible region is convex, it must also be that $v^k \leq v^{k+1}$ and $u^k \geq u^{k+1}$.

It remains to prove that f^i indeed has the threshold structure from the lemma's statement. Clearly, for the $(k+1)$-th extreme point to achieve at least the objective value than the k-th, it must be that:

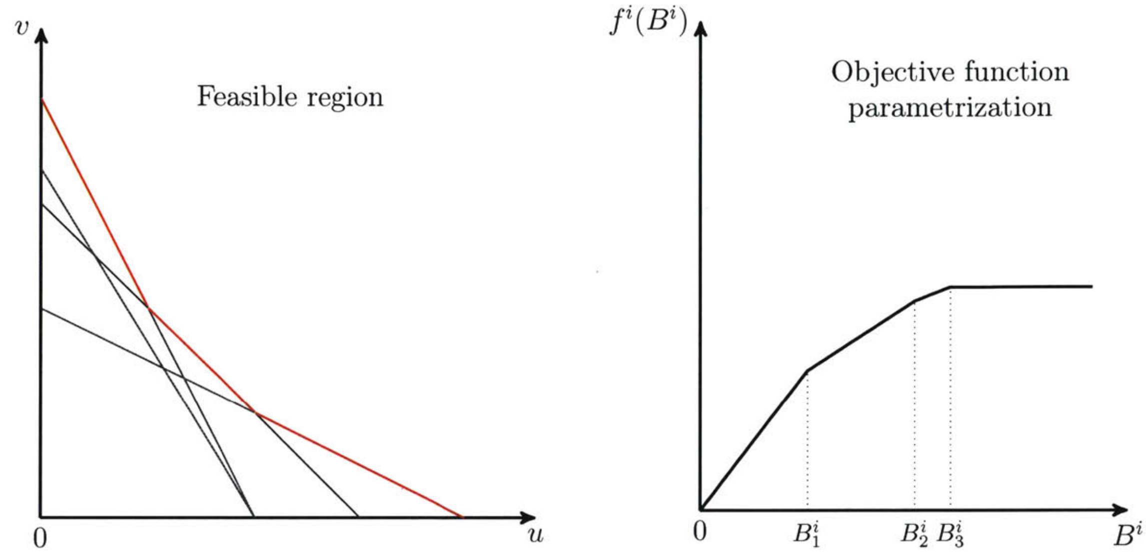
$$B^i > \frac{v_{k+1} - v_i}{u_k - u_{k+1}}$$

and a direct inductive argument yields that extreme point (u^k, v^k) is optimal for B^i in the range

$$[B^i_{k-1}, B^i_k) = \left[\frac{v_k - v_{k-1}}{u_{k-1} - u_k}, \frac{v_{k+1} - v_k}{u_k - u_{k+1}}\right).$$

Moreover, it is a well known fact in computational geometry that finding the active constraints that define the envelope can be reduced to finding the convex hull of at most $d+1$ points in the plane (Har-Peled (2011)). Hence, the constraints that determine the upper envelope can be computed in $O(d \log d)$ time.

Figure 4-2: Feasible regions and objective value parametrization of the i-th subproblem.



Having parametrized the subproblems, we now compute the optimal partitioning of B, i.e. find:

$$\begin{aligned} (B^1, \ldots, B^I) \in \operatorname{argmax} \quad & \sum_i f^i(B^i) \\ \text{subject to} \quad & \sum_i B^i \leq B, \\ & B^i \geq 0 \quad \forall\, i, \end{aligned} \tag{4.4}$$

where f^i is the parametrization of the objective of the i-th subproblem (as in Lemma 4.3.1). Fortunately, it turns out that the optimal solution to this problem has a similar structure to the solution to a simple fractional knapsack program. In particular, the following algorithm computes the optimal budget partition $(B^1, \ldots, B^I)$ by simply allocating the total budget B in order of highest marginal return:

---

initialize $L = \emptyset$ and $B^i = 0, \forall i$

initialize unallocated budget $r = B$;

for $i = 1, \dots, I$ **do**

calculate parameters of subproblem i's segments $\{(u_i^k, v_i^k, \Delta_i^k)\}$ of f^i;

add all slopes $u_i^k > 0$ of f^i to the list L;

end for

sort L in decreasing order;

while $r > 0$ **do**

pick next $u_i^k \in L$ and increase B^i by $\min\{r, \Delta_i^k\}$;

end while

Informally, the algorithm initializes $(B^1, \dots, B^n)$ to 0 and increases the resource allocation B^i to the subproblem i with the highest possible marginal return, given by u_j^i, to the point that either B is exhausted or it not possible to increase budgets with positive marginal returns. Hence, the computational primitive that we require is the ability to sort all slopes (i.e. marginal returns) in decreasing. The theorem below shows that such a procedure indeed computes an optimal allocation of B.

Theorem 4.3.2 *An optimal resource partition $(B^1, \dots, B^I)$ that minimizes (4.4) is found in $O(E \log E)$ time by the above algorithm.*

Proof We show by contradiction that the optimal solution has the structure output by our algorithm. Let us account for the optimal resource allocation $(B^1, \dots, B^I)$ in a more detailed way: in particular, let δ_i^k be the resource amount that is allocated in this optimal solution to the k-th segment of subproblem i; by definition $\sum_k \delta_i^k = B^i$. There exists an optimal solution with all $B^i \le \sum \Delta_i^k$, since increasing budget past that point does not change the objective value of subproblem i.

Our algorithm produces an allocation with the following property

$$\text{for any indices } i_1, i_2, k_1, k_2 \text{ with } u_{i_1}^{k_1} > u_{i_2}^{k_2},\ \delta_{i_2}^{k_2} > 0 \text{ iff } \delta_{i_1}^{k_1} = \Delta_{i_1}^{k_1} \tag{4.5}$$

Let us assume for a contradiction that the optimal resource allocation vector does not have this property. Then, there must exist indices i_1, i_2, k_1, k_2 with $u_{i_1}^{k_1} > u_{i_2}^{k_2}$ and

$\nu > 0$ such that $\delta_{i_1}^{k_1} \leq \Delta_{i_1}^{k_1} - \nu$ and $\delta_{i_2}^{k_2} > \nu$ (this corresponds to a scenario where the algorithm has allocated some amount ν to a segment with a lower slope instead of one with a higher one). Then, one can decrease $\delta_{i_2}^{k_2}$ by ν and increase $\delta_{i_1}^{k_1}$ by the same amount; this will preserve the overall allocation level $\sum_i B^i$, while it will increase the objective value by $\nu(u_{j_1}^{i_1} - u_{j_2}^{i_2}) \geq 0$, leading to a solution that strictly dominates the one we had started with. Therefore any optimal budget allocation must have property (4.5). It can be easily verified that any allocation satisfying this property yields the same objective value, thus proving that the algorithm terminates with an optimal objective.

Since the algorithm needs to sort at most as many ratios as there are edges in the graph, the complexity is $O(E \log E)$.

So far, we have built a vector of allocations of B to each subproblem that achieves the optimum. What is left to do is find a primal assignment that corresponds that this optimum value. The following lemma provides this by proving that it is easy to convert optimal dual solutions to optimal primal solutions. It relies on a standard complementary slackness-based argument which we delay to the appendix.

Lemma 4.3.3 *Given a level of resource allocation B^i and an optimal dual solution (u, v), the optimal (primal) solution to the i-th subproblem can be found in $O(d)$ time.*

4.3.2 Updating Weights.

So far we have shown how given a set of weights w one can solve Relax(w) efficiently. In this section, we make use of the multiplicative weights machinery due to the Plotkin, Shmoys and Tardos (Plotkin et al. (1991)) to yield an algorithm that will convergence to an optimal solution in an appropriate sense by solving a sequence of such relaxation linked together by multiplicative updates of the weight vector w.

Given an initial set of weights w^t, and a solution to the program Relax(w^t) (obtained via the procedure above), we proceed to generate an updated set of weights w^{t+1} as follows. Now a solution to Relax(w^t) may well violate some (if not most) of the resource constraints in the LP we are trying to solve, (4.1). We first compute the extent by which each of these constraints is violated as defined by

$$v_a(z) = \sum_e c_{a,e} z_e - B_a.$$

Before we update weights we define an error parameter ϵ and a parameter ρ typically called the LP *width* which equals $\max_a \max_z |v_a(z)|$. Now we update weights according to

$$w_a^{t+1} = w_a^t \left(1 - \epsilon \frac{v_a(z)}{\rho}\right).$$

Finally, we normalize w^{t+1} so that the entire weight vector sums to unity.

We are left with showing that the algorithm does not need to perform too many weight updates before converging to a good solution. As hinted before, we appeal to the multiplicative weights (MW) framework for solving packing/covering LPs due to Plotkin et al. (1991) and Arora et al. (2005). We note however that the framework gives us algorithmic freedom in terms of what relaxation to choose for (4.1). In particular, a first attempt at choosing a relaxation for which the multiplicative weights algorithm could be applied would relax all constraints of the LP; in this case, Relax(w) would become a fractional knapsack which, as mentioned in the previous section, admits a solution that is both very efficient and amenable to being Map Reduced. Such a straightforward knapsack relaxation fails to take advantage of the special structure of our problem, and leads to both a higher number of iterations in theory (depending on $\log(IA)$ instead of $\log A$) as well as worse practical performance. In fact, it is our intuition that the encouraging practical performance which we will highlight in our experiments section is due to this particular choice of relaxation rather than due to the choice of weight update method.

The following theorem bounds the number of weight updates, and is a application of of the machinery from Arora et al. (2005):

Theorem 4.3.4 *Let $\delta > 0$ be an error parameter and $\epsilon = \min\left\{\frac{\delta}{4\rho}, \frac{1}{2}\right\}$. For $T \geq \frac{16\rho \log A}{\delta^2}$, consider the sequence $z^1, \ldots, z^T$ of solutions obtained thorough performing T*

multiplicative weight updates with update step size ϵ. Let $\overline{z}$ be the average solution, $\overline{z} = \sum_{t=1}^{T} z^t$. Then $\overline{x}$ satisfies:

1. *$\overline{z} \in \chi$ and $\overline{z}$ is δ-feasible for all resource constraints, $\sum_e c_{a,e}\overline{z}_e \leq B_a + \delta, \forall\ a$.*

2. *The objective value of $\overline{z}$ at least equal to the optimal objective of (4.1).*

Proof The first part follows directly from applying Corollary 4 from Arora et al. (2005). The second part follows from the fact that the objective value of each relaxation is greater by construction than the optimal objective of (4.1).

Before proceeding, we note that this approach is completely symmetrical: instead of relaxing all resource constraints while keeping the source constraints explicit, it is equally possible to relax the source constraints instead, and obtain a "mirrored" relaxation

$$\begin{aligned}
\max_{z\geq 0} & \sum_e p_e z_e \\
\text{s.t.} & \sum_e c_{a,e} z_e \leq B_a \quad \forall\ a \\
& \sum_e w_{a(e)} z_e \leq \sum_a w_a.
\end{aligned}$$

A mirrored argument could show what the complexity of solving this relaxation also reduced to a non-linear knapsack problem which mirrors the $O(E \log E)$ one from the previous section. While in this case, the theoretical number of iterations increases from $\log A$ to $\log I$, using this relaxation is worthwhile in practice. In fact, in our experiments described in Section 4.3.4, we test both approaches and find that choosing between these two relaxations carefully can have significant impact on performance. In addition, as a direction of future research, we think that the idea of using both relaxations in parallel could lead to an algorithm with improved performance over the one considered here.

4.3.3 Distributed Implementation of the Algorithm.

While a comprehensive systems design document is beyond the scope of this paper, we outline how the inner algorithm is expressible in the MapReduce framework. One solve of Relax(w) is carried in two Map-Reduce steps, with the main challenge being performing the large and computing a value to the non-linear knapsack (4.4). The pseudocode is given below:

MapReduce round 1

Mapper i

Input: subproblem i data

Calculate the set of slopes and budget cutoffs of f^i,

$$L_i = \{(i, j, u_i^k, \Delta_i^k), \text{ for all } j \in [|\{e : i(e) = i\}| + 1]\}$$

Emit $(\texttt{key, value}) = (i, j, u_i^k, \Delta_i^k)$

Partitioner

Sample `num_reducers` keys from $\cup_i L_i$

Assign u_i^k to Reducer j iff $\texttt{key}[j-1] \leq u_i^j < \texttt{key}[j]$

Reducer j

$B_j = 0$

for all u_i^k received **do**

$B_j \leftarrow B_j + \Delta_i^k$

$L^j = L^j \cup \{(i, j, u_i^k, \Delta_i^k)\}$

end for

Emit (j, B_j, L_j)

Synchronization round

Compute threshold j^* at which $\sum_{1 \leq j \leq j^*} B_j \geq B$

Output $(j^*, B - \sum_{1 \leq j \leq j^*-1} B_j)$

MapReduce round 2

Mapper k

Input: $j^*, B - \sum_{1 \leq j \leq j^*-1} B_j, L_j$

if $j \neq j^*$ **then**

Allocate Δ_i^k of B along all slopes u_i^k in L_j

else

Allocate B in increasing slope order up to level $B - \sum_{1 \leq j \leq j^*-1} B_j$

end if

for all slopes in L_j **do**

Emit (`key, value`) $= (i, (u_i^k, i, k, \texttt{resource\_allocation}))$

end for

Reducer i

Aggregate optimal resource allocation B^i for subproblem i

Compute and output primal solution of subproblem i

In the first Map Reduce step, the segments describing each subproblem are computed and sorted. Note that, as in the implementation of TeraSort from O'Malley (May 2008), we use a customized partitioner with randomized pivots for the aggregate list of segment slopes output by the mappers; this ensures that the list of slopes is globally sorted, and that the load on each reducer is balanced. The aggregate list of *sorted* segment slopes is then partitioned and sent to reducers, each of which then calculates how much budget would be used if each segment were to receive an allocation.

Since the slopes were partitioned to the reducers after being sorted, it is now straightforward for a synchronization round to go through the reducers and figure out the reducer index j^* past which the aggregate allocation exceeds B. Finally, in the second Map Reduce, the optimal resource allocations to each segment can be computed given j^* and, consequently, the optimal primal assignment.

4.3.4 Experimental Performance.

We implemented a shared memory parallel version of our algorithm using C++ and OpenMP. The code can be found at: `https://sites.google.com/site/nips2014mrlp/`. We run our experiments on a machine with dual 2.93GHz Intel Xeon 6-core processors and 128Gb of memory. While our algorithm is ultimately designed to port to the *non*-shared memory model, the purpose of this shared memory implementation is to allow benchmarking versus: (i) state-of-the-art shared memory implementations of simplex: the benchmark we use is CPLEX 12.5 (using primal simplex) and (ii) first-order methods that are also amenable to distributed implementations: we pick the primal/ dual method ("AW P/D") of Awerbuch and Khandekar (2008); Manshadi et al. (2013) as a benchmark and note that this method also employs multiplicative updates for the dual variables.

We test our implementation on a family of synthetic instances of Ad-Display and AdWords problems. Each instance has 1mm impressions and 1mm advertisers. We generate the matrix of bids from advertisers to impressions in the following way: for each advertiser, we choose 100 impressions uniformly at random for which we assign an edge (i.e., a non-zero bid.) At this sparsity, storing the bid matrix (represented as a list) in memory requires 3Gb. We generate the bid values according to a factor model that is designed to simulate "hot"/"cold" advertisers and impressions. In particular, for each impression i, we generate a 5-dimensional feature vector ϕ_i with each component drawn i.i.d. from a log-normal distribution. We also generate a similar feature vector θ_a for each advertiser. We then set $b_{i,a} =< \phi_i, \theta_a >$ for each impression to advertiser edge.

We choose advertiser budgets in order to simulate several load factor scenarios. Depending on the instance type, we define its load factor in alternate ways:

$$\mathrm{LF} = \begin{cases} \frac{\sum_a B_a}{I} & \text{if instance is Ad-Display} \\ \frac{\sum_a B_a}{I\mathsf{E}[b_{i,a}]} & \text{if instance is AdWords} \end{cases}$$

We set the budgets B_a uniformly to achieve load factors taking values in $\{0.5, 0.75, 1, 1.5, 2\}$. We heuristically initialize the multiplicative weight of advertiser a to be proportional to the average bid going into a ($w_a \propto \sum_i b_{i,a}/100$.) For both sets of experiments, we set $\epsilon = 0.5$. Also, we set a burn-in period for which still do the regular multiplicative updates but which we do not count into the calculation of the primal solution; we use a burn-in of either 25 or 50 iterations and report whichever run results in fewer iterations. As noted before, our algorithm works analogously if we relax the source (impression) constraints rather than the resource (advertiser) ones; in the results below, we report performance of the best.

Table 4.1 reports the number of iterations required for our algorithm to reach 95% optimality and Figure 4-3 shows the speed of convergence for a particular Ad-Display instance where we set the load factor to 1. We note that the algorithm is fast at reaching 95% optimal solutions, requiring fewer than 100 iterations. Since in many distributed frameworks such as MapReduce, the setup time to run one round is quite high, an algorithm which requires a small number of rounds is highly desirable. The slow convergence beyond the 95% point can be improved by lowering ϵ at the expense of slower initial progress.

Table 4.2 compares the wall clock times required by our algorithm versus CPLEX and an implementation of AW P/D [1]; we measure the time required to reach 95% the optimal values for out set of Ad-Display and, respectively, AdWords experiments:

1. Comparison with CPLEX: Our algorithm is consistently as fast, if not better, than CPLEX. This is quite surprising, since CPLEX is optimized for speed in multicore shared memory environments.

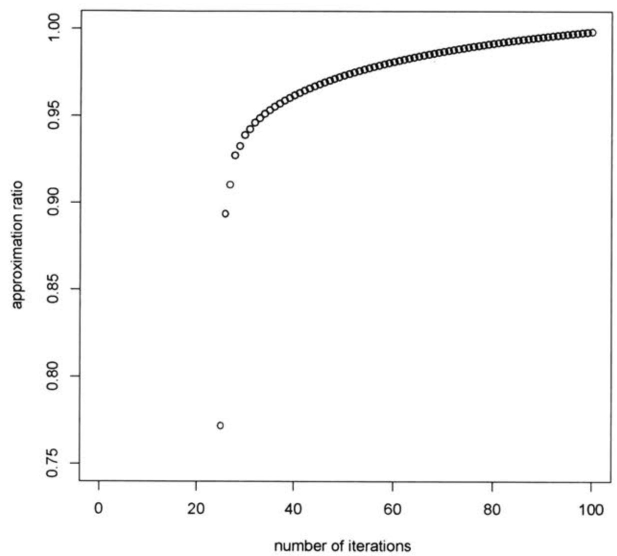
2. Comparison with AW P/D: On Ad-Display instances, AW P/D has a running time of almost a factor of magnitude larger and, in fact, the best approximation we could achieve with AW P/D was only around 90% for some of our problems. Moreover, we observe that it requires a much larger number of iterations compared to our approach. We interpret this as an indication that the buy in our algorithm does not necessarily come from the use of a multiplicative update rule, but rather from that fact that we solve a particularly tight relaxation of the original feasible space. On our AdWords instances, AW P/D is competitive with our scheme and CPLEX; this is consistent with our intuition that AdWords type instances should be easier to approximate due to the one

[1] We also use the primal initialization and dynamic stepsizes employed in Manshadi et al. (2013)

Table 4.1: Iteration count to 95% optimality.

Instance Type \ LF	0.5	0.75	1	1.5	2
Ad-Display	52	48	59	55	73
AdWords	34	46	56	59	62

Figure 4-3: Progress per iteration for our algorithm for an Ad-Display instance with LF= 1.



to one correspondence between the objective value and the advertiser budget constraints.

4.4 Conclusions.

We have proposed an algorithm for the classical Deterministic Linear Program that forms the basis to solving many network revenue management problems of interest. Our algorithm takes advantage of the special structure of the DLP to reduce the problem of computing an optimal solution to the problem of repeatedly sorting a large vector; compared to methods such as simplex which rely on a pivot or matrix inversion step, this approach is particularly amenable to implementations in decen-

Table 4.2: 95% optimality wall clock times (in minutes) for Ad-Display.

	Ad Display				
LF / Algo	0.5	0.75	1	1.5	2
MW	10.76	10.58	13.75	11.73	14.73
CPLEX	14.86	15	15.3	16.45	14.91
AW P/D Algo	86.66	110.2	132.2	>200	>200

Table 4.3: 95% optimality wall clock times (in minutes) for AdWords.

	AdWords				
LF / Algo	0.5	0.75	1	1.5	2
MW	7.03	9.51	13.06	11.38	11.72
CPLEX	15.23	15	15.3	15.71	14.91
AW P/D Algo	8.8	12.3	20.1	78.1	180.6

tralized models of computation like Map Reduce, making it particularly attractive to extremely high dimensional applications such display advertising or sponsored search ad allocation where the data is so large that it must be distributed across multiple machines.

We find quite surprising that, beyond the scalability properties mentioned above, experiments suggest our method is highly competitive versus established linear optimization solvers which employ optimized simplex methods. Moreover, our method versus is superior to other first order methods which are generic to packing/covering LPs and which in theory would provide alternative candidates for Map Reduce implementations; this suggests that choosing our particular relaxation (instead of choosing for example a relaxation where we relaxed *all* constraints is valuable.

There are several future directions we find interesting:

1. Alternating relaxations: Our algorithm works symmetrically for the cases where we relax resource or source constraints. It would be interesting to see if an algorithm which alternates between solving a resource constraints relaxation and a source constraints one could yield better convergence.

2. Parallel multiplicative weight rounds: Another natural extension is to ask whether something one could run multiple multiplicative weight relaxations in parallel: in particular, is there something to be gained by either (i) starting from k different weight vectors and running k multiplicative updates with difference seeds or (ii) picking k different sets of constraints which can conveniently relaxed and running the k different relaxations in parallel?

3. Warm start guarantees: As suggested, the setting that is most interesting to us is one where the LP is solved repeatedly over the length of an ad campaign. Access to warm starts could potentially strengthen the convergence guarantees for our algorithm, as well as lead to better performance in practice.

Conclusions

In this thesis, we tackle several problems specific to modern applications such as online advertising; these problems are relevant in a broader scope to high dimensional network revenue management. These applications are challenging due to their extreme granularity – for example, an ad platform will observe hundreds of millions of different customer classes in a single day's worth of traffic, whereas in more traditional RM applications such as airline yield management the overall number of customer classes would not exceed several hundreds.

The motivation for Part I of this thesis is that in regime where customer classes are extremely fine grained, it becomes extremely difficult to forecast the idiosyncratic evolution of each individual customer demand stream. Hence we focus on designing schemes that (i) are robust to large demand shocks, (ii) make use of available historical data, but (iii) do not require making sophisticated forecasts about the future evolution of demand. To the best of our knowledge, our model predictive control approach admits the first constant factor guarantees against arbitrary volatility for a broad class of network revenue management problems including optimal ad allocation for sponsored search and advertising. From a theoretical standpoint, these are the first results of this type in the area of model predictive control. Lastly, the scheme yields nearly optimal performance in our experiments with real ad traffic volatility.

While Part I deals with handling demand uncertainty at a global time scale, Part II is concerned with demand uncertainty at the local time scale. In particular, assuming that we were looking at a small enough timescale that the demand distribution driving the arrivals of the different types of customers was static, we examine the sample complexity of learning such a highly dimensional probabilistic object. Here we study a particular NRM model that is specific to optimal allocation for display advertising markets. Its special structure allows us to arrive at a graceful dependence of the sample complexity in the problem parameters; quite surprisingly, we in fact show that under mild assumptions on the NRM instance, the sample complexity depends quite gracefully on the number of customer types.

Part III of the thesis focuses on computational issues related to making the above approaches practical at scale. We propose an algorithm for solving network revenue management linear programs that tailored towards distributed computational infrastructure such as Map Reduce. Remarkably, we observe that our algorithm also benchmarks quite favorably in terms of speed against the state-of-the-art shared memory solvers for linear programming. Our algorithm leverages the special structure of NRM LPs which are a special case of a packing LP; experimentally, we show that it outperforms other generic paralellizable methods for packing.

More generally, considering the decreasing cost of acquiring data in recent years, we expect that many other classical operational problems will become high dimensional as customer behavior is modeled at increasingly granular levels, as has been the case with the transition from advertising in traditional media to advertising over the Internet. The present thesis attempts to introduce three prototypical challenges we encounter in such novel settings and propose an optimization toolkit to tackle them in systematic way. Lastly, we note our main area of application, online advertising, opens up many other research directions in modern revenue management that can build upon the present work.

Appendix A

Proofs

A.1 Proofs for Section 3.2.

Proof of Lemma 1.2.2 Employing the notation $\overline{\sigma}_{T,1} \triangleq \frac{1}{T}\int_0^T \sigma_t dt$, we have:

$$
\begin{aligned}
&\frac{\mathsf{E}\left[\frac{1}{T}\int_0^T f(\Lambda_t)dt\right]}{f\left(\mathsf{E}\left[\frac{1}{T}\int_0^T \Lambda_t dt\right]\right)} \\
&\quad= \frac{\mathsf{E}\left[\frac{1}{T}\int_0^T f(\Lambda_t)dt\right]}{f\left(\frac{1}{T}\int_0^T \mathsf{E}\left[\Lambda_t\right]dt\right)} \\
&\quad\geq \frac{\mathsf{E}\left[\frac{1}{T}\int_0^T f\left(\Lambda_t\right)dt\right]}{f\left(\frac{1}{T}\int_0^T \left(\lambda_t + \frac{\sigma_t}{\sqrt{2\pi}}\right)dt\right)} \\
&\quad= \frac{1}{T}\int_0^T \left[\int_{-\infty}^{\infty} \frac{f\left((\lambda+y)^+\right)}{f\left(\lambda + \frac{1}{T}\int_0^T \frac{\sigma_t dt}{\sqrt{2\pi}}\right)} \frac{\exp(-y^2/2\sigma_t^2)}{\sqrt{2\pi\sigma_t^2}} dy\right] dt \\
&\quad\geq \frac{1}{T}\int_0^T \left[\int_{-\infty}^{\infty} \min\left\{\frac{(\lambda+y)^+}{\lambda + \frac{1}{T}\int_0^T \frac{\sigma_t dt}{\sqrt{2\pi}}}, 1\right\} \frac{\exp(-y^2/2\sigma_t^2)}{\sqrt{2\pi\sigma_t^2}} dy\right] dt \\
&\quad= \frac{1}{T}\int_0^T \left[1 - \Phi\left(\frac{\overline{\sigma}_{T,1}}{\sigma_t\sqrt{2\pi}}\right) + \int_{-\lambda}^{\overline{\sigma}_{T,1}/\sqrt{2\pi}} \frac{\lambda+y}{\lambda + \overline{\sigma}_{T,1}/\sqrt{2\pi}} \frac{\exp(-y^2/2\sigma_t^2)}{\sqrt{2\pi\sigma_t^2}} dy\right] dt \\
&\quad\geq \frac{1}{T}\int_0^T \left[1 - \Phi\left(\frac{\overline{\sigma}_{T,1}}{\sigma_t\sqrt{2\pi}}\right) + \int_{0}^{\overline{\sigma}_{T,1}/\sqrt{2\pi}} \frac{y}{\lambda + \overline{\sigma}_{T,1}/\sqrt{2\pi}} \frac{\exp(-y^2/2\sigma_t^2)}{\sqrt{2\pi\sigma_t^2}} dy\right] dt \\
&\quad\geq 0.342
\end{aligned}
$$

In the above sequence of bounds, the first inequality comes from the fact that $f(\cdot)$ was assumed non-decreasing, and the fact that for a Normal random variable with mean μ and variance σ^2, one has

$$\mathsf{E}[X^+] \leq \mu + \frac{\sigma}{\sqrt{2\pi}},$$

so that $\mathsf{E}[\Lambda_t] \leq \lambda_t + \sigma_t/\sqrt{2\pi}$. The second equality follows from Fubini's theorem. The second inequality follows from property 3 of Lemma 1.3.7 applied to $f(\cdot)$. The final inequality is Lemma 14 of Chen and Farias (2013).

A.2 Proofs for Section 1.3.2.

Proof of Proposition 1.3.1 Let $\{z_t\} \in \Pi_N$ be an ϵ-optimal admissible control policy feasible for (2.3) (which exists for arbitrary $\epsilon > 0$, see Bertsekas and Shreve (2007)). For a given sample path ω, define

$$\hat{z}_e = \frac{\int_0^T \Lambda_{i(e),t} z_{e,d(t)} \mathbf{1}_{\{I_{e,t}\}} dt}{\int_0^T \Lambda_{i(e),t}}$$

Then $\hat{z}_e$ is feasible for (1.2) by the definition of $I_{e,t}$ so that it yields a solution to (1.2) of value no greater than $J^*_{\{\Lambda_t\}}(x_0)$. Moreover, the value of this solution is precisely the value garnered by $\{z_t\}$ on ω. Taking expectations yields $\mathsf{E}\left[J^*_{\{\Lambda_t\}}(x_0)\right] \geq J^{*,N}(x_0) - \epsilon$. Since our choice of $\epsilon > 0$ was arbitrary, the result follows.

A.3 Proofs for Section 1.3.3.

Proof of Lemma 1.3.2 Observe that the quantity of resource type k consumed by edge e in the interval $[jT/N, (j+1)T/N)$ is at least

$$\frac{T}{N}\left(\Lambda^{\min}_{i(e),jT/N} z^{\mathrm{R}}_{e,jT/n} A_{k,e} - \sum_{e' \neq e} \left(\Lambda^{\max}_{i(e'),jT/N} - \Lambda_{i(e'),jT/N}\right) z^{\mathrm{R}}_{e',jT/n} A_{k,e'}\right)$$

It follows that the revenues garnered along edge e over that interval are at least

$$\frac{T}{N}p_e\left(\Lambda^{\min}_{i(e),jT/N}z^{\mathrm{R}}_{e,jT/n} - C_e\sum_{e'\neq e}\left(\Lambda^{\max}_{i(e'),jT/N} - \Lambda_{i(e'),jT/N}\right)\right)$$

where $C_e = \max_{\{e',k:A_{k,e}>0\}} A_{k,e'}/A_{k,e}$. It then follows that total revenues over the interval are at least

$$\begin{aligned}
&\frac{T}{N}\Bigg(\sum_e p_e z^{\mathrm{R}}_{e,jT/n}\Lambda_{i(e),jT/N} + \sum_e p_e z^{\mathrm{R}}_{e,jT/n}\Lambda^{\min}_{i(e),jT/N} - \sum_e p_e z^{\mathrm{R}}_{e,jT/n}\Lambda_{i(e),jT/N}\\
&\quad - \sum_e C_e p_e \sum_{e'\neq e}\left(\Lambda^{\max}_{i(e'),jT/N} - \Lambda_{i(e'),jT/N}\right)\Bigg)\\
\geq&\frac{T}{N}\left(\sum_e p_e z^{\mathrm{R}}_{e,jT/N}\Lambda_{i(e),jT/N} - C\left(\sum_e p_e\right)\sum_e\left(\Lambda^{\max}_{i(e),jT/N} - \Lambda^{\min}_{i(e),jT/N}\right)\right)\\
=&\frac{T}{N}\left(\frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_{jT/N}\right)}{T - jT/N} - C\left(\sum_e p_e\right)\sum_e\left(\Lambda^{\max}_{i(e),jT/N} - \Lambda^{\min}_{i(e),jT/N}\right)\right)
\end{aligned}$$

where $C = \max C_e$. Summing over intervals yields the result.

Proof of Corollary 1.3.3 Given the continuity of the sample path $\Lambda_{i,t}$ in t for all i, we have that:

$$\lim_N \frac{T}{N}\sum_{j=0}^{N-1}\Lambda^{\max}_{i,jT/N} = \lim_N \frac{T}{N}\sum_{j=0}^{N-1}\Lambda^{\min}_{i,jT/N}.$$

for all i. The result then follows from Lemma 1.3.2.

Proof of Corollary 1.3.5 We begin with making two elementary observations. First,

$$\mathrm{LP}(\Lambda, t, x-\delta) \geq \mathrm{LP}(\Lambda, t, x) - \sum_k \delta_k \frac{\sum_e p_e}{\min_{e:A_{k,e}>0} A_{k,e}}.$$

Second, $\mathrm{LP}(t, \Lambda, x)$ is a component-wise monotone function of x. These observations with the result of the balancing lemma then immediately yield:

$$
\begin{aligned}
&\frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_{jT/N}\right)}{T - jT/N} \\
&\geq \frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_0(N-j)/N\right)}{T - jT/N} + \\
&\sum_k \left[\sum_{l=0}^{j-1} \sum_e \frac{T}{N} A_{k,e} \left(\Lambda_{i(e),lT/N} - \Lambda^{\max}_{i(e),lT/N}\right)\right] \frac{\sum_e p_e}{\min_{e:A_{k,e}>0} A_{k,e}} \\
&\geq \frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_0(N-j)/N\right)}{T - jT/N} + M\left[\sum_{l=0}^{N-1} \sum_i \frac{T}{N} \left(\Lambda^{\min}_{i,lT/N} - \Lambda^{\max}_{i,lT/N}\right)\right]
\end{aligned}
$$

where $M \triangleq \frac{\max_{k,e} A_{k,e}}{\min_{k,e:A_{k,e}>0} A_{k,e}} (\sum_e p_e) KE$. Consequently,

$$
\begin{aligned}
&\frac{T}{N} \sum_{j=0}^{N-1} \frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_{jT/N}\right)}{T - jT/N} \\
&\geq \frac{T}{N} \sum_{j=0}^{N-1} \left(\frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_0(N-j)/N\right)}{T - jT/N} + M\left[\sum_{l=0}^{N-1} \sum_i \frac{T}{N} \left(\Lambda^{\min}_{i,lT/N} - \Lambda^{\max}_{i,lT/N}\right)\right]\right) \\
&= \frac{T}{N} \left(\sum_{j=0}^{N-1} \frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, x_0(N-j)/N\right)}{T - jT/N} + MT \sum_{l=0}^{N-1} \sum_i \left(\Lambda^{\min}_{i,lT/N} - \Lambda^{\max}_{i,lT/N}\right)\right)
\end{aligned}
\tag{A.1}
$$

Then, using the fact that the continuity of the sample path $\Lambda_{i,t}$ in t for all i yields

$$
\lim_N \frac{T}{N} \sum_{j=0}^{N-1} \Lambda^{\max}_{i,jT/N} = \lim_N \frac{T}{N} \sum_{j=0}^{N-1} \Lambda^{\min}_{i,jT/N}.
$$

for all i, we may take the limit infimum on both sides of (A.1) to arrive at the result.

A.4 Proofs for Section 1.3.4.

Proof of Lemma 1.3.7 We have

1. That $f(0) = 0$ and f is continuous and non-decreasing follows immediately from the definition of f.

2. $f(\cdot)$ is concave since $\min\left\{\frac{w}{\Lambda}, 1\right\}$ is concave, and since summations preserve concavity.

3. For $w \geq v$, $\min\left\{\frac{w}{v}, 1\right\} = 1$ and $\frac{f(w)}{f(v)} \geq 1$ by virtue of f being non-decreasing. For $w < v$, since f is concave, and $w, v > 0$ with $w/v < 1$ we have $f(w) = f(0 + w/v \cdot v) \geq w/v \cdot f(v)$. Thus, $\frac{f(w)}{f(v)} \geq \frac{w}{v}$.

4. The result is a consequence of Jensen's inequality.

Proof of Lemma 1.3.8 Define $\tilde{z} \in \mathbb{R}^E$ according to $\tilde{z}_e = z_e(t, \Lambda, x)\min\{\Lambda_{i(e)}/u_{i(e)}, 1\}$. Observe that $\tilde{z}$ is a feasible solution to LP(t, u, x). In particular,

$$\begin{aligned}
\sum_e A_{k,e}\tilde{z}_e u_{i(e)}(T-t) =& \sum_e A_{k,e} z_e(t, \Lambda, x)\min\{\Lambda_{i(e)}/u_{i(e)}, 1\} u_{i(e)}(T-t) \\
\leq& \sum_e A_{k,e} z_e(t, \Lambda, x)\Lambda_{i(e)}(T-t) \\
\leq& x_k
\end{aligned}$$

where the last inequality is due to the feasibility of $z(t, \Lambda, x)$ for LP(t, Λ, x). Further, $\tilde{z} \in \mathcal{Z}$ by construction. Now, the objective value of this solution is precisely $\sum_{i=0}^{I} f_i^{(t,\Lambda,x)}(u_i)$, and the result follows since the value of an optimal solution to LP(t, u, x) should be at least as large.

A.5 Proofs for Section 1.3.6.

Proof of Proposition 1.3.12 Begin by considering a policy $\{z_t^{\mathrm{U}}\}$ defined according to $z_{t,(1,1)}^{\mathrm{U}} = 0, z_{t,(2,1)}^{\mathrm{U}} = 1$. Clearly, $J^{\mathrm{U}}(x_0) = T^{4/3+\epsilon}/2$. Consequently,

$$J^{\mathrm{UB}}(x_0) \geq T^{4/3+\epsilon}/2. \tag{A.2}$$

Now consider our re-optimization policy with $N = 1$. It is easy to see that for this policy, we have $z_{t,(1,1)}^{\mathrm{R}} = z_{t,(2,1)}^{\mathrm{R}} = 1$. We now compute an upper bound on $J^{\mathrm{R},1}(x_0)$. First, note that $\mathsf{E}[\Lambda_{1,s}] = \frac{1}{2}\sqrt{s}$ and $\mathrm{Var}[\Lambda_{1,s}] = s(\pi - 1)$. Hence, for any $0 \leq t \leq T$, the quantity $\int_0^t \Lambda_{1,s} ds$ has moments $\mathsf{E}\left[\int_0^t \Lambda_{1,s} ds\right] = \frac{t}{2} + \frac{3}{2}t^{3/2}$ and $\mathrm{Var}\left[\int_0^t \Lambda_{1,s} ds\right] = \frac{\pi-1}{2}t^2$. By Chebyshev's inequality, it follows that

$$\mathsf{P}\left[\left|\int_0^t \Lambda_{1,s} ds - \left(\frac{t}{2} + \frac{3}{2}t^{3/2}\right)\right| \geq \frac{3}{4}t^{3/2}\right] \leq \frac{8(\pi-1)}{9}\frac{1}{t}.$$

In particular, with probability at least $1 - \frac{8(\pi-1)}{9t}$, $\int_0^t \Lambda_{1,s} ds \geq \frac{t}{2} + \frac{3}{4}t^{3/2}$. Now, define τ to be the solution to the equation $\tau + \frac{3}{4}\tau^{3/2} = T$. One may consequently interpret τ as the first time t at which $x_t = 0$ assuming the Λ_1 process followed its mean.

Now define the stopping time $\hat{\tau}$ according to $\hat{\tau} = \inf\left\{t : \int_0^\tau \Lambda_{1,s} + \Lambda_{2,s} ds \geq T\right\}$. Now on the event where $\hat{\tau} \leq \tau$, we have:

$$\begin{aligned}
\int_0^{\hat{\tau}} \Lambda_{1,s} ds &= T - \int_0^{\hat{\tau}} \Lambda_{2,s} ds \\
&\geq T - \int_0^{\tau} \Lambda_{2,s} ds \\
&= \frac{\tau}{2} + \frac{3}{4}\tau^{3/2}.
\end{aligned}$$

Consequently,

$$\begin{aligned}
\int_0^{\hat{\tau}} \Lambda_{2,s} ds &= T - \int_0^{\hat{\tau}} \Lambda_{1,s} ds \\
&\leq T - \frac{\tau}{2} + \frac{3}{4}\tau^{3/2} \\
&= \frac{\tau}{2}
\end{aligned}$$

It follows that on the event where $\tau \leq \hat{\tau}$, we have:

$$\begin{aligned}
J^{\mathrm{R},1}_{\{\Lambda_t\}}(x_0) &= T^{1/3+\epsilon}\int_0^{\hat{\tau}} \Lambda_{2,s} + \left(T - \int_0^{\hat{\tau}} \Lambda_{2,s} ds\right) \\
&\leq T^{1/3+\epsilon}\frac{\tau}{2} + T - \frac{\tau}{2}
\end{aligned}$$

On the complementary event, we consider the trivial upper bound

$$J^{\mathrm{R},1}_{\{\Lambda_t\}}(x_0) \leq T^{4/3+\epsilon}$$

Now notice that $\mathsf{P}(\hat{\tau} \leq \tau) = \mathsf{P}\left(\int_0^\tau \Lambda_{1,s} + \Lambda_{2,s} ds \geq T\right) \geq 1 - \frac{8(\pi-1)}{9\tau}$, so that taking expectations immediately yields with the above two inequalities:

$$J^{\mathrm{R},1}(x_0) \leq \left(1 - \frac{8(\pi-1)}{9\tau}\right)\left(T^{1/3+\epsilon}\frac{\tau}{2} + T - \frac{\tau}{2}\right) + \frac{8(\pi-1)}{9\tau} T^{4/3+\epsilon} \tag{A.3}$$

But it is easy to verify that $\tau = \Theta(T^{2/3})$, so that the right hand side of the above inequality is in fact $\Theta(T^{1+\epsilon})$. The result then follows immediately from (A.3) and (A.2).

A.5.1 Proof of Theorem 1.3.14.

First, let us define

$$f(N) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{i=1}^{I}\left(\Lambda^{\max}_{i,jT/N} - \Lambda^{\min}_{i,jT/N}\right).$$

We begin with a simple proposition

Lemma A.5.1

$$\frac{T}{N}\sum_{j=0}^{N-1}\frac{\mathrm{LP}(0, \Lambda_{jT/N}, x_0)}{T} - \frac{1}{T}\int_0^T \mathrm{LP}(0, \Lambda_t, x_0)dt \geq -\max_e p_e \; f(N)$$

Proof We have

$$\begin{aligned}
&\frac{T}{N}\sum_{j=0}^{N-1}\frac{\mathrm{LP}(0, \Lambda_{jT/N}, x_0)}{T} - \frac{1}{T}\int_0^T \mathrm{LP}(0, \Lambda_t, x_0)dt \\
&\geq \frac{1}{T}\sum_{j=0}^{N-1}\left(\mathrm{LP}(0, \Lambda_{jT/N}, x_0)\frac{T}{N} - \mathrm{LP}(0, \Lambda^{\max}_{jT/N}, x_0)\frac{T}{N}\right) \\
&\geq -\max_e p_e \; f(N)
\end{aligned}$$

where the first inequality follows from the monotonicity of $\mathrm{LP}(\cdot,\cdot,\cdot)$ in its second argument, and the final inequality follows from the fact that for any $\delta \geq 0$ and any

$u, x \geq 0$, we must have

$$\mathrm{LP}(0, u+\delta, x) - \mathrm{LP}(0, u, x) \leq \max_e p_e \sum_i \delta_i$$

We next define a few useful 'error terms', namely $h_1(N) = TC(\sum_e p_e)Ef(N)$, $h_2(N) = MT^2 f(N)$, and finally $h_3(N) = \max_e p_e f(N)$. The constants C and M are defined in Lemma 1.3.2 and Corollary 1.3.5 respectively. It will be useful to characterize the rate at which these terms approach zero. In particular, define

$$g(N) = h_1(N) + h_2(N) + h_3(N).$$

We have:

Lemma A.5.2

$$\limsup_N \frac{g(N)}{\sqrt{2\log N/N}} \leq C' \sum_i \sigma_i \quad \text{a.s.}$$

where $C' \triangleq TC(\sum_e p_e)E + MT^2 + \max_e p_e$. *Consequently,*

$$\limsup_N \frac{\mathsf{E}\left[g(N)\right]}{\sqrt{2\log N/N}} \leq C' \sum_i \sigma_i$$

Proof We have that for almost all ω and N sufficiently large,

$$\begin{aligned} g(N) &= C' f(N) \\ &\leq C' \sum_i \frac{1}{N} \sum_{j=0}^{N-1} \max_j \left(\Lambda^{\max}_{i,jT/N} - \Lambda^{\min}_{i,jT/N}\right) \\ &= C' \sum_i \max_j \left(\Lambda^{\max}_{i,jT/N} - \Lambda^{\min}_{i,jT/N}\right) \end{aligned}$$

From the modulus of continuity of the sample paths of the rate process, Theorem 1.3.13, we have that

$$\limsup_N \frac{g(N)}{\sqrt{2\log N/N}} \leq C' \sum_i \sigma_i \quad \text{a.s.}$$

An application of the reverse Fatou's Lemma yields

$$\limsup_N \frac{\mathsf{E}\left[g(N)\right]}{\sqrt{2\log N/N}} \le C' \sum_i \sigma_i$$

We are now in a position to prove our result. In particular:

$$\begin{aligned}
J^{\mathrm{R},N}_{\{\Lambda_t\}}(x_0) &\ge \frac{T}{N}\sum_{j=0}^{N-1} \frac{\mathrm{LP}(jT/N, \Lambda_{jT/N}, x_{jT/N})}{T - jT/N} - h_1(N) \\
&\ge \frac{T}{N}\sum_{j=0}^{N-1} \frac{\mathrm{LP}(jT/N, \Lambda_{jT/N}, x_0(N-j)/N)}{T - jT/N} - h_1(N) - h_2(N) \\
&= \frac{T}{N}\sum_{j=0}^{N-1} \frac{\mathrm{LP}(0, \Lambda_{jT/N}, x_0)}{T} - h_1(N) - h_2(N) \\
&\ge \frac{1}{T}\int_0^T \mathrm{LP}(0, \Lambda_T, x_0)dt - h_1(N) - h_2(N) - h_3(N)
\end{aligned}$$

where the first inequality follows from the proof of Lemma 1.3.2, the second inequality follows from the proof of Corollary 1.3.5, the equality follows from the fact that $\mathrm{LP}(t, \Lambda, x(1-t/T)) = \frac{T-t}{T}\mathrm{LP}(0, \Lambda, x)$, and the final inequality follows from Lemma A.5.1. Now, taking expectations yields

$$\begin{aligned}
\mathsf{E}\left[J^{\mathrm{R},N}_{\{\Lambda_t\}}(x_0)\right] &\ge \mathsf{E}\left[\frac{1}{T}\int_0^T \mathrm{LP}(0, \Lambda_T, x_0)dt\right] - \mathsf{E}\left[g(N)\right] \\
&\ge 0.342\mathsf{E}\left[J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)\right] - \mathsf{E}\left[g(N)\right]
\end{aligned}$$

where the second inequality follows from the proof of our approximation guarantee in the setting with a large number of re-optimizations, Theorem 1.3.9. Defining $\Delta(N) = \mathsf{E}\left[g(N)\right]$ with the result of Lemma A.5.2 yields the proof of the Theorem.

A.6 Proofs for Section 1.4.2.

Proof of Lemma 1.4.1 Let us define $\overline{p} \triangleq \max_e p_e$ and $\overline{A} \triangleq \max_{k,e} A_{k,e}$. We begin by observing that

$$\begin{aligned}
J^{\mathrm{R}-\beta}_{\{\Lambda_t\}}(x_0) &= \int_0^T \sum_e p_e \left((1-\beta)\hat{z}^{\mathrm{R}}_{e,d(t)} + \beta \hat{z}^{\mathrm{D}}_{e,d(t)}\right) \Lambda_{i(e),t} \mathbb{I}_{\{I_{e,t}\}} dt \\
&\geq \int_0^T \sum_e p_e \left((1-\beta)\hat{z}^{\mathrm{R}}_{e,d(t)} + \beta \hat{z}^{\mathrm{D}}_{e,d(t)}\right) \Lambda_{i(e),t} dt \\
&\quad - \overline{p} \sum_k \left(\sum_e \int_0^T A_{k,e} \Lambda_{i(e),t} z^{\mathrm{R}-\beta}_{e,t} dt - x_{0,k} \right)^+ \\
&\geq (1-\beta) J^{\mathrm{R}}_{S,\{\Lambda_t\}}(x_0) + \beta J^{\mathrm{R}}_{D,\{\Lambda_t\}}(x_0) \\
&\quad - \overline{p} \sum_k \left(\sum_e \int_0^T A_{k,e}(1-\beta) \Lambda_{i(e),t} \hat{z}^{\mathrm{R}}_{e,t} dt - (1-\beta) x_{0,k} \right)^+ \\
&\quad - \overline{p} \sum_k \left(\sum_e \int_0^T A_{k,e} \beta \Lambda_{i(e),t} \hat{z}^{\mathrm{D}}_{e,t} dt - \beta x_{0,k} \right)^+
\end{aligned} \tag{A.4}$$

Next, observe that

$$\begin{aligned}
\sum_e \int_0^T A_{k,e} \Lambda_{i(e),t} \hat{z}^{\mathrm{D}}_{e,t} dt &\leq \frac{T}{N} \sum_{j=0}^{N-1} \sum_e \hat{z}^{\mathrm{D}}_{e,jT/N} A_{k,e} \Lambda^{\max}_{i(e),jT/N} \\
&\leq \frac{T}{N} \left(\sum_{j=0}^{N-1} \sum_e \hat{z}^{\mathrm{D}}_{e,jT/N} A_{k,e} \Lambda_{i(e),jT/N} + \overline{A} \sum_{j=0}^{N-1} \sum_e \Lambda^{\max}_{i(e),jT/N} - \Lambda_{i(e),jT/N} \right) \\
&\leq \frac{T}{N} \left(\sum_{j=0}^{N-1} \sum_e z^{\mathrm{D}}_{e,jT/N} A_{k,e} \lambda_{i(e),jT/N} + \overline{A} \sum_{j=0}^{N-1} \sum_e \Lambda^{\max}_{i(e),jT/N} - \Lambda_{i(e),jT/N} \right)
\end{aligned} \tag{A.5}$$

But for any i,

$$\lim_N \frac{T}{N} \sum_{j=0}^{N-1} \Lambda^{\max}_{i,jT/N} - \Lambda_{i,jT/N} = 0$$

by virtue of the continuity of $\Lambda_{i,t}$ and further,

$$\lim_N \frac{T}{N} \sum_{j=0}^{N-1} z^{\mathrm{D}}_{e,jT/N} A_{k,e} \lambda_{i(e),jT/N} = z^{\mathrm{D}}_e A_{k,e} \int_0^T \lambda_{i(e),t} dt,$$

since $\lambda_{i,t}$ is continuous in t for all i. Since, by the definition of z^{D}, we have that $\sum_e z_e^{\mathrm{D}} A_{k,e} \int_0^T \lambda_{i(e),t} dt \leq x_{0,k}$, it then follows from (A.5) that

$$\limsup_N \sum_e \int_0^T A_{k,e} \Lambda_{i(e),t} \hat{z}_{e,t}^{\mathrm{D}} dt \leq x_{0,k}$$

for all k. Consequently, for all k,

$$\lim_N \left(\sum_e \int_0^T A_{k,e} \beta \Lambda_{i(e),t} \hat{z}_{e,t}^{\mathrm{D}} dt - \beta x_{0,k} \right)^+ = 0.$$

Now, since

$$\left(\sum_e \int_0^T A_{k,e} \beta \Lambda_{i(e),t} \hat{z}_{e,t}^{\mathrm{D}} dt - \beta x_{0,k} \right)^+ \leq E\overline{A} \sum_i \max_{t \in [0,T]} \Lambda_{i,t}$$

the dominated convergence theorem allows us to conclude that for all k,

$$\lim_N \mathsf{E} \left[\left(\sum_e \int_0^T A_{k,e} \beta \Lambda_{i(e),t} \hat{z}_{e,t}^{\mathrm{D}} dt - \beta x_{0,k} \right)^+ \right] = 0. \tag{A.6}$$

Further, we must have that

$$\sum_e \int_0^T A_{k,e} \Lambda_{i(e),t} \hat{z}_{e,t}^{\mathrm{R}} dt \leq x_{0,k} + \max_{j,i} \frac{T}{N} (\Lambda_{i,jT/N}^{\max} - \Lambda_{i,jT/N}^{\min}) \overline{A}$$

by virtue of the dynamics of $\hat{x}_t^{\mathrm{R}}$ and the definition of $\hat{z}_{jT/N}^{\mathrm{R}}$ which guarantees that $A_{k,e} \hat{z}_{e,d(t)}^{\mathrm{R}} = 0$ if $\hat{x}_{k,d(t)}^{\mathrm{R}} = 0$. It follows that

$$\lim_N \left(\sum_e \int_0^T A_{k,e} (1-\beta) \Lambda_{i(e),t} \hat{z}_{e,t}^{\mathrm{R}} dt - (1-\beta) x_{0,k} \right)^+ = 0.$$

As in (A.6), the dominated convergence theorem applies to yield

$$\lim_N \mathsf{E} \left[\left(\sum_e \int_0^T A_{k,e} (1-\beta) \Lambda_{i(e),t} \hat{z}_{e,t}^{\mathrm{R}} dt - (1-\beta) x_{0,k} \right)^+ \right] = 0 \tag{A.7}$$

Taking an expectation followed by the limit infimum on both sides of (A.4) then yields by (A.6) and (A.7) the result.

A.7 Proofs for Section 1.4.3.

Proof of Lemma 1.4.4 Let us define the process $\tilde{\Lambda}_t$ according to $\tilde{\Lambda}_t = (\Lambda_t - \lambda_t)^+$ and observe that $\tilde{\Lambda}_t$ satisfies the conditions of Theorem 1.3.9. Now, we have:

$$\liminf_N \frac{\mathsf{E}\left[J^{\mathrm{R}-\beta}_{S,\{\Lambda_t\}}(x_0)\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\tilde{\Lambda}_t\}}(x_0)\right]} \geq \frac{\mathsf{E}\left[\liminf_N \frac{T}{N}\sum_{j=0}^{N-1} \frac{\mathrm{LP}\left(jT/N, \Lambda_{jT/N}, \hat{x}_0^{\mathrm{R}}(N-j)/N\right)}{T-jT/N}\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\tilde{\Lambda}_t\}}(x_0)\right]}$$
$$\geq \frac{\mathsf{E}\left[\liminf_N \frac{T}{N}\sum_{j=0}^{N-1} \frac{\mathrm{LP}\left(jT/N, \tilde{\Lambda}_{jT/N}, \hat{x}_0^{\mathrm{R}}(N-j)/N\right)}{T-jT/N}\right]}{\mathsf{E}\left[J^{\mathrm{UB}}_{\{\tilde{\Lambda}_t\}}(x_0)\right]}$$

where the first inequality is a consequence of Fatou's lemma and (1.6), and the second inequality follows from the monotonicity of $\mathrm{LP}(t, \Lambda, x)$ in Λ, and the fact that $\Lambda_t \geq \tilde{\Lambda}_t$. The remainder of the proof is then identical to Theorem 1.3.9.

Proof of Lemma 1.4.6 We have

$$\begin{aligned}
J^{\mathrm{UB}}_{\{\lambda_t\}}(x_0) + J^{\mathrm{UB}}_{\{(\overline{\Lambda}_t-\lambda_t)^+\}}(x_0) &= \mathrm{LP}\left(0, \frac{1}{T}\int_0^T \lambda_t dt, x_0\right) + \mathrm{LP}\left(0, \frac{1}{T}\int_0^T (\overline{\Lambda}_t - \lambda_t)^+ dt, x_0\right) \\
&\geq \mathrm{LP}\left(0, \frac{1}{T}\int_0^T \lambda_t + (\overline{\Lambda}_t - \lambda_t)^+ dt, x_0\right) \\
&\geq \mathrm{LP}\left(0, \frac{1}{T}\int_0^T \Lambda_t dt, x_0\right) \\
&= J^{\mathrm{UB}}_{\{\Lambda_t\}}(x_0)
\end{aligned}$$

The inequality follows from the concavity of $\mathrm{LP}(t, u, x)$ in u and since any concave function $h : \mathbb{R}_+{}^n \to \mathbb{R}$ with $h(0) = 0$ must be super-additive. The second inequality follows from the monotonicity of $\mathrm{LP}(t, u, x)$ in u and since $\lambda_t + (\overline{\Lambda}_t - \lambda_t)^+ \geq \Lambda_t$. Taking expectations in the inequality above yields the result.

A.8 Proofs for Section 3.2

Proof of Lemma 3.2.1. Define, for each impression type, the frequency count $C^T(x) = |\{X_t, t \in [T] \text{ s.t. } X_t = x\}|$, as well as the linear program

$$\begin{aligned} \mathrm{LP}(C^T) = \max \quad & \sum_a \sum_{x \in \mathcal{X}} r_a \mathbb{1}(x \in \mathcal{X}_a) z(x,a) C^T(x) \\ \text{subject to} \quad & \sum_{x \in \mathcal{X}} z(x,a) C^T(x) \leq TB_a \\ & \sum_a z(x,a) \leq 1 \\ & z \geq 0. \end{aligned}$$

Now consider an ϵ-optimal sequence of controls $o = \{o^1, \ldots, o^T\}$ for $\mathrm{OPT}(T, TB)$, which must exist for arbitrary $\epsilon > 0$ (Bertsekas and Shreve (2007)). Fix some realization ω of impressions arrivals, and let $\hat{z}$ be such that

$$\hat{z}(x,a) = \frac{\sum_t \mathbb{1}(X_t = x) o_a^t}{C^T(x)}.$$

Since o is a sequence of admissible controls, one can show by a simple induction argument that $\sum o_a^t \leq TB$ so $\hat{z}$ is feasible for $\mathrm{LP}_\mu(C^T)$. Taking expectations yields $\mathrm{OPT}(T, TB) - \epsilon \leq \mathsf{E}[\mathrm{LP}(C^T)]$. Since the choice of ϵ was arbitrary, $\mathrm{OPT}(T, TB) \leq \mathsf{E}[\mathrm{LP}(C^T)]$. The lemma follows from the fact that $\mathsf{E}[\mathrm{LP}(C^T)] \leq T\mathrm{LP}_\mu$ by Jensen's inequality.

A.9 Proofs for Section 3.2.1

We first prove the following lemma which bounds the revenue loss from using an optimal bid price control versus a primal optimum solution to the linear program given any distribution ν that satisfies a granularity condition. (We note that the granularity condition involves a term ρ which we have defined with respect to the true distribution μ; this will allow us to quantify the error from the bid price policy in terms of the magnitude of LP_{μ}.)

Lemma A.9.1 *For any ν that is ϵ-good,*

$$LP_{\nu} - Rev_{\nu}(\beta_{\nu}^{*}) \leq \epsilon LP_{\mu},$$

where $\beta_{\nu}^{*} \in \arg\min D\text{-}LP_{\nu}$.

Proof Through randomly perturbing the allocation rewards by by an arbitrarily small amount $\delta(x,a)$ before we solve the dual problem, we can guarantee that with probability, there can be at most m ties in the adjusted bids (this is a standard argument in the literature; for a more in-depth treatment of this issue, see Agrawal et al. (2014).) The loss from setting all allocations with ties to 0 is at most

$$\begin{aligned}
\mathrm{LP}_{\nu} - \mathrm{Rev}_{\nu}(z^{\beta_{\nu}}) &\leq m r_{\max} \max_{x\in\mathcal{X}} \nu(x) \\
&\leq \epsilon \mathrm{LP}_{\mu}.
\end{aligned}$$

Proof of Lemma 3.2.3. We make use in the proof of the following simple bound on the tails of a binomial random variable:

Fact 1 (Dudley (2002)) $\mathbb{P}\left[Bin(N,p) \geq k\right] \leq \left(\frac{Np}{k}\right)^{k} e^{k-Np}$ *if $k \geq Np$, where $Bin(N,p)$ is a binomial distribution with N trials and success probability p.*

Let us bound the probability that $\hat{\mu}$ is ϵ-bad.

$$\begin{aligned}
\mathbb{P}\left[||\hat{\mu}_N||_{\infty} \geq \rho\frac{\epsilon}{m}\right] &\leq \sum_{x\in\mathcal{X}} \mathbb{P}\left[\hat{\mu}_N(x) \geq \rho\frac{\epsilon}{m}\right] \\
&\leq n\mathbb{P}\left[\mathrm{Bin}\left(N, \rho\frac{\epsilon}{m^2}\right) \geq \rho\right] \\
&\leq n\left(\frac{e}{m}\right)^{N\rho\frac{\epsilon}{m}} e^{N\rho\frac{\epsilon}{m}\left(1-\frac{1}{m}\right)} \\
&\leq n\left(\frac{e}{m}\right)^{N\rho\frac{\epsilon}{m}}
\end{aligned}$$

where the first inequality is a union bound over $x \in \mathcal{X}$ and the third inequality uses Fact 1. In order to make this probability lower than some δ, one must hence choose

$$\begin{aligned} N &= \frac{4m}{\rho\epsilon} \frac{\log \frac{n}{\delta}}{\log m} \\ &= \frac{4}{\rho} \frac{m}{\epsilon} \frac{\log \frac{n}{\delta}}{\log m}. \end{aligned}$$

A.10 Lemmas for Section 3.4

Lemma A.10.1 *Consider a random variable $X_N = \frac{1}{N}\sum_{i=1}^{N} Y_i$, where the Y_i's are a sequence of i.i.d. Bernoulli random variables with $\mathsf{E}[Y_i] = \mu, \sigma(Y_i) = \sigma$, and a threshold $b \geq 0$. Then,*

1.
$$\left|\mathsf{E}\left[\min\{b, X_N\} - \min\{b, \mu\}\right]\right| \leq \frac{\sigma}{\sqrt{2\pi N}} + \frac{3(1-2\mu)}{N}$$

2. *In the special case that $b = \mu$,*
$$\left|\mathsf{E}\left[\min\{b, X_N\} - \min\{b, \mu\}\right]\right| \geq \frac{\sigma}{\sqrt{2\pi N}} - \frac{3(1-2\mu)}{N}$$

Proof We prove both of these results by approximating X_N with a Gaussian random variable for which the computation of the above error is easy. For part 1, by triangle inequality,

$$\begin{aligned} \left|\mathsf{E}\left[\min\{b, X_N\} - \min\{b, \mu\}\right]\right| &\leq \left|\mathsf{E}\left[\min\{b, Z\} - \min\{b, \mu\}\right]\right| \\ &\quad + \left|\mathsf{E}\left[\min\{X_N - \mu, b - \mu\} - \min\{Z - \mu, b - \mu\}\right]\right|, \end{aligned}$$

where Z is a Gaussian r.v. with mean and variance identical to X_N's. By Lemma A.10.2, we can bound the second term from above by $\frac{3(1-2\mu)}{N}$. We will now give precise bounds on the first term (where we have replaced X_N with Z), which will asymptotically dominate the second.

Case 1: $b \le \mu$. Then,

$$\begin{aligned}|\mathsf{E}[\min\{b, Z\} - \min\{b,\mu\}]| &= \left|\mathsf{E}\left[(Z-b)^-\right]\right| \\ &\le \left|\mathsf{E}\left[(Z-\mu)^-\right]\right| \\ &= \mathsf{E}\left[(Z-\mu)^+\right] \\ &= \frac{\sigma}{\sqrt{2\pi N}}\end{aligned}$$

where we have used the assumption that $b \le \mu$ in the first inequality, and the fact that, if if Z is a 0 mean Gaussian, $\mathsf{E}[Z^+] = \frac{\sigma(Z)}{\sqrt{2\pi}}$.

Case 2: $b > \mu$. For this case,

$$\begin{aligned}|\mathsf{E}[\min\{b, Z\} - \min\{b,\mu\}]| &= |\mathsf{E}[\min\{Z-\mu, b-\mu\}]| \\ &= \left|\frac{1}{2}\mathsf{E}[Z-\mu|Z-\mu \le 0] + \frac{1}{2}\mathsf{E}[\min\{Z-\mu, b-\mu\}\,|Z-\mu > 0]\right| \\ &\le \left|\frac{1}{2}\mathsf{E}[Z-\mu|Z-\mu \le 0]\right| \\ &= |\mathsf{E}[\min\{Z-\mu, 0\}]| \\ &= \frac{\sigma}{\sqrt{2\pi N}},\end{aligned}$$

where we have used the fact that in the first inequality, $\mathsf{E}[\min\{Z-\mu, b-\mu\}\,|Z-\mu > 0]$ is positive due to $b > \mu$ and less than $\mathsf{E}[Z-\mu|Z-\mu \le 0]$ by the symmetry of Z.

Proving part 2 of the lemma is similar, except that now we bound

$$\begin{aligned}|\mathsf{E}[\min\{b, X_N\} - \min\{b,\mu\}]| &= \left|\mathsf{E}\left[(X_N-\mu)^-\right]\right| \\ &\ge \left|\left|\mathsf{E}\left[(Z-\mu)^-\right]\right| - \left|\mathsf{E}\left[(X_N-\mu)^- - (Z-\mu)^-|\right]\right|\right| \\ &\ge \frac{\sigma}{\sqrt{2\pi N}} - \frac{3(1-2\mu)}{N}\end{aligned}$$

where we have used the reverse triangle inequality in the first inequality, and for the second, Lemma A.10.2.

Lemma A.10.2 *Consider a random variable* $X_N = \frac{1}{N}\sum_{i=1}^{N} Y_i$*, where the* Y_i*'s are a sequence of i.i.d. Bernoulli random variables with* $\mathsf{E}[Y_i] = \mu, \sigma(Y_i) = \sigma$*. Then there exists a Gaussian random variable* Z *such that* $\mathsf{E}[Z] = \mathsf{E}[X_N], \sigma(Z) = \sigma(X_N)$ *and*

$$|\mathsf{E}\left[\min\{X_N - \mu, b - \mu\} - \min\{Z - \mu, b - \mu\}\right]| = \frac{3(1-2\mu)}{N}.$$

Proof For notational convenience, let $f(x) = \min\{x - \mu, b - \mu\}$ and notice that f is a 1-Lipschitz function. Moreover, let $W_N = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}(Y_i - \mu)$ be the transformation of X_N into a random variable with standard normal mean and variance.

The expression in the statement of the lemma becomes

$$\begin{aligned}
|\mathsf{E}\left[f(X_N) - f(Z)\right]| &\leq \sup_{h:||h'||_\infty \leq 1} |\mathsf{E}\left[h(X_N) - h(Z)\right]| \\
&= d_W(X_N, Z) \\
&= \frac{\sigma}{\sqrt{N}} d_W(W_N, \frac{\sigma}{\sqrt{N}}(Z - \mu)) \\
&\leq \frac{\sigma}{\sqrt{N}} \frac{3\mathsf{E}[(|Y_i - \mu|/\sigma)^3]}{\sqrt{N}} \\
&= \frac{3(1-2\mu)}{N},
\end{aligned}$$

where we have used the Lipschitz property of f in the first inequality, used the definition of the Wasserstein metric in the first equality, applied Proposition A.10.3 for the second inequality, and used the fact that $\mathsf{E}[|Y_i - \mu|^3] = \mu(1-\mu)^3 - (1-\mu)\mu^3$ for the last equality.

Finally, we state the following finite sample Central Limit Theorem convergence result under the Wasserstein metric, which we used in the proof of the above lemma; the result is derived using Stein's method and a proof can be found in Chen et al. (2011).

Proposition A.10.3 *Consider a sequence* $X_1, \ldots, X_N$ *of independent random vari-*

ables with $\mathsf{E}[X_i] = 0, \mathsf{E}[X_i^2] = 1$ *and* $\mathsf{E}[|X_i|^3] < \infty$. *Then,*

$$d_W(W_N, Z) \leq \frac{3}{N^{3/2}} \sum_{i=1}^{N} \mathsf{E}[|X_i|^3],$$

where d_W *is the Wasserstein distance defined as*

$$d_W(U, V) = \sup_{h:||h'||_\infty \leq 1} \left| \mathsf{E}h(U) - \mathsf{E}h(V) \right|,$$

$W_N = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} X_i$ *and* Z *is a Gaussian random variable with mean and variance equal to* W_N*'s.*

A.11 Lemmas for Section 3.4

Lastly, we the Bounded Differences Inequality which we have used for Lemma 3.4.2 (a proof of this inequality can be found in Motwani and Raghavan (1995)):

Proposition A.11.1 *(Bounded Differences Inequality.) Suppose that* $f : \mathbb{R}^n \to \mathbb{R}$ *satisfies,* $\forall 1 \leq k \leq n$,

$$|f(x_1, \dots, x_i, \dots, x_n) - f(x_1, \dots, x_i', \dots, x_n)| \leq L.$$

Consider a vector $X = (X_1, \dots, X_n)$ *with independent components. Then,*

$$\mathbb{P}\left[|f(X) - \mathsf{E}[f(X)]| \geq t\right] \leq 2 \exp\left(-\frac{t^2}{2nL^2}\right).$$

A.12 Proofs for Section 3.5

Proof of Theorem 3.5.1. We first consider a policy which goes through the impression types in $\mathcal{X}$ in sequential order and assigns each x to the advertiser $a(x)$ (as defined in Generative Model 1). Define $N(a) \subseteq \mathcal{X}$ to be the set of impression types which this procedure allocates to advertiser a. Viewing the procedure above as assigning balls (impression types) to random bins (advertisers), it follows from Berenbrink et al. (2000) that with probability $1 - o(1)$,

$$N(a) \leq \frac{n}{m} + O\left(\sqrt{\frac{n \log m}{m}}\right), \ \forall a \in [m].$$

We show that this assignment of impressions to advertisers leads to a feasible allocation (which is a lower bound on the expected value of LP_μ) with expected value $\Theta(1)$.

Since in order to guarantee feasibility we have to truncate the allocation into an advertiser by that advertiser's budget, the expected size of the matching becomes

$$\begin{aligned}
&\mathsf{E}\left[\sum_a r_a \min\left\{B_a, \sum_{x\in N(a)} \mu(x)\right\}\right] \\
&= \sum_a r_a \mathsf{E}\left[\min\left\{B_a, \sum_{x\in N(a)} \mu(x)\right\}\right] \\
&\geq \sum_a r_a \mathsf{E}\left[\min\left\{B_a, \sum_{x\in N(a)} \mu(x)\right\} \mid B_a \geq \frac{1}{m}\right] \mathbb{P}\left[B_a \geq \frac{1}{m}\right] \\
&= \alpha \sum_a r_a \mathsf{E}\left[\min\left\{\frac{1}{m}, \sum_{x\in N(a)} \mu(x)\right\}\right],
\end{aligned}$$

where we have used property (2.b) of the generative model in the last line.

Furthermore, since $\sum_a N(a) = n$ and $\max_a N(a) \leq n/m + O(\sqrt{\frac{n \log m}{m}})$ with high probability, the following event has probability $o(1)$:

$$\mathcal{E} = \left\{ \nexists a \text{ s.t. } N(a) \leq \frac{n}{m} - O(m\sqrt{\frac{n \log m}{m}}) \right\}$$

In addition, let $\underline{N} = \frac{n}{m} - O(\sqrt{nm \log m})$. Hence

$$
\begin{aligned}
&\mathsf{E}\left[\sum_a r_a \min\left\{B_a, \sum_{x \in N(a)} \mu(x)\right\}\right] \\
&\geq \alpha \sum_a r_a \mathsf{E}\left[\min\left\{\frac{1}{m}, \sum_{x \in N(a)} \mu(x)\right\}\right] \\
&\geq \alpha \sum_a r_a \mathsf{E}\left[\min\left\{\frac{1}{m}, \sum_{x \in N(a)} \mu(x)\right\} | \mathcal{E}^C, \sum_{x \in N(a)} \mu(x) \geq \frac{|N(a)|}{n}\right]\cdot \\
&\quad \mathbb{P}\left[\mathcal{E}^C\right] \mathbb{P}\left[\sum_{x \in N(a)} \mu(x) \geq \frac{|N(a)|}{n}\right] \\
&\geq \alpha\beta(1 - o(1)) \sum_a r_a \mathsf{E}\left[\min\left\{\frac{1}{m}, \frac{1}{m} - \frac{O(\sqrt{nm \log m})}{n}\right\}\right] \\
&= O(r_{\text{avg}}),
\end{aligned}
$$

where for the second to last inequality we have used property$(3.b)$ of our generative model definition along with the high probability guarantee on the event $\mathcal{E}$, and for the last equality we have assumed the condition $\frac{m \log m}{n} = o(1)$ that our generative model assumes. Since our allocation policy was suboptimal, the theorem follows.

A.13 Proofs for Section 4.3.1

Proof of Lemma 4.3.3. Let us begin by writing down the complementary slackness conditions for the optimal u, v, z:

$$
\begin{aligned}
u\left(\sum_{e:i(e)=i} c_e z_e - B^i\right) &= 0 \\
v\left(\sum_{e:i(e)=i} z_e - 1\right) &= 0 \\
z_e\left(u c_e + v - p_e\right) &= 0, \forall e \,\text{subject to}\, i(e) = i.
\end{aligned}
$$

There are three cases:

1. $u = 0, v > 0$. Then, $\sum_{e:i(e)=i} z_e = 1$ and the optimal primal is found by allocating fully to the edge e with maximum price p_e. This follows from the feasibility condition $uc_e + v \geq p_e, \forall e$ subject to $i(e) = i$, so only the z_e with maximal price can be positive. Note that we are assuming here there is a single item with maximal price - this can achieved without loss of generality by an arbitrarily small random perturbation of the price vector.

2. $v = 0, u \geq 0$. Similarly to the first case, it follows from the complementary slackness conditions that it is optimal to fully allocate along the edge e with the maximal ratio p_e/c_e.

3. $u > 0, v > 0$. Thus implies that $\sum_{e:i(e)=i} z_e = 1$ and $\sum_{e:i(e)=i} c_e z_e = B^i$. By construction of the envelope, we cannot have more than two tight constraints $v + uc_e - p_e = 0$, so by the last complementary slackness condition only two z_e's can be positive. Therefore, we can completely identify the primal solution.

In either case, the solution can be determined using no more than $O(d)$ calculations.

Bibliography

Zoe Abrams, Ofer Mendelevitch, and John Tomlin. Optimal delivery of sponsored search advertisements subject to budget constraints. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 272–278, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-653-0. doi: 10.1145/1250910.1250950. URL `http://doi.acm.org/10.1145/1250910.1250950`.

D. Adelman. Dynamic bid price in revenue management. *Operations Research*, 55: 647–661, July-August 2007.

Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online Vertex-Weighted Bipartite Matching and Single-bid Budgeted Allocations. In *Proceedings of the 22th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 1253–1264, 2011.

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 2014.

M. Akan and B. Ata. Bid-price controls for network revenue management: Martingale characterization of optimal bid prices. *Mathematics of Operations Research*, 34(4): 912–936, 2009.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta algorithm and applications. Technical report, 2005.

Baruch Awerbuch and Rohit Khandekar. Stateless distributed gradient descent for positive linear programs. In *STOC*, pages 691–700, 2008.

M. O. Ball and M. Queyranne. Toward robust revenue management: Competitive analysis of online booking. *Operations Research*, 57(4):950–963, 2009.

Santiago Balseiro, Besbes Omar, and Gabriel Y. Weintraub. Repeated auctions with budgets in ad exchanges: Approximations and design. 2012. URL `http://dx.doi.org/10.2139/ssrn.2149319`.

M. Bayati, D. Shah, and M. Sharma. Max-product for maximum weight matching: Convergence, correctness, and lp duality. *IEEE Trans. Inf. Theor.*, 54(3):1241–1251, March 2008a. ISSN 0018-9448. doi: 10.1109/TIT.2007.915695. URL `http://dx.doi.org/10.1109/TIT.2007.915695`.

Mohsen Bayati, Christian Borgs, Jennifer Chayes, and Riccardo Zecchina. On the exactness of the cavity method for weighted b-matchings on arbitrary graphs and its relation to linear programs. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(06):L06001+, June 2008b. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/06/l06001. URL `http://dx.doi.org/10.1088/1742-5468/2008/06/l06001`.

A. Bemporad. Model-based predictive control design: New trends and tools. In *CDC06: 45th Conference on Decision and Control*, pages 6678–6683, San Diego, CA, 2006.

Petra Berenbrink, Artur Czumaj, Angelika Steger, and Berthold Vöcking. Balanced allocations: The heavily loaded case. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 745–754, New York, NY, USA, 2000. ACM. ISBN 1-58113-184-4. doi: 10.1145/335305.335411. URL `http://doi.acm.org/10.1145/335305.335411`.

D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case.* Athena Scientific, 2007.

D. Bertsimas and R. Demir. An approximate dynamic programming approach to multidimensional knapsack problems. *Management Science*, 48:550–565, April 2002.

O. Besbes and C. Maglaras. Revenue optimization for a make-to-order queue in an uncertain market environment. *Oper. Res.*, 57:1438–1450, November 2009.

M. Bramson. State space collapse with application to heavy traffic limits for multiclass queueing networks. *Queueing Systems*, 30:89–148, June 1998.

N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34:270–286, May 2009.

H. Chen and D. D. Yao. Dynamic scheduling of a multiclass fluid network. *Operations Research*, 41:1104–1115, November 1993.

Louis H.Y. Chen, Larry Goldstein, and Qi-Man Shao. *Normal Approximation by SteinÕs Method.* Springer-Verlag, 2011. ISBN 978-3-642-15007-4.

Y. Chen and V. F. Farias. Simple policies for dynamic pricing with imperfect forecasts. *Operations Research*, 61(3):612–624, 2013. URL `http://dblp.uni-trier.de/db/journals/ior/ior61.html#ChenF13`.

D.F. Ciocan and Vivek Farias. Model predictive control for dynamic resource allocation. mathematics of operations research. *Mathematics of Operations Research*, 37 (3):501–525, 2012.

Nikhil R. Devanur and Thomas P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC '09, pages 71–78, New York,

NY, USA, 2009. ACM. ISBN 978-1-60558-458-4. doi: 10.1145/1566374.1566384. URL http://doi.acm.org/10.1145/1566374.1566384.

R.M. Dudley. *Real Analysis and Probability*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2002. ISBN 9780521007542. URL http://books.google.com/books?id=7UuT7UZViN0C.

Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet Advertising and the Generalized Second Price Auction: Selling Billions of Dollars Worth of Keywords. NBER Working Papers 11765, National Bureau of Economic Research, Inc, November 2005. URL http://ideas.repec.org/p/nbr/nberwo/11765.html.

V. F. Farias and B. Van Roy. An approximate dynamic programming approach to network revenue management. 2007. Preprint.

V. F. Farias and B. Van Roy. Dynamic pricing with a prior on market response. *Oper. Res.*, 58:16–29, January 2010.

Jon Feldman, Nitish Korula, Vahab Mirrokni, S. Muthukrishnan, and Martin Pal. Online ad assignment with free disposal. In Stefano Leonardi, editor, *Internet and Network Economics*, volume 5929 of *Lecture Notes in Computer Science*, pages 374–385. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-10840-2. doi: 10.1007/978-3-642-10841-9_34. URL http://dx.doi.org/10.1007/978-3-642-10841-9_34.

Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *Proceedings of the 18th annual European conference on Algorithms: Part I*, ESA'10, pages 182–194, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15774-2, 978-3-642-15774-5. URL http://dl.acm.org/citation.cfm?id=1888935.1888957.

G. Gallego and G. van Ryzin. A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, 45(1):pp. 24–41, 1997.

David Gamarnik, Devavrat Shah, and Yehua Wei. Belief propagation for min-cost network flow: Convergence and correctness. *Oper. Res.*, 60(2):410–428, March 2012. ISSN 0030-364X. doi: 10.1287/opre.1110.1025. URL http://dx.doi.org/10.1287/opre.1110.1025.

Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM J. Comput.*, 37(2):630–652, May 2007. ISSN 0097-5397. doi: 10.1137/S0097539704446232. URL http://dx.doi.org/10.1137/S0097539704446232.

Ashish Goel, Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Advertisement allocation for generalized second pricing schemes, 2010.

Sariel Har-Peled. *Geometric Approximation Algorithms.* American Mathematical Society, Boston, MA, USA, 2011. ISBN 0821849115, 9780821849118.

Krishnamurthy Iyer, Ramesh Johari, and Mukund Sundararajan. Mean field equilibria of dynamic auctions with learning. 2012. URL `http://dx.doi.org/10.2139/ssrn.1799085`.

S. Jasin and S. Kumar. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. 37(2):313–345, 2012.

I. Karatzas and S.E. Shreve. *Brownian motion and stochastic calculus.* Springer, 1991.

R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 630–631, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

Vijay Krishna. *Auction Theory.* Academic Press, 2002.

Michael Luby and Noam Nisan. A parallel approximation algorithm for positive linear programming. pages 448–457, 1993.

C. Maglaras and J. Meissner. Dynamic pricing strategies for multiproduct revenue management problems. *Manufacturing Service Operations Management*, 8(2):136–148, 2006.

M. Mahdian, H. Nazerzadeh, and A. Saberi. Online optimization with uncertain information, 2009. Preprint.

Faraz Makari Manshadi, Baruch Awerbuch, Rainer Gemula, Rohit Khandekar, Julián Mestre, and Mauro Sozio. A distributed algorithm for large-scale generalized matching. *PVLDB*, 6(9):613–624, 2013.

A. Mehta, A. Saberi, U Vazirani, and V. Vazirani. Adwords and generalized on-line matching. In *FOCS Š05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 264–273. IEEE Computer Society, 2005.

Marco Molinaro and R. Ravi. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *ICALP (1)*, Lecture Notes in Computer Science, pages 701–713. Springer, 2012. ISBN 978-3-642-31593-0.

Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms.* Cambridge University Press, New York, NY, USA, 1995. ISBN 0-521-47465-5, 9780521474658.

Owen O'Malley. Terabyte sort on apache hadoop, May 2008.

Serge A. Plotkin, David B. Shmoys, and Eva Tardos. Fast approximation algorithms for fractional packing and covering problems. 1991.

M. I. Reiman and Q. Wang. An asymptotically optimal policy for a quantity-based network revenue management problem. *Mathematics of Operations Research*, 33 (2):257–282, 2008.

Sujay Sanghavi, Devavrat Shah, and Alan S. Willsky. Message passing for maximum weight independent set. *IEEE Transactions on Information Theory*, 55(11):4822–4834, 2009. URL `http://dblp.uni-trier.de/db/journals/tit/tit55.html#SanghaviSW09`.

D. Shah and D. Wischik. Switched networks with maximum weight policies: Fluid approximation and multiplicative state space collapse. *CoRR*, abs/1004.1995, 2010. URL `http://dblp.uni-trier.de/db/journals/corr/corr1004.html#abs-1004-1995`.

Kalyan Talluri and Garrett Van Ryzin. An analysis of bid-price controls for network revenue management. *Manage. Sci.*, 44(11):1577–1593, November 1998. ISSN 0025-1909. doi: 10.1287/mnsc.44.11.1577. URL `http://dx.doi.org/10.1287/mnsc.44.11.1577`.

Hal R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163 – 1178, 2007. ISSN 0167-7187. doi: http://dx.doi.org/10.1016/j.ijindorg.2006.10.002. URL `http://www.sciencedirect.com/science/article/pii/S0167718706001317`.

Erik Vee, Sergei Vassilvitskii, and Jayavel Shanmugasundaram. Optimal online assignment with forecasts. In *ACM EC*, 2010.

D. Zhang and D. Adelman. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science*, 43: 381–394, August 2009.