Learning While Bidding in Real Time Auctions with Multiple Item Types and Unknown Price Distribution

by

Amirhossein Boreiri

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A Real-Time Bidding (RTB) network is a real-time auction market, primarily used for advertising space sales. Within this environment, clients participate by bidding on preferred items and subsequently purchasing them upon winning. This thesis addresses the problem of optimal real-time bidding within a second-price Vickrey auction setting, where the distribution of prices is unknown. Our focus centers on second-price auction mechanisms, which offers unique properties that enable the development of compelling algorithms. We introduce the concept of a demand-side platform (DSP), acting as an intermediary representing clients in the auction market. With no prior knowledge of typical prices, the DSP must determine optimal bidding strategies for each item and distribute won items among clients to fulfill their contracts while minimizing expenses. When the distribution of the prices of items is known, this optimal bidding problem can be solved by classic convex optimization algorithms such as ADMM. However, market properties may vary over time, and access to competitor behavior or bidding information is limited. Consequently, the DSP must continually update its information about the price distribution, while adapting bidding estimations in real-time. Our primary contribution lies in devising efficient online optimization algorithms that accurately find the optimal bids. To tackle this, we employ tools from convex optimization analysis, including duality, along with stochastic optimization algorithms, notably stochastic approximation. Moreover, techniques such as projection and penalty term methods are utilized to enhance algorithm performance.

Acknowledgements

I am deeply grateful to Professor Ravi Mazumdar for his guidance, support, and inspiration throughout my academic journey at the University of Waterloo. His mentorship has been invaluable, and I am thankful for his dedication to my success.
I would like to extend my appreciation to Professor Patrick Mitran, Professor Andrew Heunis, Professor Liang-Liang Xie, and Dr Ryan kinnear for their valuable contributions to this thesis. Their insights and feedback have greatly enriched my research.
I am indebted to my fellow students Kasra Fallah, Samira Ghanbarian, and Parsa Rangriz for their collaboration, suggestions, and support throughout my research endeavors. Their input has been instrumental in shaping my work. A special acknowledgment goes to my dear friend Miswar Syed for his unwavering support.

Dedication

I dedicate this thesis to my beloved parents and my dear sister for their unconditional love, encouragement, and unwavering support. Their presence in my life has been my source of strength and motivation.

Table of Contents

List of Figures

Chapter 1

Introduction

The present work is motivated by problems arising within a particular advertising mechanism known as real-time bidding (RTB) [13, 40]. RTB is the mechanism for placing ads on web pages visited by users, with the current objective being to target ads based on user profiles, leveraging the growing availability of user information. The defining feature of RTB is that advertisers can bid on every individual *impression* (*i.e.,* the right for their chosen message to be displayed to a desirable user, also referred to generically in this thesis as an *item*), as opposed to fixed advertising contracts specifiying a-priori a service-level agreement.

In this thesis, we focus on real-time, sealed-bid, and second-price auctions, due to their ubiquity in the context of ad placement on web pages and wireless phones. These are referred to as *Vickrey* auctions [36]). The auctions are facilitated by *ad exchanges* (*e.g.,*Google AdX [30], OpenX, Index Exchange, etc.) that provide the technology to match buyers to sellers and implement the auction. The winner of the auction (that is always the highest bidder) for a particular item gains the right to have their content displayed to the user, and pays the amount corresponding to the second highest bid. Given the scale of the industry and the rapidity of these auctions, the development of computational bidding algorithms is essential.

The complexity of the bidding process has led to the emergence of consolidators or intermediaries who act on behalf of advertisers. These intermediaries are called *Demand Side Platforms* or DSPs. The economic benefits these intermediaries can provide to advertisers are numerous: the advertiser can offload the risk of adverse price movements, relieve themselves of the need to maintain their own complex bidding infrastructure, make more certain up-front spending estimates, potentially access a wider array of advertising channels, etc. These motivations and the contract design problem are studied in [3].

The problem we address in this thesis uses the framework of [26] which considers how these intermediaries can manage a collection of contracts demanding that the DSP acquire items having certain characteristics at a specified rate. The goal of the intermediary is to fulfill these contracts at minimum cost, hence maximizing their profit. In [26] a theoretical framework is presented where the market price characteristics are known to the bidder. In reality such characteristics are unknown and it can be assumed that the so-called supply characteristics of ad types (ads that target specific profiles) are drawn from an unknown distribution. In this thesis we develop algorithms that do not make use of any prior estimates of typical prices, and require only the censored price observations (prices that are only revealed when the bids are successful) obtained through the process of bidding. That is, the DSP can *bid, learn, and optimize* simultaneously.

1.1 Overview and Contributions

The formulation of our main problem is similar to [26], but considers the setting where there is no prior information about prevailing market prices. The goal of the DSP is to acquire items having some specified average value to the advertiser. This is as opposed to planning, using prior information, for fulfilling a contract over some fixed time horizon. In practice, these approaches are complementary: the optimal item acquisition rates estimated by a long-term plan with prior information can be used as set-points for an adaptive algorithm. We assume that the type space for items is finite and is in fact without loss of generality valid for second price auctions. A type is an attribute associated with contracts and might be common to many contracts. An example of a type could be the age profile that an ad might target, the gender, location, etc. Typical contracts might specify many targeting criteria or types. Thus a given item might satisfy the criteria of many contracts.

The main tool used in this thesis is stochastic approximation (SA) [33, 28, 5, 37] , a formalism for analyzing adaptive algorithms operating in a stochastic environment when the underlying statistics are unknown. First, we develop a quadratic penalty stochastic approximation method on two time-scales (see *e.g.,*[34, 38]): A *fast time-scale* is used to calculate what bid should be placed in order to win items at a specified rate , and a *slow time-scale* which determines the actual rates at which items should be allocated to contracts (Section 2.2.2). The fast time-scale of our SA solves a problem similar to that of [23, 43, 22], where some given set-point needs to be maintained. The slow time-scale of our algorithm globally coordinates the bidding process in a way that is roughly analogous to the *E-step* in [44]. Then, drawing on duality theory [10] and employing convex optimization methods [39, 27, 9] like Subgradient Descent and Projected Gradient Descent, we refine our approach by formulating an enhanced convex problem. This reformulation enables us to directly updating of optimal bids, ensuring an effective optimization process for real-time bidding in auction settings.

We note that the key fact regarding the correspondence between optimal bids and objective function derivatives only holds for second price auctions – this suggests a relevant direction for future research: and a question remains as to whether (or to what extent) the methods developed here can be extended to more general auction mechanisms. In [26] it is shown that under further assumptions, like log-concavity, both the first and second price cases correspond to convex optimization problems that are different. In this thesis we focus on the second price case, that is more common and does not require log-concavity.

1.2 Outline

We begin by defining a market model and contract management for the Real-Time auctions setup in Section 2.1.1, where we introduce the concept of a supply curve in Definition 2.1.1. This curve describes the probability of winning an item of a particular type. The contract management problem is then formulated in Definition 2.1.2, where we describe the Real-Time Bidding (RTB) problem as a convex optimization problem with contract constraints represented as equality conditions.

In Section 2.2, we derive a Two Timescale Stochastic Approximation algorithm capable of solving the contract management problem without prior knowledge of the supply curve. This algorithm is divided into two parts. First, in Section 2.2.1, we present a stochastic approximation algorithm for estimating the bids required to win and acquire a specified amount of item supply rate. Then, in Section 2.2.2, we focus on designing another stochastic approximation algorithm to solve the convex optimization problem, and find the optimal bids, which the demand side platform (DSP) need to bid to fulfill the contracts. Finally, we combine these two components into a Two Timescale SA algorithm. This algorithm operates entirely online, allowing the Demand Side Platform (DSP) to bid, learn, and optimize simultaneously.

In Chapter 3, we analyze the duality of the convex optimization model of the contract management problem. Utilizing properties of duality [10], we construct a new convex problem in Section 3.1 and make observations to aid in developing an algorithm to solve the dual problem. Sections 3.2, 3.3, and 3.4 detail the development of three separate algorithms to solve the dual problem. In each section, we prove the convergence of these algorithms and validate their performance through simulations.

Finally, we conclude in Chapter 4. The theorems utilized throughout the thesis to prove convergence are provided in the Appendices.

Chapter 2

Problem Definition and Learning Acquisition rates

In this chapter, we begin by reviewing the earlier work [26, 25, 24], where they addressed the modeling and formulation of Real-Time Bidding problems. In Section 2.1, we present a model for the market, item arrival dynamics, and price distribution, formulating Real-Time Bidding as a convex optimization problem.

However, the classic convex optimization algorithms are inadequate when the price distribution is unknown. Therefore, we employ an improved stochastic approximation algorithm to update the bids while learning the supply rates. In Section 2.2, we introduce our first algorithm aimed at addressing this challenge.

2.1 Market Modeling and Contract Management

2.1.1 Market Pricing Distribution

In earlier work [26], the Real-Time Bidding process was formulated as an Online Convex Optimization problem where the cost was a function of the probability of winning the items, instead of the amount of bid itself. This formulation required a precise model of how individual prices are distributed and how items arrive in the market.

First, let's model the arrival of items. Suppose there is an auction market with M item types denoted as $[M] \triangleq \{1, \dots, M\}$, where items of random types arrive according to a random marked point process with a rate of $\lambda > 0$. The inter-arrival times, denoted by $\tau_1, \tau_2, \ldots$, are independent and identically distributed with mean $1/\lambda$ and a finite variance. The marks correspond to the types $\phi_n \in [M]$ of the n^{th} arriving item are drawn independently from a distribution $\mathbb{P}\{\phi_n = j\} = \eta_j > 0$, and thus the rate of arrivals of items of type $j \in [M]$ is itself a random process of rate $\lambda_j = \eta_j \lambda$.

Now, let's model the price distribution: The *price* (*i.e.,* highest competing bid) of the n^{th} arriving item is distributed, conditional on $\phi_n = j$, according to the cumulative distribution function W_j on $\mathbb{R}$, assumed to have density $W_j'(x)$. That is, $(p_n \mid \phi_n = j) \sim W_j$. The probability of winning an item of type j with a bid of x is given by $W_j(x) = \mathbb{P}\{p_n \leq x \mid \phi_n = j\}$ and thus the rate of items of type j won at auction by an exogenous agent placing a constant bid $x \geq 0$ is $\lambda_j W_j(x)$.

The function

$$f_j(x) \triangleq \mathbb{E}[p_n \mathbb{1}(p_n \leq x) \mid \phi_n = j] = \int_0^x u W_j'(u) \mathsf{d}u \tag{2.1}$$

is the *cost curve* and measures the expected cost of bidding x on an item of type j. The function

$$\Lambda_j(q) \triangleq f_j \circ W_j^{-1}(q) \overset{(a)}{=} \int_0^q W_j^{-1}(u) \mathsf{d}u \tag{2.2}$$

is the *acquisition cost curve* and measures the expected cost of bidding to win an item of type j with probability $q \in [0,1]$. The equality (a) can be established through an elementary change of variables. The derivative of Λ_j is W_j^{-1} on $(0,1)$, is monotone increasing, and hence the function Λ_j is convex on $[0,1]$. When extended to all of $\mathbb{R}$ to satisfy $\Lambda_j(s) = 0 \ \forall s \leq 0$ and $\Lambda_j(s) = \infty \ \forall s > \lambda_j$ it is also lower semicontinuous. The implications of these properties are developed in detail by [26], but the details pertinant to this thesis are summarized in the following Proposition 2.1.1. Before presenting this proposition, we first formalize important assumptions and conventions for the function W in the following definition.

Definition 2.1.1 (Supply Curve)**.** *For some fixed item type $j \in [M]$, omitted from the notation, we refer to the function $x \mapsto W(x)$ as the supply curve for items of type j. This supply curve is assumed to be continuous on $\mathbb{R}$ and differentiable on the interval $(0, \bar{x}_j)$, where $\bar{x}_j$ is a finite value representing the maximum bid that any individual is willing to pay, therefore $W(x) = 1$ for $x > \bar{x}_j$. The derivative of the supply curve, $W'(x)$, is uniformly bounded and satisfies $0 < \epsilon \leq W'(x) \leq \frac{1}{\epsilon} < \infty$, for all $x \in (0, \bar{x}_j)$ and for some $\epsilon > 0$. Thus, $W(x)$ is strictly monotone and Lipschitz on the interval $(0, \bar{x}_j)$. Therefore, due to the strict monotonicity, the inverse of W is defined in the usual manner within the interval $(0, \bar{x}_j)$. It is convenient to extend this function to all of $\mathbb{R}$ as follows:*

$$W^{-1}(q) = \begin{cases} 0 & \textit{if } q \leq 0 \\ x \textit{ such that } W(x) = q & \textit{if } 0 < q < 1 \\ \bar{x} & \textit{if } q \geq 1 \end{cases} \tag{2.3}$$

Proposition 2.1.1 (Convex Acquisition Costs [26]). *Let $W(x)$ be a supply curve. Then, in a second price auction, the acquisition cost function $\Lambda_{2nd}(q) = f_{2nd} \circ W^{-1}(q)$ is given by $\int_0^q W^{-1}(u)\mathsf{d}u$ on $q \in [0,1]$. If this is extended to:*

$$\widetilde{\Lambda}_{2nd}(q) \triangleq \begin{cases} 0 & \textit{if } q \leq 0 \\ \int_0^q W^{-1}(u)\mathsf{d}u & \textit{if } 0 < q < 1, \\ f(\bar{x}) + (q-1)\bar{x} & \textit{if } q \geq 1 \end{cases} \tag{2.4}$$

then $\widetilde{\Lambda}_{2nd}$ is a proper, lower semi-continuous, non-decreasing, and convex function on $\mathbb{R}$. Moreover, $\widetilde{\Lambda}_{2nd}$ is strictly convex over $[0,1]$, differentiable, and the derivative of it is the extended $W^{-1}(q)$.

Other than the fact that $\widetilde{\Lambda}_j$ is a convex function, the key observation to make from Proposition 2.1.1 is that the derivative satisfies $\widetilde{\Lambda}'_j(q) = W^{-1}(q)$. The implication of this relation is that the bid $x_j = W_j^{-1}(q)$ required to win an item of type j is exactly equal to the derivative of the function $\widetilde{\Lambda}_j$. This is the key fact that we exploit in this thesis, and is used in Section 2.2.

2.1.2 Formulating the Contract Management Problem

In the context of real-time bidding, we will define a *contract* $i \in [N]$ as a tuple $\left(C_i, (v_{ij})_{j\in[M]}\right)$ where $v_{ij} \geq 0$ represents the value of an item of type j for contract i, and $C_i > 0$ is a target *rate* at which item value should be acquired. The values v_{ij} can be interpreted as conversion probabilities if they are constrained to $v_{ij} \in [0,1]$ (though this constraint is not essential), in which case the interpretation of the contract is that conversions should be acquired at the rate C_i.

Precisely, the *contract management problem* is a convex optimization problem that models the goal of a DSP to fulfill N contractual obligations at minimum cost:

$$
\begin{aligned}
\underset{R}{\text{minimize}} \quad & \sum_{j=1}^{M} \lambda_j \Lambda_j \left(\frac{1}{\lambda_j} \sum_{i=1}^{N} R_{ij}\right) \\
\text{subject to} \quad & \sum_{j=1}^{M} v_{ij} R_{ij} = C_i \\
& R_{ij} \geq 0.
\end{aligned} \tag{P}
$$

The variable R_{ij} indicates the number of items of type j that should be acquired and allocated towards contract i per unit time. Given an optimal matrix R, $\frac{1}{\lambda_j} \sum_{i=1}^{N} R_{ij}$ represents the portion of type j items that need to be acquired, which can be interpreted as the probability of winning item j. Consequently, the optimum bids x_j to place on items of type j can be calculated by $x_j = W_j^{-1}\big(\frac{1}{\lambda_j} \sum_{i=1}^{N} R_{ij}\big)$.

Therefore, the average cost of acquiring item j is $\lambda_j \Lambda_j (\frac{1}{\lambda_j} \sum_{i=1}^{N} R_{ij})$, and by minimizing the sum of all these average costs, we can determine the optimum cost. A detailed analysis of this problem, including a thorough analysis of duality and the associated bidding interpretations, is available in [26]. It is shown there that the following assumption is essential:

Assumption 2.1.1 (Adequate Supply)**.** *We suppose that there exists a strictly feasible point $R \in \mathbb{R}^{N \times M}$ for Problem (P), satisfying $R > 0$, $\sum_{j=1}^{M} v_{ij} R_{ij} = C_i$ and $\sum_{i=1}^{N} R_{ij} < \lambda_j$.*

Under Assumption 3.1.1, solutions to Problem (P) exist and are such that the optimum bids are finite: $x_j < \infty$.

Problem (P) can be slightly simplified. Indeed, the array v_{ij} can be expected to be *sparse* (many contracts may have zero valuation for certain item types). This motivates the definition for the set of *usable items* and the set of *fulfillable contracts*

$$
\begin{aligned}
\mathcal{A}_i &\triangleq \{j \in [M] \mid v_{ij} > 0\}, \\
\mathcal{B}_j &\triangleq \{i \in [N] \mid v_{ij} > 0\},
\end{aligned} \tag{2.5}
$$

respectively. Clearly, optimal solutions will have $R_{ij} = 0$ whenever $j \notin \mathcal{A}_i$ so that the actual dimensionality of $R \in \mathbb{R}^{N \times M}$ is given only by $d = \sum_{i=1}^{N} |\mathcal{A}_i| = \sum_{j=1}^{M} |\mathcal{B}_j| \geq N$, rather than MN, and we would generally expect $d \ll MN$.

2.2 Optimizing Bids and Acquisition Rates in the Primal Problem

To address problem P, we adopt an algorithm inspired by [39], which explores an algorithm for finding approximate solutions to a convex optimization problem with inequality conditions. In our approach, we augment the objective function with quadratic penalty terms to penalize violations of the inequality constraints.

Specifically, we consider the objective function with a penalty parameter $\mu > 0$:

$$\mathcal{L}_\mu(R) = \sum_{j=1}^{M} \lambda_j \widetilde{\Lambda}_j\big(\frac{1}{\lambda_j}\sum_{i\in\mathcal{B}_j} R_{ij}\big) + \frac{1}{2}\mu \sum_{j=1}^{M}\big[\sum_{i\in\mathcal{B}_j}(R_{ij})_-^2 + \big(\sum_{i\in\mathcal{B}_j} R_{ij} - \lambda_j\big)_+^2\big], \tag{2.6}$$

where the penalty terms are added to ensure that R_{ij} remains positive, and $\sum_{i\in\mathcal{B}_j} R_{ij}$ is less than λ_j.

In addition to this term, it can be expected that Problem (P) admits multiple solutions. Indeed, it can be the case that there are many ways of fulfilling contracts with the same item acquisition rates. For example, if contracts i and i' have the same valuations for item types j and j', the proportions of those types of items allocated to those contracts makes no difference for the cost function. To ensure a unique solution, we adopt the "least norm" approach, where we select the solution $R^\star$ such that, out of all other solutions, $||R^\star||_2$ is minimized.

Therefore, the final cost function will be:

$$\begin{aligned}\mathcal{L}_\mu(R) = &\sum_{j=1}^{M} \lambda_j \widetilde{\Lambda}_j\big(\frac{1}{\lambda_j}\sum_{i\in\mathcal{B}_j} R_{ij}\big) + \frac{1}{2\mu}||R||_2^2 \\ &+ \frac{1}{2}\mu \sum_{j=1}^{M}\big[\sum_{i\in\mathcal{B}_j}(R_{ij})_-^2 + \big(\sum_{i\in\mathcal{B}_j} R_{ij} - \lambda_j\big)_+^2\big],\end{aligned} \tag{2.7}$$

In order to approximate solutions of Problem (P), consider the following simplified optimization problem

$$\begin{aligned}&\underset{R}{\text{minimize}} && \mathcal{L}_\mu(R) \\ &\text{subject to} && \sum_{j\in\mathcal{A}_i} v_{ij} R_{ij} = C_i.\end{aligned} \tag{P^μ}$$

Minimizers of Problem (P^μ), which will be denoted R^μ, act as approximate solutions of Problem (P). moreover, $R^\mu \to R^\star$ as $\mu \to \infty$, where $R^\star$ is the least norm solution of Problem (P). In some instances it can hold that $R^\mu = R^\star$ for each $\mu \geq \mu^\star$ for some $\mu^\star \geq 0$, see [17].

2.2.1 Updating the Bids

To solve P when Λ_j are unknown, we will develop an algorithm that updates the bids at each step using an approximation of the cost function, which itself will be updated while bidding.

We develop an algorithm to calculate the optimal array R that solves Problem (P^μ), and hence it enables us to determine the required supply rates $s_j \triangleq \sum_{i \in \mathcal{B}_j} R_{ij}$, which feedback into the first algorithm for bid calculation.

The bid x_j will be calculated on a *fast timescale* and the estimation of R on a *slow timescale*; the meaning of these terms is to be clarified. The key to this algorithm is that the aforementioned bid x_j is exactly equal to the derivative of the extended acquisition function $\widetilde{\Lambda}_j$ at s_j/λ_j.

Throughout this section we assume that we have in hand an array R, which optimizes Problem (P^μ). Consequently, we can determine $s_j = \sum_{i \in \mathcal{B}_j} R_{ij}$, representing the total optimal supply rate for item of type j. Given the supply rates, it is still necessary to compute the appropriate bids x_j such that $\lambda_j W_j(x_j) = s_j$. In principle, this bid can be calculated given knowledge of W_j, but this function is unknown apriori. To address this problem we use a stochastic approximation algorithm which, for a particular target supply s_j, attempts to find the corresponding bid value.

Since the problem of calculating the bid x_j given a desired supply s_j is separable across j, let us fix some item type $j \in [M]$ that we focus on. Throughout this section we will consider only this single item type and therefore omit the j subscript from the notation: W instead of W_j, λ instead of λ_j, *etc.* Recalling the notation from Section 2.1, items of this type will be supposed to arrive according to a random process at the rate $\lambda > 0$, and we will denote the independent and identically distributed inter-arrival times τ_n. Additionally, each item is marked with independent prices $p_n \sim W$. It will often be convenient to write $\bar{\tau} = \mathbb{E}\tau_n = 1/\lambda$, which is the mean inter-arrival time. The c.d.f. W is not assumed to be known. Instead, we rely on the only feedback we get from the market regarding the outcome of the last bid, specifically whether the bid was successful in winning the auction or not (an item is won if $p_{n+1} \leq x(n)$). We represent this information using the function

$\mathbb{1}(p_{n+1} \leq x(n))$. Thus, the algorithm is robust to the censoring of prices wherein only the winning bidder learns what the item sold for.
The following algorithm

$$x(n+1) = x(n) + a_n\big[s\tau_{n+1} - \mathbb{1}(p_{n+1} \leq x(n))\big],$$

with some arbitrary initial point $x(0) = x_0$, and a_n as a non-negative step size, is known as a *stochastic approximation* (SA), see [5]. By re-writing this recursion as

$$x(n+1) = x(n) + a_n\big[s\bar{\tau} - W(x(n))\big] + a_n\big[\big(s\tau_{n+1} - s\bar{\tau} + W(x(n)) - \mathbb{1}(p_{n+1} \leq x(n))\big)\big],$$

Now, as long as $s \in (0, \lambda)$, it can be shown that the ODE $\dot{x} = s\bar{\tau} - W(x)$ has a unique globally asymptotically stable equilibrium $W^{-1}(s/\lambda)$, and therefore $x(n)$ can be expected to converge to this point. In fact, we can establish the convergence of a slightly more sophisticated algorithm which simultaneously approximates $\bar{\tau}$(the interarrival time of the items) as well. We will again use an approximation parameter $\mu > 0$. We consider the *Bid Adaptation* iterations:

$$\begin{aligned} \hat{\tau}(n+1) &= \hat{\tau}(n) + a_n\big[\tau_{n+1} - \hat{\tau}(n)\big], \\ x(n+1) &= x(n) + a_n\Big[s\hat{\tau}(n) - \mathbb{1}(p_{n+1} \leq x(n)) - \frac{1}{\mu}[(x(n) - \bar{x})_+ - (x(n))_-]\Big], \end{aligned} \tag{BA}$$

where the penalty terms $(x)_-$ and $(x(n) - \bar{x})_+$ are used to keep the iterates within the range $[0, \bar{x}]$(interval $[0, \bar{x}]$ is the compact support of the distribution $W(x)$). As μ approaches infinity, the equilibrium of this ordinary differential equation (ODE) will tend to $W^{-1}(s/\lambda)$.

We establish the convergence of this algorithm in the following proposition.

Proposition 2.2.1 (Bid Adaptation)**.** *Suppose that τ_n, p_n are drawn according to the market model described in Section 2.1.1 and for a fixed type $j \in [M]$ (omitted from the notation) with (differentiable) supply curve W. If a_n satisfies the Robbins-Monro conditions $\sum_{n=1}^{\infty} a_n = \infty$ and $\sum_{n=1}^{\infty} a_n^2 < \infty$ and $\mu > 0$, $s \in \mathbb{R}$ are fixed, then the iterations described in bid adaptation Equation (BA), converges almost surely:*

$$(\hat{\tau}(n), x(n)) \overset{a.s.}{\to} (\bar{\tau}, x^*) \text{ as } n \to \infty, \tag{2.8}$$

where $W_\mu(x) \triangleq W(x) + \frac{1}{\mu}[(x_n - \bar{x})_+ - (x_n)_-]$ is strictly increasing on $\mathbb{R}$. Additionally, $W_\mu(x^*) = s/\lambda$, and $\bar{\tau} = 1/\lambda$.

Proof. Consider the stochastic approximation (BA), which can be rewritten in the canonical form

$$\hat{\tau}(n+1) = \hat{\tau}(n) + a_n\big[\bar{\tau} - \hat{\tau}(n) + \big(\tau_{n+1} - \bar{\tau}\big)\big], \tag{2.9}$$

$$x(n+1) = x(n) + a_n\Big[\Big(W\big(x(n)\big) - \mathbb{1}(p_{n+1} \le x(n))\Big) + \Big(s\hat{\tau}(n) - W\big(x(n)\big) - \frac{1}{\mu}[(x(n)-\bar{x})_+ - (x(n))_-]\Big)\Big] \tag{2.10}$$

$$= x(n) + a_n\Big[\big(s\hat{\tau}(n) - W_\mu\big(x(n)\big)\big)\Big] + a_n\Big[\Big(W\big(x(n)\big) - \mathbb{1}(p_{n+1} \le x(n))\Big)\Big]. \tag{2.11}$$

In more abstract terms, the recursions can be written as

$$z(n+1) = z(n) + a_n[h(z(n)) + M_{n+1}],$$

where $z_n = (\hat{\tau}(n), x(n))$, h is the function

$$h(\tau, x) = \big(\bar{\tau} - \tau, s\tau - W_\mu\big(x\big)\big)$$

which summarizes the dynamics, and

$$M_{n+1} = \Big(\tau_{n+1} - \bar{\tau}, W\big(x(n)\big) - \mathbb{1}\big(p_{n+1} \le x(n)\big)\Big)$$

is the noise term.
In order to establish convergence, we need to verify the assumptions required by Theorem A.1.1. In particular, we show that h is Lipschitz, M_n is a uniformly square integrable martingale difference sequence, and a Lyapunov function exists. Furthermore, due to strict monotonicity of $W_\mu(x)$, the equation $W_\mu(x) = s\bar{\tau}$ has a unique solution, denoted as x^*. Additionally, since for all $x \in [0, \bar{x}]$, $W'_\mu(x) = W'(x) < 1/\epsilon$ (see Definition 2.1.1), and for all x outside $[0, \bar{x}]$, $W'_\mu(x) = 1/\mu$; we can conclude that $W'_\mu(x) \le \min(1/\epsilon, 1/\mu) = c_1$, establishing the Lipschitz property of $W_\mu(x)$. Similar reasoning shows that $W'_\mu(x) \ge \max(\epsilon, 1/\mu) = c_2$ .In addition, the linearity of $s(\tau - \bar{\tau})$ implies that the functions $\big(\bar{\tau} - \tau, s\tau - s\bar{\tau}\big)$ and $\big(\bar{\tau} - \tau, W_\mu\big(x^*\big) - W_\mu\big(x\big)\big)$ are both Lipschitz, and their sum remains Lipschitz as well. In addition, $\mathbb{E}[M_{n+1}\,|\mathcal{F}_n] = 0$ $\big($since $\mathbb{E}[\tau_{n+1}\,|\mathcal{F}_n] = \bar{\tau}$ and $\mathbb{E}[\mathbb{1}(p_{n+1} \le x(n))\,|\mathcal{F}_n] = W(x_n)\big)$, and $\mathbb{E}[||M_{n+1}||^2\,|\mathcal{F}_n] = var(\tau) + W(x(n))(1 - W(x(n))) \le var(\tau) + 1 = \sigma^2 < \infty$. We have demonstrated that equation (BA) satisfies first four conditions in Theorem A.1.1. Now, to establish the stability, it is necessary to find a Lyapunov function that satisfies the properties mentioned in 5^{th} condition of Theorem A.1.1.
let us define the vector $\theta = (\tau - \bar{\tau}, kx - kx^*)$, and $\theta^* = (0, 0)$, where k is a constant . Consequently, $h(\tau, x) = 0$ when $\theta = \theta^*$. We can express the derivative of θ as,

$$\dot{\theta} = (\dot{\tau}, k\dot{x}) = (\bar{\tau} - \tau, ks\tau - kW_\mu(x)).$$

In addition, we define the Lyapunov function as $V(\theta) = \frac{||\theta-\theta^*||^2}{2}$. Thus, we have:

$$\dot{V}(\theta) = \langle \theta - \theta^*, \dot{\theta} \rangle = \Big\langle \Big(\tau - \bar{\tau}, kx - kx^*\Big), \Big(\bar{\tau} - \tau, ks(\tau - \bar{\tau}) - k\big(W_\mu(x) - W_\mu(x^*)\big)\Big) \Big\rangle$$

By simplifying the terms and using the fact that $|W_\mu(x) - W_\mu(x^*)| \geq c_2|x - x^*|$, we optain

$$\begin{aligned} \dot{V} &\leq -\big[(\tau - \bar{\tau})^2 - ks(\tau - \bar{\tau})(kx - kx^*) + c_2(kx - kx^*)^2\big] \\ &\leq -\langle(\theta - \theta^*), P(\theta - \theta^*)\rangle, \end{aligned}$$

where $P = \begin{bmatrix} 1 & -\frac{1}{2}ks \\ -\frac{1}{2}ks & c_2 \end{bmatrix}$. And by selecting $0 < k < \sqrt{\frac{4c_2}{s^2}}$ one can demonstrate that $P \succ 0$, and therefore $\dot{V}(\theta) \leq -\sqrt{\lambda_{min}(P)}\ ||\theta - \theta^*||^2 = -b||\theta - \theta^*||^2$.
Finally, we showed that equation (BA) satisfies all the conditions in Theorem A.1.1 therefore, (τ, x) will converges almost surely.

□

2.2.2 Solving the Primal Problem

The previous section describes a method which, given a desired target supply s_j, estimates the bid $x_j = W_j^{-1}(s_j/\lambda_j)$ which attains s_j supply. Using this information, we want to estimate the solution $R \in \mathbb{R}^{N\times M}$ of Problem (P), and hence derive the supply requirement $s_j = \sum_{i\in\mathcal{B}_j} R_{ij}$ which feeds into the stochastic approximation of Section 2.2. Recall from Proposition 2.1.1 that the derivative of $\widetilde{\Lambda}_j$, for $q \in \mathbb{R}$, is given by $W_j^{-1}(q)$ for the extended inverse function of Equation (2.3), which is equal to x_j, the bid required to win items of type j with probability q. It is this property that we exploit in deriving a stochastic approximation for the solution of Problem (P): since it is necessary to estimate the bid x_j that attains the supply rate s_j, the derivative of the objective function comes to us *for free*. Motivated by this derivative property, we will analyze a first order projected stochastic gradient algorithm for solving Problem (P^μ), and thus obtaining approximate solutions to the contract management problem (P). The derivatives of the objective $\mathcal{L}_\mu(R)$ are given by

$$\frac{\partial \mathcal{L}_\mu}{\partial R_{ij}}(R) = W_j^{-1}\big(\frac{1}{\lambda_j}\sum_{i\in\mathcal{B}_j} R_{ij}\big) + \frac{1}{\mu}R_{ij} + \mu\big[(R_{ij})_- + \big(\sum_{i\in\mathcal{B}_j} R_{ij} - \lambda_j\big)_+\big],\ ^{1} \tag{2.12}$$

[1] $(\sum_{i\in\mathcal{B}_j} R_{ij} - \lambda_j)_+ := max(0, \sum_{i\in\mathcal{B}_j} R_{ij} - \lambda_j)$

if $j \in \mathcal{A}_i$ and 0 otherwise. And, projection $\Pi_{\mathcal{S}_C}(R)$ of R onto the convex set

$$\mathcal{S}_C = \{R \subseteq \mathbb{R}^{M \times N} \mid \sum_{j \in \mathcal{A}_i} v_{ij} R_{ij} = C_i\}$$

is given by the Affine mapping

$$[\Pi_{\mathcal{S}_C}(R)]_{ij} = R_{ij} - \big(\sum_{j \in \mathcal{A}_i} v_{ij}\big)^{-1} \big[\sum_{j \in \mathcal{A}_i} v_{ij} R_{ij} - C_i\big],$$

which can be written as $\Pi_{\mathcal{S}_C}(R)_{ij} = \Pi_{\mathcal{S}_0}(R)_{ij} + U_{ij}$. where $U_{ij} = \big(\sum_{j \in \mathcal{A}_i} v_{ij}\big)^{-1} C_i$, and $\Pi_{\mathcal{S}_0}(R)$ is a linear mapping over the subspace $\mathcal{S}_0 = \{R \subseteq \mathbb{R}^{M \times N} \mid \sum_{j \in \mathcal{A}_i} v_{ij} R_{ij} = 0\}$, i.e,

$$[\Pi_{\mathcal{S}_0}(R)]_{ij} = R_{ij} - \big(\sum_{j \in \mathcal{A}_i} v_{ij}\big)^{-1} \big[\sum_{j \in \mathcal{A}_i} v_{ij} R_{ij}\big].$$

Recall that we have an *i.i.d.* sequence (θ_n, p_n) of (type, price) pairs modelling the items arriving at auction. Specifically, we have $\mathbb{P}\{\theta_n = j\} = \eta_j > 0$ and $p_n \mid \theta_n \sim W_{\theta_n}$ (see Section 2.1.1). The stochastic approximation estimating the bid $x_j(n)$ for items of type j will be updated upon each arrival of an item of type j; in this sense the algorithm is *asynchronous* – only a single component is updated at once, and there are time delays between updates. To model this, we use the indicator function $\mathbb{1}_j(\theta)$ which takes the value 1 if $\theta = j$ and 0 otherwise.

Using this notation, and combining the derivatives in Equation (3.5) with the bid adaptation algorithm of Equation (BA), a complete *asynchronous* and *two timescale* stochastic approximation algorithm for learning solutions to Problem (P) is specified by Algorithm 1.

The first two steps of Algorithm 1 are used to provide adaptive estimates of the gradient of each $\widetilde{\Lambda}_j$, as well as $\hat{\tau}_j$ used in penalty term $(s_j(n) - \frac{1}{\hat{\tau}_j(n)})_+$, to constraint the supply s_j. The remaining lines carry out a projected stochastic gradient to estimate R^μ.

Proposition 2.2.2 (Primal Algorithm Convergence)**.** *Suppose there is adequate supply (Assumption 3.1.1) for Problem (P). As well, suppose that both a_n, b_n satisfy the Robbins-Monro conditions and $b_n/a_n \to 0$ as $n \to \infty$. Then, for $\mu > 0$, the iterates $R(n) \to R^\mu$ converge almost surely to the solution of Problem (P^μ) as $n \to \infty$.*

Proof. First, we establish that the iterates for R_{ij} follows a stochastic approximation:

$$\begin{aligned} R_{n+1} &= \Pi_{\mathcal{S}_C}(R_n - b_n \nabla \mathcal{L}_\mu(R_n)) \\ &= \Pi_{\mathcal{S}_0}(R_n - b_n \nabla \mathcal{L}_\mu(R_n)) + U \\ &= \Pi_{\mathcal{S}_C}(R_n) - b_n \Pi_{\mathcal{S}_0}(\nabla \mathcal{L}_\mu(R_n)) \\ &= R_n - b_n \Pi_{\mathcal{S}_0}(\nabla \mathcal{L}_\mu(R_n)). \end{aligned}$$

Algorithm 1: Online Bidding

for *Each Arriving Item* $\tau_{n+1}, \theta_{n+1}, p_{n+1}$ **do**

Update interarrival time estimates:

$\hat{\tau}_j(n+1) = \hat{\tau}_j(n) + a_n \mathbb{1}_j(\theta_{n+1})\big[\tau^j_{n+1} - \hat{\tau}_j(n)\big]$

$\hat{\tau}(n+1) = \hat{\tau}(n) + a_n\big[\tau_{n+1} - \hat{\tau}(n)\big]$

Place bid $x_j(n)$ observe $\mathbb{1}(p_{n+1} \leq x_j(n))$ and update bid:

$x_j(n+1) = x_j(n) + a_n \mathbb{1}_j(\theta_{n+1})\Delta x_j(n),$

$\Delta x_j(n) = s_j(n)\hat{\tau}_j(n) - \mathbb{1}(p_{n+1} \leq x_j(n)) - \frac{1}{\mu}[(x_j(n) - \bar{x}_j)_+ - (x_j(n))_-]$

Take a gradient step for $\mathcal{L}_\alpha(R)$:

$\tilde{R}_{ij}(n+1) = R_{ij}(n) - b_n \mathbb{1}_j(\theta_{n+1})\Delta R_{ij}(n),$

$\Delta R_{ij}(n) = x_j(n) + \frac{1}{\mu}R_{ij}(n) + \mu\big[(R_{ij}(n))_- + \big(s_j(n) - \frac{1}{\hat{\tau}_j(n)}\big)_+\big]$

Linear projection of $\tilde{R}(n+1)$ on $\mathcal{S}_C$:

$R(n+1) = \Pi_{\mathcal{S}_C}(\tilde{R}_{ij}(n+1))$

Keep track of total supply targets:

$s_j(n+1) = \sum_{i \in \mathcal{B}_j} R_{ij}(n+1)$

We used the fact that $R_n \in \mathcal{S}_C$. Consequently, the iterations involving the vectors $(x, \hat{\tau})$ and R are equivalent to a two timescale asynchronous Stochastic Approximation. As a result, in order to establish convergence, we must verify the conditions outlined in Theorem A.1.4, as indicated by Proposition A.1.1(which deals with the asynchronous nature of Algorithm 1).

By Proposition 2.2.1, for fixed $s_j \in \mathbb{R}^M$, the iterates $\hat{\tau}_j(n), x_j(n)$ converge almost surely to $1/\lambda_j$, and $W_j^{-1}(s_j/\lambda_j)$ respectively, which are globally asymptotically stable equilibrium of the associated ODEs; notice that the separability of these equations means that the $\mathbb{1}_j(\theta_{n+1})$ terms can only impact the rate of convergence, and not the asymptotic value.

Hence, we have successfully established the stability of $\hat{\tau}_j(n)$ and $x_j(n)$ for a fixed s_j which fulfils the first condition in Theorem A.1.4. Next, it is necessary to verify the stability of of $R(n)$ and confirming the satisfaction of the second condition. For this purpose, we observe that the iterates of $R_{ij}(n)$ approximate the ordinary differential equation (ODE),

$$\dot{R} = -\Pi_{\mathcal{S}_0} \nabla \mathcal{L}_\mu(R),$$

which is obtained by substituting the asymptotic values of $\hat{\tau}_j, x_j$ into the limiting ODE for the iterates R_{ij}.

Furthermore, $\mathcal{L}_\mu(R)$ is convex, proper, lower semi-continuous, and coercive. Additionally, the set $\mathcal{S}_C$ is a convex, nonempty, and closed set, and $\mathcal{S}_C \cap \mathrm{dom}\mathcal{L}_\mu(R) \neq \varnothing$ (from assumption 3.1.1). Consequently, $\mathcal{L}_\mu|_{V_C}$ has a minimizer, denoted R^μ. Moreover, due to the strong convexity of $\mathcal{L}_\mu(R)$ (which results from the quadratic penalty $\frac{1}{\mu}||R||_F^2$), R^μ is unique. Furthermore, minimizing $\mathcal{L}_\mu(R)$ under the constraint $R \in \mathcal{S}_C$ is equivalent to minimizing $\mathcal{L}_\mu(R) + i_{\mathcal{S}_C}(R)$. By applying the Fermat rule, it can be shown that $0 \in \partial(\mathcal{L}_\mu(R^\mu) + i_{\mathcal{S}_C})$ which implies that $-\nabla\mathcal{L}_\mu(R^\mu) \in N_{\mathcal{S}_C}(R^\mu)$. Therefore, we can conclude $\Pi_{\mathcal{S}_C}(R^\mu - \nabla(\mathcal{L}_\mu(R^\mu)) = R^\mu$ which implies that $\Pi_{\mathcal{S}_0}(\nabla\mathcal{L}_\mu(R^\mu)) = 0$. Hence R^μ satisfies (I) $\Pi_{\mathcal{S}_0}(\nabla\mathcal{L}_\mu(R^\mu)) = 0$, and (II) $\Pi_{\mathcal{S}_C}(R^\mu) = R^\mu$.
Moreover $\nabla\mathcal{L}_\mu$ is L-Lipschitz, and due to the linearity of $\Pi_{\mathcal{S}_0}$, so is $\Pi_{\mathcal{S}_0}(\nabla\mathcal{L}_\mu)$.

Finally, we will show that the solution R^μ of Problem (P^μ) is a unique globally asymptotically stable equilibrium for this ODE. By applying Theorem A.1.1, our objective is to show that $R(n) \overset{a.s.}{\to} R^\mu$. To this end, we apply Lyapunov's direct method. Let

$V(R) = \frac{1}{2}||R - R^\mu||_F^2$, which is a coercive function. We have

$$\dot{V}(R) = \langle R - R^\mu, -\Pi_{\mathcal{S}_0} \nabla \mathcal{L}_\mu(R) \rangle \tag{2.13}$$

$$\stackrel{(a)}{=} \langle R - R^\mu, \Pi_{\mathcal{S}_0} \nabla \mathcal{L}_\mu(R^\mu) - \Pi_{\mathcal{S}_0} \nabla \mathcal{L}_\mu(R) \rangle \tag{2.14}$$

$$\stackrel{(b)}{=} -\langle R - R^\mu, \nabla \mathcal{L}_\mu(R) - \nabla \mathcal{L}_\mu(R^\mu) \rangle \tag{2.15}$$

$$\stackrel{(c)}{\leq} -\frac{1}{\mu}||R^\mu - R||_F^2, \tag{2.16}$$

where (a) follows by the fact that $\Pi_{\mathcal{S}_0} \nabla \mathcal{L}_\mu(R^\mu) = 0$, (b) follows by the orthogonality principle for subspace projections ($R - R^\mu \in \mathcal{S}_0$, and $\langle s, h - \Pi_{\mathcal{S}_0}(h) \rangle = 0$ or $\langle s, \Pi_{\mathcal{S}_0}(h) \rangle = \langle s, h \rangle$ for any $s \in \mathcal{S}_0$), and (c) follows by strong convexity. Therefore by Theorem A.1.4, R will converge almost surely to R^μ.
Also $\dot{V}(R) \leq -\frac{1}{\mu} V(R)$ gives $V(t) \leq V_0 e^{-\frac{t_n}{\mu}}$. Therefore the rate of convergence is $||R^\mu - R|| \sim O(e^{-\sum b_n/2\mu})$. And when $b_n = 1/n$ we have convergence is $||R^\mu - R|| \sim O(\frac{1}{n^{1/2\mu}})$ □

We observed that $||R^\mu - R(n)||$ decreases at a rate of $\sim O(\frac{1}{n^{1/2\mu}})$. Moreover, the approximation error between R^μ (the optimal solution of problem (P^μ)) and R^* (the optimal solution of problem P) is $||R^\mu - R^*|| \sim O(\frac{1}{\mu})$, , a relationship proven in the work of [21].

Hence, we have $||R^* - R(n)|| \sim O(\frac{1}{n^{1/2\mu}}) + O(\frac{1}{\mu})$. And in the case that μ is scaled with n through the function $\mu(n) = (1 + \delta)^n$, we have $||R^* - R(n)|| \sim O(\frac{1}{n^\epsilon})$.

Chapter 3

Dual Optimization

Previously, we discussed the primal problem of optimal Real Time Bidding :

$$\begin{aligned} \underset{R}{\text{minimize}} \quad & \sum_{j=1}^{M} \lambda_j \Lambda_j (\frac{S_j}{\lambda_j}) \\ \text{subject to} \quad & \sum_{j=1}^{M} v_{ij} R_{ij} = C_i \\ & \sum_{i=1}^{N} R_{ij} = S_j, R_{ij} \geq 0. \end{aligned} \tag{P}$$

To find the optimal value in the case where no prior information about the bidding Density Functions is available, we utilized the Two Timescale Asynchronous Stochastic Approximation method. Furthermore, the optimization variable, allocation rates R_{ij} (determining the distribution of each winning item to each contract), enabled us to ascertain the supply rates s_j and the optimal bids x_j for each item.

This algorithm demonstrated that supply rates was updated at a slower time scale compared to allocation rates. This slower update is advantageous when the demand side platform (DSP) prioritizes contract aspect. However, it becomes a drawback when the focus shifts towards the market. Thus, this chapter proposes a new algorithm specifically focus on updating the bids and supply rates.

In this chapter, we tackle the dual problem of the convex optimization problem. We begin this chapter in Section 3.1, where we formulate the dual problem, followed by a analysis of duality properties. Moreover, leveraging the principle of strong convexity [10, 7], we will demonstrate that both the primal and dual problems converge to the same optimal solution for the Real-Time Bidding (RTB). Furthermore, we will present several observations critical for the algorithms designed to solve the dual problem.

While various convex optimization algorithms exist for minimizing a convex function subject to linear equality and inequality constraints, our problem presents unique challenges. We lack access to the cost function due to the absence of prior information on the distributions of item prices $W_j(x_j)$ and arrival rates λ_j. Consequently, we cannot directly utilize functions such as Λ_j and Λ_j^*. However, we do have access to noisy derivatives of the cost function, derived from previous bids and the outcomes of won and lost bids. These limitations restrict the algorithmic options available to us.

In the following sections, we will solve the dual problem using various algorithms. Specifically, we will explore the Penalty Term Method for Inequality Constraints in Section 3.2, followed by the Projected Noisy Gradient Descent method in Section 3.3, and finally, the Subgradient Method in Section 3.4.

For each of these algorithms, we will first outline the algorithmic steps, followed by a mathematical proof demonstrating its convergence. Subsequently, we will conduct simulations to compare the convergence rates of these algorithms in Section 3.4.3. Finally, we will compare the primal and dual problems.

3.1 Dual Analysis

In this section, we analyze the dual of the problem (P). We start by constructing the Lagrangian function:

$$\begin{aligned} L(S, R, \mu, \rho, \theta) &= \sum_{j=1}^{M} \lambda_j \Lambda_j(\frac{S_j}{\lambda_j}) + \sum_{i=1}^{N} \rho_i (C_i - \sum_{j=1}^{M} R_{ij} v_{ij}) + \sum_{j=1}^{M} \mu_j (\sum_{i=1}^{N} R_{ij} - S_j) - \sum_{i=1}^{N} \sum_{j=1}^{M} \theta_{ij} R_{ij} \\ &= \sum_{i=1}^{N} \rho_i C_i + \sum_{j=1}^{M} (\lambda_j \Lambda_j (S_j/\lambda_j) - \mu_j S_j) + \sum_{i=1}^{N} \sum_{j=1}^{M} R_{ij} (\mu_j - \theta_{ij} - v_{ij}) \end{aligned} \tag{3.1}$$

Here, the variables $\rho \in \mathbb{R}^N$ and $\mu \in \mathbb{R}^M$ are associated with the equality constraints concerning contract rates and supply rates, respectively. Additionally, matrix θ is associated with the non-negativity constraints for the acquisition rates. Consequently, the dual problem is subject to the constraint $\theta \geq 0$ and, arises from simplifying the max-min expression as follows:

$$\underset{\rho,\mu,\theta\geq 0}{\text{maximize}} \quad \inf_{s,R} \quad L(s, R, \mu, \rho, \theta). \tag{3.2}$$

To this end, we first minimize the Lagrangian function L with respect to s. To minimize L over s, we only need to minimize the term $\Lambda_j(\frac{S_j}{\lambda_j}) - \mu_j \frac{S_j}{\lambda_j}$ over S_j, which gives the Fencgel conjucate of Λ_j. Moreover, due to convexity of Λ_j, we have $\mu_j = \Lambda_j^{'}(s_j^*/\lambda_j) = W_j^{-1}(s_j^*/\lambda_j)$,where s^* is the optimal value of s.
Therefore we have,

$$L(s^*, R, \mu, \rho, \theta) = \sum_{i=1}^{N} \rho_i C_i - \sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j) + \sum_{i=1}^{N} \sum_{j=1}^{M} R_{ij}(\mu_j - \theta_{ij} - v_{ij}\rho_i). \tag{3.3}$$

Now we minimize L over R. In order to prevent the infimum from being $-\infty$, we have $\mu_j - \theta_{ij}^* - v_{ij}\rho_i = 0$, for all $i \in [N]$ and $j \in [M]$. This condition combined with dual non-negativity constraint $\theta > 0$ gives that for all $i \in [N]$ and $j \in [M]$ we have $\mu_j \geq v_{ij}\rho_i$. Consequently, we can eliminate the variable θ from the dual problem.
Therefore, we arrive at the following formulation for the dual problem:

$$\begin{aligned} \underset{\rho,\mu}{\text{minimize}} \quad & \sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j) - \sum_{i=1}^{N} \rho_i C_i \\ \text{subject to} \quad & v_{ij}\rho_i \leq \mu_j \quad \forall i \in N, j \in M \end{aligned} \tag{D}$$

Problem D is the dual to problem P. So their respective values, D^* and P^* have the relation $D^* \leq P^*$.

Before analyzing problem P, we introduce an assumption:

Assumption 3.1.1 (Adequate Supply). *We suppose that there exists a strictly feasible point $R \in \mathbb{R}^{N\times M}$ for Problem (P), satisfying $R > 0, \sum_{j=1}^{M} v_{ij}R_{ij} = C_i$ and $\sum_{i=1}^{N} R_{ij} < \lambda_j$.*

Under Assumption 3.1.1, solutions to Problem (P) exist and are such that the optimum bids are finite,i.e. $x_j < \infty$.

In the following proposition, we demonstrate that under Assumption 3.1.1, Problem (D) is guaranteed to have a unique solution as well.

Proposition 3.1.1 (Existence of dual solution)**.** *There exists a unique solution, ρ^* and μ^* to problem D, and the duality gap between the two problem D and problem P is zero.*

Proof. We begin by establishing the existence of an optimal solution to Problem (P). From our previous analysis, we know that the cost function $\sum_{j=1}^{M} \lambda_j \Lambda_j(\frac{S_j}{\lambda_j})$ is proper, convex, coercive, and lower semicontinuous.

Additionally, the feasible set $V = \{s, R \mid \sum_{j=1}^{M} v_{ij} R_{ij} = C_i, \sum_{i=1}^{N} R_{ij} = S_j, R_{ij} \geq 0\}$ is a closed, convex, nonempty set.

Now if we show that $dom\, L(s, R) \cap V \neq \emptyset$ then by Key Minimum Existence Theorem [10], there exist an optimal solution R^* to problem P.

To show this we have $dom\, L(s, R) = \{s \,|\, 0 \leq S_j \leq \lambda_j\}$, and therefore $dom\, L(s, R) \cap V = \{s, R \mid \sum_{j=1}^{M} v_{ij} R_{ij} = C_i, \sum_{i=1}^{N} R_{ij} \leq \lambda_j, R_{ij} \geq 0\}$, which by assumption 3.1.1 is non empty.

Furthermore, due to the strong convexity of $\sum_{j=1}^{M} \lambda_j \Lambda_j(\frac{S_j}{\lambda_j})$ with respect to s, the S_j values are unique (note that R_{ij} may not be unique).

Utilizing the strong duality theorem and the existence of a Slater point, we conclude that ρ^* and μ^* exist, and the duality gap is zero.

Finally, by minimizing the Lagrangian function with respect to s, we find that $\mu_j^* = W_j^{-1}(s_j^*/\lambda_j)$.

Since the function W_j^{-1} is monotone, and we already show that s_j^* are unique, then μ_j^* is unique.

Moreover, as we will demonstrate in the next proposition that $\rho_i^* = \min_j(\frac{\mu_j^*}{v_{ij}})$, leading to the uniqueness of ρ_i^* as well.

This proposition establishes the existence and uniqueness of the solution to Problem (D) under the assumption of adequate supply. □

Now, we proceed to make some observations on the dual problem that will be utilized further in this chapter.

Proposition 3.1.2 (Observations)**.** *Suppose Assumption 3.1.1 holds. Let R^* and s^* be optimal solutions of problem (P), and μ^*, ρ^*, θ^* are the optimal values for (D). Finally, let x_j be the optimal bid to acquire item of type j with optimal supply rate of S_j^* (i.e. $x_j = W_j^{-1}(S_j^*/\lambda_j)$). Then:*

1. *there exist unique solution for problem (D), μ^*, ρ^*.*

2. *the duality gap is zero.*

3. *(optimal bids):* μ_j^* *is equal to the optimal bid* x_j *which is equal to* $W_j^{-1}(S_j/\lambda_j)$.

4. *if* $\theta_{ij} > 0$ *then* $R_{ij} = 0$ *and if* $R_{ij} > 0$ *then* $\theta_{ij} = 0$.

5. $\forall j \in [M]$ *there exists* $i \in [N]$ *s.t* $\rho_i^* = \frac{\mu_j^*}{v_{ij}}$, *or in another word,* $\mu_j^* = \max_{i \in \mathcal{B}_j}(v_{ij}\rho_i^*)$.

6. $\forall i \in [N]$ *there exists* $j \in [M]$ *s.t* $\mu_j^* = v_{ij}\rho_i^*$ *, or in another word,* $\rho_i^* = \min_{j \in \mathcal{A}_i}(\frac{\mu_j^*}{v_{ij}})$.

7. *(positive values):* $\forall i, j,\ \rho_i^*, \mu_j^* > 0$

Proof. We have already established the proofs for the first three observations.
To prove observation 4, we used the complementary slackness [10, 7] and we have $\theta_{ij}R_{ij} = 0$.
Observation 6 stems from the monotonicity of $\sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j)$. When minimizing the cost function with fixed ρ_i values, μ_j must be minimized until, for at least one j, $\mu_j = v_{ij}\rho_i^*$.
Similarly, Observation 5 is derived from the monotonicity of $-\sum_{i=1}^{N} \rho_i C_i$.
Finally, the domain of $\Lambda_j^*(\mu_j)$ consists of positive μ_j values; otherwise, the function becomes infinite. Therefore, $\mu_j^* \geq 0$, and observation 6 further gives $\rho_j^* \geq 0$.

□

3.2 Penalty Term Method for Inequality Constraints

In this section, motivated by [39, 1],we will analyze the first algorithm, the Penalty Term Method. The concept involves changing the cost function by adding a quadratic form of the constraint functions, $g_i(x)_+$[1], as a penalty, thereby to minimize the new cost function the optimal point need to be as close as possible to the feasible points.

Consequently, the new cost function is defined as:

$$f_\alpha(x) = f(x) + \frac{\alpha}{2} \sum_{i=1}^{m} (g_i(x))_+^2$$

Now, with $g_i(x)$ representing convex functions and $g_i(x) \leq 0$ denoting the inequality constraints, it's important to note that the new optimization problem remains convex. This is because $(g_i(x))_+^2$ are also convex functions.

[1] $(g_i(x))_+ := max(0, g_i(x))$

Since there are no constraints in this new optimization problem, it can be efficiently solved using a stochastic approximation algorithm.

Furthermore, as α tends to infinity, the augmented cost function $f_\alpha(x)$ behaves as follows:

$$\lim_{\alpha\to\infty} f_\alpha(x) = \begin{cases} f(x), & x \text{ feasible} \\ +\infty, & \text{otherwise} \end{cases};$$

Here, feasible points are defined as the points x where $g_i(x) \le 0$ for all $i \in 1, ..., m$.

Therefore, by increasing α to infinity, we can enforce the constraints and reach the same solution as the dual problem. This proposition will be further elaborated and proven in Proposition 3.2.1.

3.2.1 Algorithm and convergence

For solving the dual problem D, to construct the new cost function, we incorporate penalty terms for the inequality conditions $g_{ij}(\mu, \rho) = v_{ij}\rho_i - \mu_j \le 0$. Additionally, we introduce regularization to the cost function by including the norm of the variables. This regularization term, $\frac{||\mu||^2+||\rho||^2}{2\alpha}$, enhances the speed of the algorithm by transforming the cost function into a strongly convex function. Finally, as α tends to infinity, this regularization term becomes negligible, ensuring that its inclusion does not impact the convergence point.

Consequently, the updated cost function is expressed as follows:

$$\mathcal{L}_\alpha(\mu, \rho) = \sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j) - \sum_{i=1}^{N} \rho_i C_i + \frac{\alpha}{2} \sum_{j=1}^{M} \sum_{i=1}^{N} (v_{ij}\rho_i - \mu_j)_+^2 + \sum_{j=1}^{M} \frac{\mu_j^2}{2\alpha} + \sum_{i=1}^{N} \frac{\rho_i^2}{2\alpha}, \tag{3.4}$$

Moreover the objective derivatives of this cost function will be:

$$\begin{aligned} \frac{\partial \mathcal{L}_\alpha}{\partial \mu_j}(\mu, \rho) &= \lambda_j W_j(\mu_j) - \alpha \sum_{i=1}^{N} (v_{ij}\rho_i - \mu_j)_+ + \frac{\mu_j}{\alpha} \\ \frac{\partial \mathcal{L}_\alpha}{\partial \rho_i}(\mu, \rho) &= -C_i + \alpha \sum_{j=1}^{M} v_{ij}(v_{ij}\rho_i - \mu_j)_+ + \frac{\rho_i}{\alpha} \end{aligned} \tag{3.5}$$

Now, we make use of the fact that $W_j(\mu_j) = \mathbb{E}[\mathbb{1}[p_{n+1} \le \mu_j(n)]]$. Due to the lack of access to $W_j(\mu_j)$, we replace it with its noisy version $\mathbb{1}[p_{n+1} \le \mu_j(n)]$. With access to these derivatives, we can now outline the stochastic approximation update steps:

$$\rho_i(n+1) = \rho_i(n) - a_n\big[\alpha \sum_{j=1}^{M} v_{ij}(v_{ij}\rho_i - \mu_j)_+ - C_i + \frac{\rho_i(n)}{\alpha}\big]$$
$$\mu_j(n+1) = \mu_j(n) - a_n\big[\lambda_j \mathbb{1}[p_{n+1} \le \mu_j(n)] - \alpha \sum_{i=1}^{N}(v_{ij}\rho_i - \mu_j)_+ + \frac{\mu_j(n)}{\alpha}\big] \qquad (PT/SA)$$

In the following proposition we will prove that these steps will converge to the optimal solution of problem D.

Proposition 3.2.1. *[Penalty method convergence]*
The update steps for $\mu(n), \rho(n)$ in the penalty term algorithm converge to a unique solution μ^α, ρ^α that minimizes $L_\alpha(\mu, \rho)$.

This solution can also be found by solving the following equations:

$$\frac{\partial \mathcal{L}_\alpha}{\partial \mu_j}(\mu^\alpha, \rho^\alpha) = \lambda_j W_j\big(\mu_j^\alpha\big) - \alpha \sum_{i=1}^{N}(v_{ij}\rho_i^\alpha - \mu_j^\alpha)_+ + \frac{\mu_j^\alpha}{\alpha} = 0$$
$$\frac{\partial \mathcal{L}_\alpha}{\partial \rho_i}(\mu^\alpha, \rho^\alpha) = -C_i + \alpha \sum_{j=1}^{M} v_{ij}(v_{ij}\rho_i^\alpha - \mu_j^\alpha)_+ + \frac{\rho_i^\alpha}{\alpha} = 0 \qquad (3.6)$$

Furthermore, μ^α, ρ^α will converge to μ^*, ρ^* as α tends to infinity, where μ^*, ρ^* represent the solution to problem D.

Proof. We first address the latter part of the proposition. $L_\alpha(\mu, \rho)$ is a convex, coercive, and proper function. These properties guarantee the existence of a unique minimum. Moreover, Given the differentiability of $L_\alpha(\mu, \rho)$, we have $\frac{\partial \mathcal{L}_\alpha}{\partial \mu} = \frac{\partial \mathcal{L}_\alpha}{\partial \rho} = 0$ at the minimum point.

As α becomes significantly large, the regularization terms $\frac{\rho_i^2}{2\alpha}$ and $\frac{\mu_j^2}{2\alpha}$ tend towards zero and the penalty term $\frac{\alpha}{2}\sum_{j=1}^{M}\sum_{i=1}^{N}(v_{ij}\rho_i - \mu_j)_+^2$ enforces the constraints $v_{ij}\rho_i - \mu_j = 0$, to avoid the cost function escalating towards infinity. Therefore, this optimizing problem will become the same as problem D as α goes to infinity.

In conclusion, using Proposition 3.1.1, which establishes the existence of a unique solution for problem D, we can deduce that μ^α, ρ^α will converge to μ^*, ρ^* as α goes to infinity.

Now, to demonstrate the convergence of the steps outlined in PT/SA to μ^α, ρ^α, it is essential to apply the principles from Theorem A.1.1 and Theorem A.1.2.

Here, the stochastic approximation variable is the vector $z = [\mu, \rho] \in \mathbb{R}^{(N+M)}$. We now proceed to find the noise vector and the ordinary differential equations (ODEs).

The noise vector is given by:

$$M(n+1) = [\ \lambda_j W_j(\mu_j) - \lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)]\ ,\ 0]$$

Calculating the noise mean and variance, we have:

$$||M(n+1)||^2 = \sum_{j=1}^{M} \Big(\lambda_j W_j(\mu_j) - \lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)]\Big)^2$$

$$E[\ ||M(n+1)||^2|\ \mathcal{F}_n] = \sum_{j=1}^{M} \lambda_j^2 W_j(\mu_j)(1 - W_j(\mu_j)) \leq \sum_{j=1}^{M} \lambda_j^2$$

$$E[\ M(n+1)|\ \mathcal{F}_n] = 0\ ,\ E[\ ||M(n+1)||^2|\ \mathcal{F}_n] \leq \sigma^2$$

The mean is zero and the variance limited. Thus, the third condition is satisfied.

Moreover, the ODEs are formulated as follows:

$$\dot{\mu}_j = -\frac{\mu_j}{\alpha} - \lambda_j W_j(\mu_j) + \alpha \sum_{i=1}^{N} (v_{ij}\rho_i - \mu_j)_+ = -\frac{\partial \mathcal{L}_\alpha}{\partial \mu_j} \tag{3.7}$$

$$\dot{\rho}_i = -\frac{\rho_i}{\alpha} + C_i - \alpha \sum_{j=1}^{M} v_{ij}(v_{ij}\rho_i - \mu_j)_+ = -\frac{\partial \mathcal{L}_\alpha}{\partial \rho_i} \tag{3.8}$$

$$\dot{z} = h(\mu, \rho) = \begin{bmatrix} -\frac{\mu_j}{\alpha} - \lambda_j W_j(\mu_j) + \alpha \sum_{i=1}^{N} (v_{ij}\rho_i - \mu_j)_+ \\ -\frac{\rho_i}{\alpha} + C_i - \alpha \sum_{j=1}^{M} v_{ij}(v_{ij}\rho_i - \mu_j)_+ \end{bmatrix} \tag{3.9}$$

Here, the function $h(\mu, \rho)$ represents the gradient of $\mathcal{L}_\alpha(\mu, \rho)$. Given that the term $\sum_{j=1}^{M} \frac{\mu_j^2}{2\alpha} + \sum_{i=1}^{N} \frac{\rho_i^2}{2\alpha}$ is $\frac{1}{\alpha}$-smooth, and the other terms being convex, $\mathcal{L}_\alpha(\mu, \rho)$ is $\frac{1}{\alpha}$-smooth. Consequently, h is $\frac{1}{\alpha}$-Lipschitz, satisfying the second condition.

Now, to demonstrate the convergence of the ODEs, we employ Theorem A.1.2. To do that, we need define the function $h_\infty(\mu, \rho)$ as follows:

$$h_\infty(\mu, \rho) = \lim_{c \to \infty} \frac{h(c\mu, c\rho)}{c} = \begin{bmatrix} -\frac{\mu_j}{\alpha} + \alpha \sum_{i=1}^{N} (v_{ij}\rho_i - \mu_j)_+ \\ -\frac{\rho_i}{\alpha} - \alpha \sum_{j=1}^{M} v_{ij}(v_{ij}\rho_i - \mu_j)_+ \end{bmatrix} \tag{3.10}$$

We observe that $h_\infty(\mu, \rho)$ is the gradient of the function $\frac{\alpha}{2} \sum_{j=1}^{M} \sum_{i=1}^{N} (v_{ij}\rho_i - \mu_j)_+^2 + \sum_{j=1}^{M} \frac{\mu_j^2}{2\alpha} + \sum_{i=1}^{N} \frac{\rho_i^2}{2\alpha}$, which is L-smooth.

To establish that the new ODE,

$$\begin{bmatrix} \dot{\mu} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} -\frac{\mu_j}{\alpha} + \alpha \sum_{i=1}^{N} (v_{ij}\rho_i - \mu_j)_+ \\ -\frac{\rho_i}{\alpha} - \alpha \sum_{j=1}^{M} v_{ij}(v_{ij}\rho_i - \mu_j)_+, \end{bmatrix} \tag{3.11}$$

has has the origin as its unique globally asymptotically stable equilibrium, we apply Lyapunov's direct method.

The chosen Lyapunov function, $V(\mu, \rho)$ is chosen as the norm of z, and is defined as:

$$V(\mu, \rho) = \frac{||\mu||^2}{2} + \frac{||\rho||^2}{2}$$

The derivative of V, denoted $\dot{V}$, is:

$$\dot{V} = \langle \mu \,, \dot{\mu} \rangle + \langle \rho \,, \dot{\rho} \rangle = \sum_{j=1}^{M} \Big(-\frac{\mu_j^2}{\alpha} + \alpha \sum_{i=1}^{N} \mu_j (v_{ij}\rho_i - \mu_j)_+ \Big) + \sum_{i=1}^{N} \Big(-\frac{\rho_i^2}{\alpha} - \alpha \sum_{j=1}^{M} v_{ij}\rho_i (v_{ij}\rho_i - \mu_j)_+ \Big)$$

$$\dot{V} = -\sum_{j=1}^{M} \frac{\mu_j^2}{\alpha} - \sum_{i=1}^{N} \frac{\rho_i^2}{\alpha} - \alpha \sum_{i=1}^{N} \sum_{j=1}^{M} (v_{ij}\rho_i - \mu_j)(v_{ij}\rho_i - \mu_j)_+$$

$$\dot{V}(\mu, \rho) = -\Big(\frac{||\mu||^2}{\alpha} + \frac{||\rho||^2}{\alpha} + \alpha \sum_{i=1}^{N} \sum_{j=1}^{M} (v_{ij}\rho_i - \mu_j)_+^2 \Big) \leq 0.$$

We have $\dot{V}(\mu, \rho) \leq 0$, and $\dot{V}(\mu, \rho) = 0$ gives $\mu = 0, \rho = 0$. Therefore the origin is unique globally asymptotically stable equilibrium of the ODE $\dot{z} = h_\infty(z)$. Therefore, using Theorem A.1.2, we have $\mu(n)$ and $\rho(n)$ will converge.

□

By integrating an update step for arrival times (in case that the arrival rates λ_j are unknown as well), and combining it with the bid adaptation algorithm of Equation PT/SA, a complete stochastic approximation algorithm for learning solutions to Problem D is specified by Algorithm 2.

Algorithm 2: SA and penalty method

---

for *Each Arriving Item* $\tau_{n+1}, \theta_{n+1}, p_{n+1}$ **do**

Update inter-arrival time estimates,

$\hat{\tau}_j(n+1) = \hat{\tau}_j(n) + a_n\big[\mathbb{1}_j(\theta_{n+1})\tau_{n+1} - \hat{\tau}_j(n)\big]$

$\hat{\tau}(n+1) = \hat{\tau}(n) + a_n\big[\tau_{n+1} - \hat{\tau}(n)\big]$

Place bid $x_j(n)$ observe $\mathbb{1}[p_{n+1} \leq x_j(n)]$ and update μ_j,

$\mu_j(n+1) = \mu_j(n) - a_n\big[\frac{\mathbb{1}_j(\theta_{n+1})}{\hat{\tau}(n)}\mathbb{1}[p_{n+1} \leq \mu_j(n)] - \alpha\sum_{i=1}^{N}(v_{ij}\rho_i - \mu_j)_+ + \frac{\mu_j(n)}{\alpha}\big]$

Update ρ_i

$\rho_i(n+1) = \rho_i(n) - a_n\big[\alpha\sum_{j=1}^{M} v_{ij}(v_{ij}\rho_i - \mu_j)_+ - C_i + \frac{\rho_i(n)}{\alpha}\big]$

Keep track of total supply targets

$s_j(n+1) = s_j(n) + a_n\mathbb{1}_j(\theta_{n+1})\big[\frac{1}{\hat{\tau}_j(n)}\mathbb{1}[p_{n+1} \leq \mu_j(n)] - s_j(n)\big]$

The first step of Algorithm 2 is used to provide adaptive estimates of the τ_j (used in penalty term $\frac{1}{\hat{\tau}_j(n)}\mathbb{1}[p_{n+1} \leq \mu_j(n)]$, when updating μ_j and s_j). The remaining steps update the bids and the supply rates.

3.2.2 Simulation

In order to illustrate the performance of our methods, we have carried out numerical simulations for an example contract management problem with $M = 5$ distinct item types, and $N = 6$ contracts. The prices (denominated in arbitrary monetary units) $\boldsymbol{p}(n) = \big(p_j(n), j \in [M]\big)$ for each type are drawn *i.i.d.* from Gamma distributions and arrive according to Poisson processes of rates (having units of Hz) $\boldsymbol{\lambda} = \big(\lambda_j, j \in [M]\big)$. The specific parameters are given by

$$\begin{aligned} \boldsymbol{\lambda} &= (3.0, 16.0, 18.0, 17.0, 18.0), \\ \mathbb{E}[\boldsymbol{p}(n)] &= (20, 23, 26, 29, 32), \\ \mathrm{Var}[\boldsymbol{p}(n)] &= (20, 18, 16, 14, 12), \end{aligned} \tag{3.12}$$

which is enough to fully specify the market model. The item type $j = 1$ is the cheapest, yet has by far the lowest supply and has a high variance – these parameters stress the stochastic approximation since the optimal solution is likely to require nearly all of the supply of type $j = 1$ that is available. The $N = 6$ contracts are specified through the sets $\mathcal{A} = \big(\mathcal{A}_i, i \in [N]\big)$ and $\boldsymbol{C} = \big(C_i, i \in [N]\big)$ as

$$V = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{C} = \big(6.0, 5.1, 2.7, 9.9, 6.6, 7.8\big). \tag{3.13}$$

Despite being the most expensive, item 5 is also the most frequent and in high demand ($V[i, 5] = 1$), suggesting that its optimal supply rate may not differ significantly from that of other items.

In order to initialize Algorithm 2, an initial array $\mu(0)$ needs to be specified. In principle, since the Algorithm is convergent for any starting point, the algorithm can be initialized at random or arbitrarily (e.g., at zero). However, superior initialization methods are available.

Firstly, if prior information is available, then the nominal solution of Problem (D) can be solved with standard convex optimization software (e.g., [14]) using an appropriate model of Λ_j which captures this prior knowledge [24], and then the solution of this program can be used as the initialization. Moreover, For the purposes of our simulation we have used $a_n = 10 * n^{-1}$.

Finally, the parameter α needs to be taken large, but using too large of a value of α can easily cause numerical overflow at the early stages of the approximation. α can be increased (e.g., via $\alpha \leftarrow (1 + \kappa)\alpha, \kappa > 0$) throughout the simulation (perhaps up to some large maximum value) whenever the iterates are detected to be infeasible. This latter method has been used in our simulations in order to avoid any unusual discontinuous jumps in the algorithm, with the value $\kappa = 0.01$, up to a maximum of $\alpha < 10^4$. This maximum value seems to be reasonable based on simulation evidence of [17].

Numerical results are given in Figure 3.1 and 3.2. With reference to Code , we have calculated

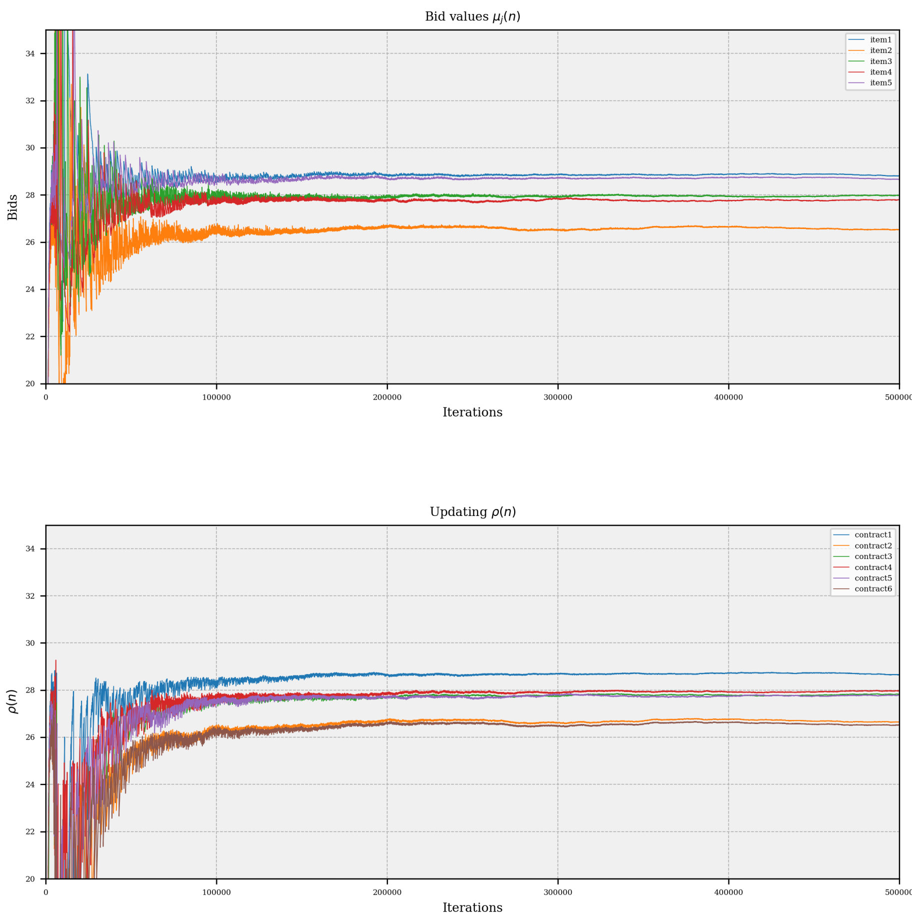

Figure 3.1: Convergence of Bids and supply Rates
Convergence of the bids for the Penalty Term Algorithm for Inequality Constraints. The iterates are extremely noisy, but they start converging after 100000 iteration.

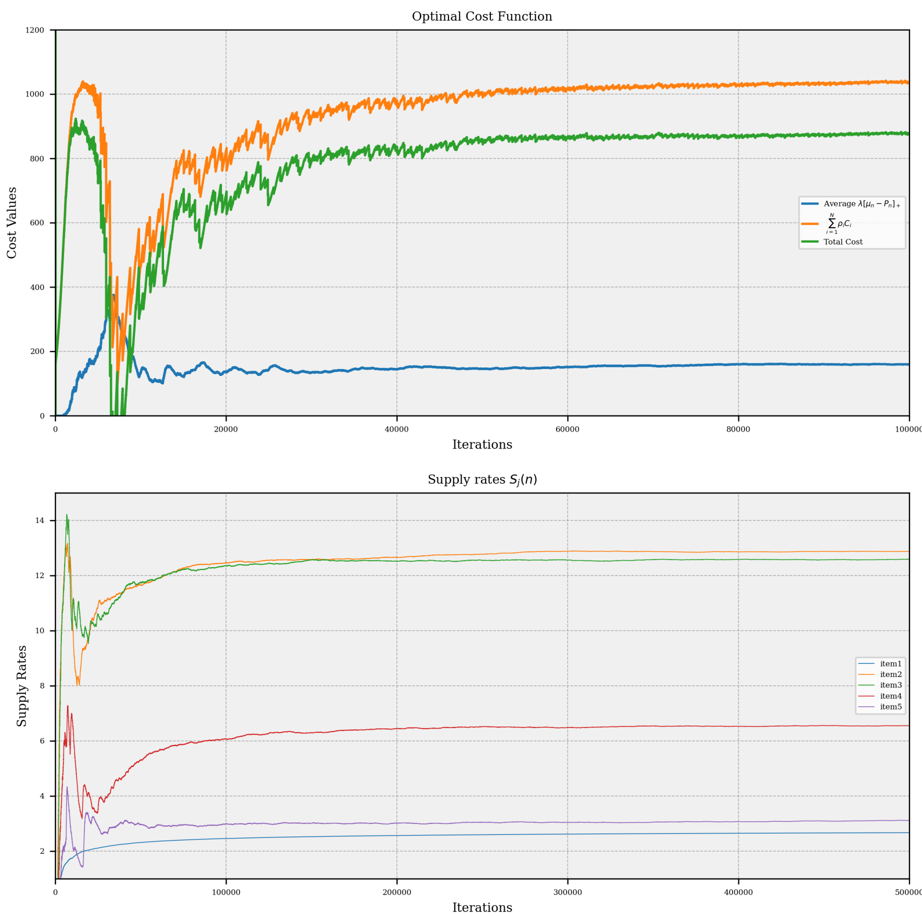

Figure 3.2: Convergence of the Cost Functions and Supply Rates
Convergence of the two component of the cost function and the total cost function for the Penalty Term Algorithm for Inequality Constraints, is shown.

$$\text{bids} = [28.66 \quad 26.52 \quad 27.98 \quad 27.80 \quad 28.66]$$
$$\text{supply rates} = [2.67 \quad 12.87 \quad 12.59 \quad 6.55 \quad 3.11]$$
$$\rho = [28.65 \quad 26.51 \quad 27.97 \quad 27.96 \quad 27.78 \quad 26.51]$$
$$R = \begin{bmatrix} 2.78 & 0 & 0 & 0 & 3.22 \\ 0 & 5.11 & 0 & 0 & 0 \\ 0 & 0 & 2.71 & 0 & 0 \\ 0 & 0 & 9.92 & 0 & 0 \\ 0 & 0 & 0 & 6.6 & 0 \\ 0 & 7.81 & 0 & 0 & 0 \end{bmatrix}$$
$$\text{Total cost} = 888.45$$

We finally see from both of these figures that the algorithms have converged after to approximately $100,000$ item arrivals. According to statistics reported by [46], these arrival rates should accurate within reasonable orders of magnitude. Still, the actual arrival rates will be highly dependent upon how the type of an item is characterized.

3.3 Projected Noisy Gradient Descent

In section 3.1, we expressed the dual problem as a minimization of a convex function subject to linear inequality constraints. All the feasible points which satisfy the constraints are denoted as G. And, in this case, G is equal to the set $\{z \in \mathbb{R}^{N+M} \mid g_k(z) \leq 0, k = 1, ..., NM\}$, where $g_k : \mathbb{R}^{N+M} \to \mathbb{R}$ are linear functions, with $g_{(i,j)}(\mu, \rho) = V_{ij}\rho_i - \mu_j$ ($g_{ij}(z) = V_{ij}z_{i+M} - z_j$). The set G is convex, nonempty, and closed.

Utilizing this observation, we aim to apply the Projected Gradient Descent algorithm to address the dual problem and solve for the optimal point.

In this section, we will implement the Projected Gradient Descent algorithm [5, 27]. Then to show the convergence of this algorithm , we will use Theorem A.1.3, which establishes that with some extra assumptions on G, the Projected Stochastic Approximation steps will converge to the optimal solution of Problem D.

The Projected Gradient Descent is applicable to convex optimization problems in the form of:

$$\min_z f(z) \text{ subject to } z \in G;$$

where f is a convex differentiable function and G is a convex set. The update steps are given by $z(n+1) = \Pi_G(z(n) - a_n \nabla f(x))$. [2]

3.3.1 Algorithm and convergence

Now Given the cost function in D, we can derive the update steps for μ and ρ as follows:

$$\begin{aligned} \tilde{\rho}_i(n+1) &= \rho_i(n) + a_n C_i, \\ \tilde{\mu}_j(n+1) &= \mu_j(n) - a_n \lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)] \end{aligned}$$
$$\begin{bmatrix} \mu(n+1) \\ \rho(n+1) \end{bmatrix} = \Pi_G\Big(\begin{bmatrix} \tilde{\mu}(n+1) \\ \tilde{\rho}(n+1) \end{bmatrix} \Big) \tag{PGD}$$

The following proposition shows the convergence of PGD steps.

Proposition 3.3.1. *[Projected Noisy Gradient Descent convergence]*
The steps $\mu(n), \rho(n)$ in the equations PGD converge to a unique solution μ^*, ρ^*, where μ^*, ρ^* represent the optimal solution to problem D.

[2] $\Pi_G(x)$ denotes the projection of the point x onto the convex set G, i.e., $\Pi_G(x) = \inf_{z \in G} ||z - x||$

Proof. To prove the convergence of this algorithm, we use Theorem A.1.3 and verify that all the conditions are satisfied. For the first two conditions we need to show the properties for set G. We can express G as:

$$\begin{aligned} &G = \{z \in \mathbb{R}^{N+M} \mid Az \leq 0\}, \\ &G = \{z \in \mathbb{R}^{N+M} \mid g_k(z) \leq 0, k = \{1, ..., NM\}\}, \end{aligned} \tag{3.14}$$

where $A \in \mathbb{R}^{NM \times (N+M)}$ is a matrix with the k^{th} row being the vector $a \in \mathbb{R}^{N+M}$, $a = [0, ..., -1, ..., 0 \mid 0, ..., v_{ij}, ..., 0]$ (-1 is at the j^{th} position, and v_{ij} at i^{th}).

We observe that G is a convex, non-empty, and closed set. Additionally, $g_k(z) = a^T z$ are continuously differentiable linear functions, and therefore the gradients of the active constraints are linearly independent (2nd condition).

The remaining conditions are satisfied similarly to the proof for Proposition 3.2.1. we only need to demonstrate that support of the conditional distribution of M_n is a closed bounded set, $A(z_n)$. We have:

$$M_j(n+1) = \lambda_j W_j(\mu_j) - \lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)]$$

which implies, $-\lambda \leq M(n) \leq \lambda$ or $M_n(z_n) \in A(z_n) = [-\lambda, \lambda]$.
Consequently, μ_n and ρ_n will converge to the point μ^*, ρ^*, where μ^*, ρ^* are in the set:

$$\text{KT} = \{z \in G \mid \forall k = \{1, ..., NM\}\ ,\ \exists \lambda_k \leq 0 \text{ such that} : \lambda_k g_k(z) = 0, -h(z) + \sum_{i=1}^{NM} \lambda_k \nabla g_k(z) = 0\}.$$

to show that μ^*, ρ^* are also the solution to problem D, we use theorem A.2.1.
Each point in the set KT satisfies all four conditions in the Karush-Kuhn-Tucker (KKT) Conditions. Moreover, since the functions $g_k(z) = a^T z$ are unbounded linear functions, there exist a slater point [3]. For example, in any point with $\mu_j > 0$ and $\rho_i < 0$, we have $g_{(i,j)}(\mu, \rho) = V_{ij}\rho_i - \mu_j < 0$; and therefore it is a Slater point.

Therefore $\mu(n), \rho(n)$ will converge to the optimal solution to problem D. □

[3]Slater point is defined as follows: $(\exists y \in \mathbb{R}^d)(\forall i \in \{1, \ldots, m\}).\quad g_i(y) < 0$

By adding an update step for arrival times (in case that the arrival rates λ_j are unknown as well), and combining it with the bid adaptation algorithm of Equation *PGD*, a complete projected stochastic approximation algorithm for learning solutions to Problem D is specified by Algorithm 3.

Algorithm 3: Projected SA

for *Each Arriving Item* $\tau_{n+1}, \theta_{n+1}, p_{n+1}$ **do**

Update inter-arrival time estimates,

$\hat{\tau}_j(n+1) = \hat{\tau}_j(n) + a_n\big[\mathbb{1}_j(\theta_{n+1})\tau_{n+1} - \hat{\tau}_j(n)\big]$

$\hat{\tau}(n+1) = \hat{\tau}(n) + a_n\big[\tau_{n+1} - \hat{\tau}(n)\big]$

Place bid $x_j(n)$ observe $\mathbb{1}[p_{n+1} \leq x_j(n)]$ and update μ_j, ρ_i,

$\tilde{\mu}_j(n+1) = \mu_j(n) - a_n \frac{\mathbb{1}_j(\theta_{n+1})}{\hat{\tau}(n)} \mathbb{1}[p_{n+1} \leq \mu_j(n)]$

$\tilde{\rho}_i(n+1) = \rho_i(n) + a_n C_i$

Project μ and ρ on G,

$$\begin{bmatrix} \mu(n+1) \\ \rho(n+1) \end{bmatrix} = \Pi_G\Big(\begin{bmatrix} \tilde{\mu}(n+1) \\ \tilde{\rho}(n+1) \end{bmatrix}\Big)$$

Keep track of total supply targets,

$s_j(n+1) = s_j(n) + a_n \mathbb{1}_j(\theta_{n+1})\big[\frac{1}{\hat{\tau}_j(n)}\mathbb{1}[p_{n+1} \leq \mu_j(n)] - s_j(n)\big]$

Remark 3.3.1. *Projection of a point x onto the set G, when $G = \{x \in \mathbb{R}^{N+M} \mid Ax \leq 0\}$, does not have a closed form solution.*
Instead, it is a convex optimization problem, formulated as: $\min_y ||x - y||^2 \text{ subject to } Ay \leq 0$. *Hence, within each step of this algorithm, we need to solve another convex optimization.*

Fortunately, this problem takes the form of quadratic convex optimization problem with linear inequality constraints, and can be solved with Quadratic programming algorithms.

During simulations, we simply employ the CVXPY library in Python, which offers a fast step for our use.

3.3.2 Simulation

To compare the results of the algorithms, we will set all the market and contract variables similar to those used in the last algorithm. Furthermore, in this algorithm, there does

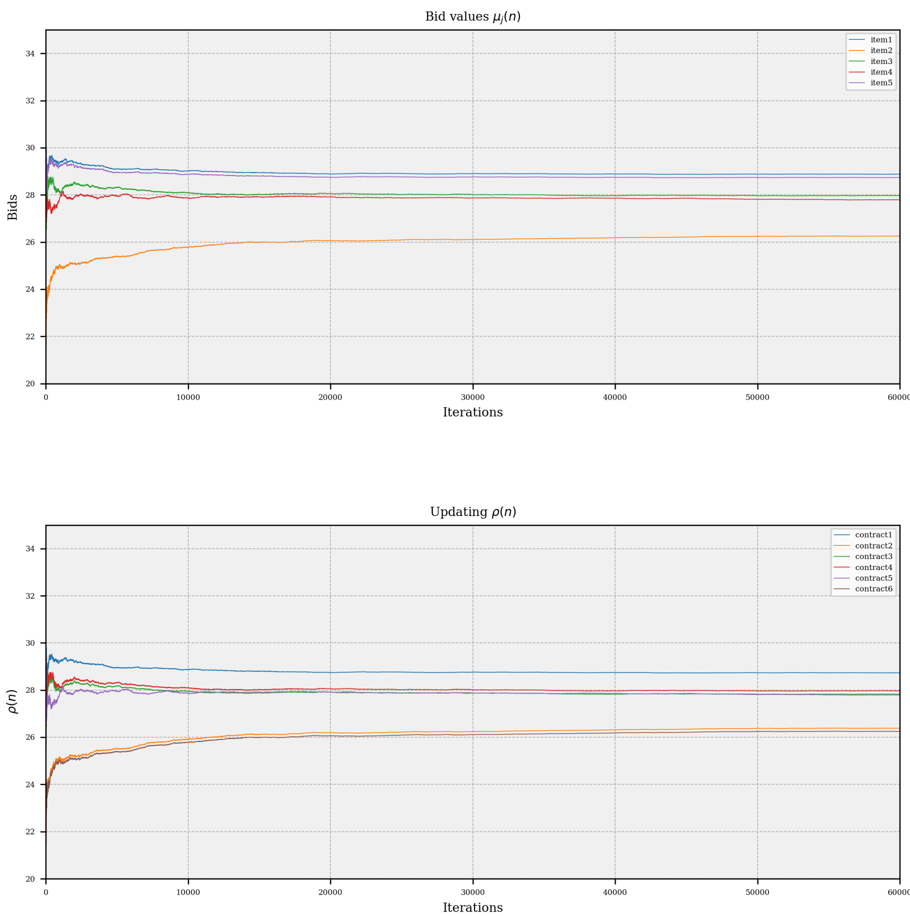

Figure 3.3: Convergence of Bids and supply Rates
Convergence of the bids for the Projected Noisy Gradient Descent. The iterates start converging after 10000 iteration.

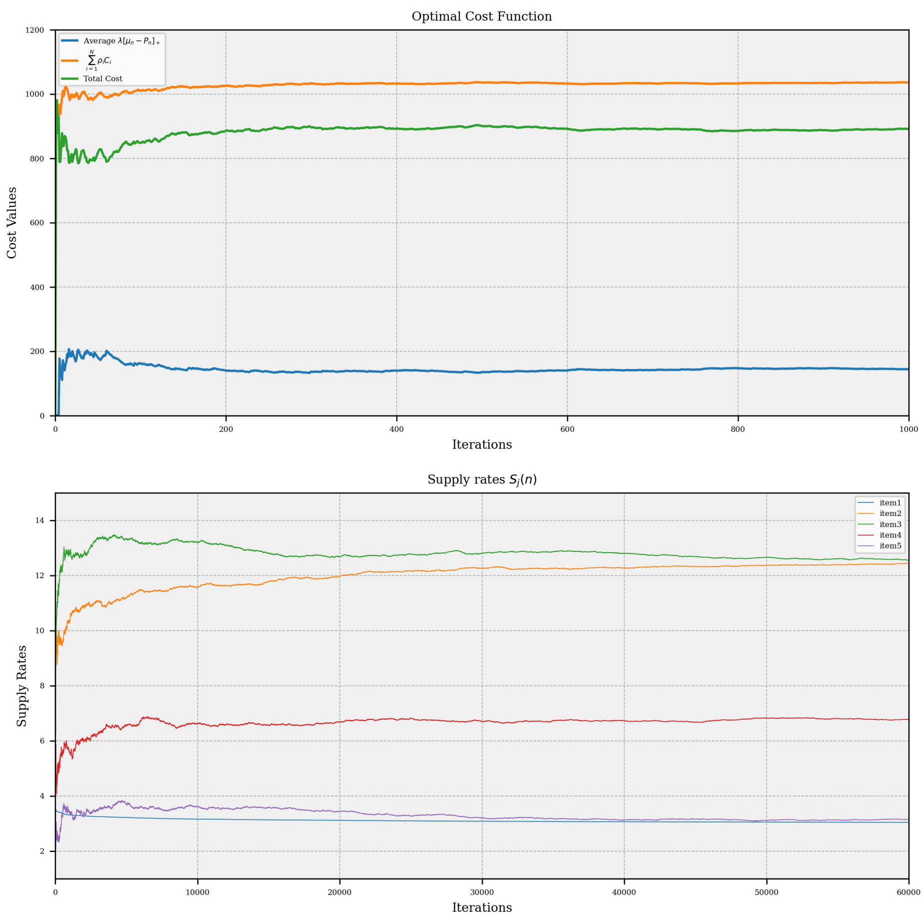

Figure 3.4: Convergence of the Cost Functions and Supply Rates
Convergence of the two component of the cost function and the total cost function for the Projected Noisy Gradient Descent, is shown.

not exist any hyperparameter α to determine. However, we need to create a function in Python using the CVXPY library to combine μ and ρ and then project them onto the set G. And again, For the purposes of our simulation we have used $a_n = 10 * n^{-1}$.

Numerical results are given in Figure 3.3 and 3.4. With reference to Code, we have calculated :

$$\text{bids} = [28.72 \quad 26.36 \quad 27.96 \quad 27.78 \quad 28.72] \tag{3.15}$$

$$\text{supply rates} = [2.89 \quad 12.64 \quad 12.53 \quad 6.61 \quad 3.13] \tag{3.16}$$

$$\rho = [28.72 \quad 26.36 \quad 27.96 \quad 27.96 \quad 27.78 \quad 26.36] \tag{3.17}$$

$$\text{Total cost} = 889.01 \tag{3.18}$$

$$R = \begin{bmatrix} 2.96 & 0 & 0 & 0 & 3.03 \\ 0 & 5.09 & 0 & 0 & 0 \\ 0 & 0 & 2.69 & 0 & 0 \\ 0.06 & 0 & 9.7 & 0 & 0.13 \\ 0 & 0 & 0.16 & 6.43 & 0 \\ 0 & 7.58 & 0 & 0.21 & 0 \end{bmatrix} \tag{3.19}$$

We finally see from both of these figures that the algorithms have converged after approximately $10,000$ item arrivals.

As predicted, the supply rate for item 1 closely matches its arrival rate, indicating that the algorithm attempts to purchase all items of type 1. And interestingly, despite being the most expensive, the supply rate for the last item is comparable to that of the others due to its high demand.

3.4 Subgradiant Method

In this part, we utilize the fact that the optimal solution to problem D, denoted as μ^* and ρ^*, satisfies the property $\mu_j^* = \max_i(v_{ij}\rho_i^*)$ and $\rho_i^* = \min_j(\frac{\mu_j^*}{v_{ij}})$ (as mentioned in observation 5 in proposition 3.1.2).
The aim is to reformulate the cost function solely in terms of μ and eliminate the inequality condition.

3.4.1 Algorithm and convergence

In problem D, we seek to minimize the function $L(\mu, \rho)$ over the closed linear set $S = \{\mu, \rho \mid v_{ij}\rho_i \leq \mu_j, \forall i \in N, j \in M\}$.
We introduce another set $\tilde{S} = \{\mu, \rho \mid \mu_j = \max_{i\in\mathcal{B}_j}(v_{ij}\rho_i), \rho_i = \min_{j\in\mathcal{A}_i}(\frac{\mu_j}{v_{ij}}), \forall i \in N, j \in M\}$. It is evident that $\tilde{S} \subset S$. Moreover, based on observations 1,5 in proposition 3.1.2, we conclude that $\min_{\mu,\rho\in S} L(\mu, \rho)$ has a unique solution μ^*, ρ^*, where $\mu^*, \rho^* \in \tilde{S}$.
Therefore, we deduce that $\min_{\mu,\rho\in\tilde{S}} L(\mu, \rho)$ also has a unique solution, which is equal to the solution of problem D.
Consequently, we can minimize the cost function over the smaller subject set, thereby simplifying the algorithm.

We introduce a new problem formulation:

$$\begin{aligned} &\underset{\rho,\mu}{\text{minimize}} \quad \sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j) - \sum_{i=1}^{N} \rho_i C_i \\ &\text{subject to } \mu_j = \max_{i\in\mathcal{B}_j}(v_{ij}\rho_i), \rho_i = \min_{j\in\mathcal{A}_i}(\frac{\mu_j}{v_{ij}}), \forall i \in N, j \in M \end{aligned} \tag{3.20}$$

The advantage of this form of the dual problem is that we can express the cost function solely in terms of either μ or ρ. Consequently, we have:

$$\underset{\mu}{\text{minimize}} \quad \sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j) - \sum_{i=1}^{N} C_i \min_{j\in\mathcal{A}_i}(\frac{\mu_j}{v_{ij}}) \tag{$\tilde{D}$}$$

Now that the function is solely a function of μ, it benefits from the advantage of no longer having a condition on the optimization. However, it also presents a disadvantage: it is no longer differentiable, thus rendering stochastic approximation ineffective due to the absence of a gradient for the function f.
Considering these factors, we recommend utilizing the subgradient method to solve problem $\tilde{D}$. This method is particularly suitable for optimizing non-differentiable functions and can efficiently handle the absence of a gradient in this context.
Given that we now have a convex optimization problem and access to noisy subgradients of the cost function, we will use the subgraiadent method. In this method, we iteratively update the step iterations using the subgradient.
Before proceeding further, we present a proposition outlining the conditions required for the convergence of the subgradient method to the minimum point of the cost function.

Proposition 3.4.1 (subgradient convergence [9])**.** *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a non differentiable convex function with a noisy subgradiant vector $\tilde{g} \in \mathbb{R}^n$.*
The stochastic subgradient method uses the standard update

$$x^{(k+1)} = x^{(k)} - \alpha_k \tilde{g}^{(k)} \tag{3.21}$$

where $x^{(k)}$ is the k^{th} iterate, $\alpha_k > 0$ is the k^{th} step size, and $\tilde{g}^{(k)}$ is a noisy subgradient of f at $x^{(k)}$,

$$\mathbf{E}\left(\tilde{g}^{(k)} \mid x^{(k)}\right) = g^{(k)} \in \partial f\left(x^{(k)}\right).$$

Moreover we will define the minimum of all the $f(x^{(i)})$ as,

$$f_{min}^{(k)} = \min\left\{f\left(x^{(1)}\right), \ldots, f\left(x^{(k)}\right)\right\}.$$

If the following conditions hold:

- *(Robbins-Monro)* $\sum_{n=0}^{\infty} a_n = \infty, \sum_{n=0}^{\infty} a_n^2 < \infty$.
- *There exist G where* $\mathrm{E}\left\|g^{(k)}\right\|_2^2 \leq G^2$ *for all k.*
- *There exist R where* $\mathbf{E}\left\|x^{(1)} - x^{\star}\right\|_2^2 \leq R^2$

Then the algorithm will converge and we have

$$\lim_{k \to \infty} f_{min}^{(k)} = f^{\star}$$

Proof. We have

$$\begin{aligned}
\mathbf{E}\left(\left\|x^{(k+1)}-x^{\star}\right\|_2^2 \mid x^{(k)}\right) &= \mathbf{E}\left(\left\|x^{(k)}-\alpha_k\tilde{g}^{(k)}-x^{\star}\right\|_2^2 \mid x^{(k)}\right) \\
&= \left\|x^{(k)}-x^{\star}\right\|_2^2 - 2\alpha_k\mathbf{E}\left(\tilde{g}^{(k)T}\left(x^{(k)}-x^{\star}\right) \mid x^{(k)}\right) + \alpha_k^2\mathbf{E}\left(\left\|\tilde{g}^{(k)}\right\|_2^2 \mid x^{(k)}\right) \\
&= \left\|x^{(k)}-x^{\star}\right\|_2^2 - 2\alpha_k\mathbf{E}\left(\tilde{g}^{(k)} \mid x^{(k)}\right)^T\left(x^{(k)}-x^{\star}\right) + \alpha_k^2\mathbf{E}\left(\left\|\tilde{g}^{(k)}\right\|_2^2 \mid x^{(k)}\right) \\
&\leq \left\|x^{(k)}-x^{\star}\right\|_2^2 - 2\alpha_k\left(f\left(x^{(k)}\right)-f^{\star}\right) + \alpha_k^2\mathbf{E}\left(\left\|\tilde{g}^{(k)}\right\|_2^2 \mid x^{(k)}\right)
\end{aligned}$$

Here, we utilize the fact that $\mathbf{E}\left(\tilde{g}^{(k)} \mid x^{(k)}\right) = g^{(k)} \in \partial f\left(x^{(k)}\right)$ and the subgradient property, for the inequality. Now we take expectation to get

$$\mathbf{E}\left\|x^{(k+1)}-x^{\star}\right\|_2^2 \leq \mathbf{E}\left\|x^{(k)}-x^{\star}\right\|_2^2 - 2\alpha_k\left(\mathbf{E}f\left(x^{(k)}\right)-f^{\star}\right) + \alpha_k^2G^2,$$

and by summing up over k, we arrive at,

$$0 \leq \mathbf{E}\left\|x^{(k+1)}-x^{\star}\right\|_2^2 \leq \mathbf{E}\left\|x^{(1)}-x^{\star}\right\|_2^2 - 2\sum_{i=1}^{k}\alpha_i\left(\mathbf{E}f\left(x^{(i)}\right)-f^{\star}\right) + G^2\sum_{i=1}^{k}\alpha_i^2.$$

Thus, we establish

$$\min_{i=1,\dots,k}\left(\mathbf{E}f\left(x^{(i)}\right)-f^{\star}\right) \leq \frac{R^2+G^2\|\alpha\|_2^2}{2\sum_{i=1}^{k}\alpha_i},$$

which shows that $\min_{i=1,\dots,k}\mathbf{E}f\left(x^{(i)}\right)$ converges to $f^{\star}$. Finally, we note that by Jensen's inequality and concavity of the minimum function, we have

$$\mathbf{E}f_{\min}^{(k)} = \mathbf{E}\min_{i=1,\dots,k} f\left(x^{(i)}\right) \leq \min_{i=1,\dots,k}\mathbf{E}f\left(x^{(i)}\right),$$

so $\mathbf{E}f_{\min}^{(k)}$ also converges to $f^{\star}$. □

To demonstrate the applicability of Proposition 3.4.1 to the dual problem $\tilde{D}$, we first check the three conditions. The cost function in problem $\tilde{D}$ is,

$$\tilde{L}(\mu) = \sum_{j=1}^{M}\lambda_j\Lambda_j^*(\mu_j) - \sum_{i=1}^{N}C_i\min_{j\in\mathcal{A}_i}\left(\frac{\mu_j}{v_{ij}}\right).$$

The derivative of the first term $\lambda_j \Lambda_j^*(\mu_j)$ is $\lambda_j W_j^*(\mu_j) = S_j$. However, since we lack prior information about the market, we replace S_j with the function $\lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)]$.
Next, to compute the subgradient of the term $\min\limits_{j\in\mathcal{A}_i}(\frac{\mu_j}{v_{ij}})$, we need to first define the set, $\mathcal{A}_i^* = \{j \in \mathcal{A}_i \mid j \in \operatorname*{argmin}\limits_{j\in\mathcal{A}_i}(\frac{\mu_j}{v_{ij}})\}$. The derivative is then given by:

$$\partial(\min_{j\in\mathcal{A}_i}(\frac{\mu_j}{v_{ij}})) = \text{convex hull} \bigcup_{j\in\mathcal{A}_i^*} [0, ..., \frac{1}{v_{ij}}, ..., 0].$$

This set is either singleton or has more than one component. In the case of the latter we can arbitrarily choose one of the $j \in \mathcal{A}_i^*$ to represent $\mathcal{A}_i^*$ in the convex hull (convex hull is any linear combination of vectors with $j \in \mathcal{A}_i^*$).
Therefore, the resulting noisy subgradient is:

$$\partial_j \tilde{L}(\mu) = \lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)] - \sum_{i=1}^{N} \frac{C_i}{v_{ij}} \mathbb{1}[j \in \mathcal{A}_i^*] \qquad (subgradiant)$$

To update the μ's in our algorithm, we proceed as follows:

$$\mu_j(n+1) = \mu_j(n) - b_n\big(\lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)] - \sum_{i=1}^{N} \frac{C_i}{v_{ij}} \mathbb{1}[j \in \mathcal{A}_i^*]\big) \qquad (SG)$$

Where b_n satisfies the Robbins-Monro conditions. To demonstrate the convergence of the subgradient update given by Equation SG, we us Proposition 3.4.1. It suffices to show that the average squared norm of the noisy subgradient is bounded.
we can write the subgradient as:

$$\partial_j \tilde{L}(\mu) = \lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)] - \sum_{i=1}^{N} \frac{C_i}{v_{ij}} \mathbb{1}[j \in \mathcal{A}_i^*] = (\lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)] - S_j) + (S_j - \sum_{i=1}^{N} \frac{C_i}{v_{ij}} \mathbb{1}[j \in \mathcal{A}_i^*])$$

We define the first term, $(\lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)] - S_j)$, as noise and denote it as the vector M_n. Thus, we have:

$$M_j(n) = \lambda_j \mathbb{1}[p_{n+1} \leq \mu_j(n)] - S_j = \lambda_j\Big(\mathbb{1}[p_{n+1} \leq \mu_j(n)] - E\big(\mathbb{1}[p_{n+1} \leq \mu_j(n)]\big)\Big)$$

Hence, we have $\mathbf{E}\left(M_j(n) \mid \mu_j(n)\right) = 0$ and,

$$\begin{aligned} Var\left(M_j(n) \mid \mu_j(n)\right) &= \lambda_j^2\Big(E\big(\mathbb{1}[p_{n+1} \leq \mu_j(n)]\big) - E^2\big(\mathbb{1}[p_{n+1} \leq \mu_j(n)]\big)\Big) \\ &= \lambda_j^2 W_j(\mu_j(n))\Big(1 - W_j(\mu_j(n))\Big) \leq \lambda_j^2 \end{aligned} \qquad (3.22)$$

Therefore $\mathbf{E}\,\|M_n\|_2^2 \leq \sum_{j=1}^{M} \lambda_j^2$.
Likewise we have a upper bound for the deterministic part of subgradiant which is $\sum_{j=1}^{M} \lambda_j^2 + \sum_{j=1}^{M}(\sum_{i=1}^{N} \frac{C_i}{v_{ij}})^2$
So there exist $G^2 = 2\sum_{j=1}^{M} \lambda_j^2 + \sum_{j=1}^{M}(\sum_{i=1}^{N} \frac{C_i}{v_{ij}})^2$ where $\mathrm{E}\left\|\partial\tilde{L}(n)\right\|_2^2 \leq G^2$ for all n.
By Proposition 3.4.1, the algorithm described in Equation *SG* will converge. Considering the aforementioned analysis in the beginning of this section, we conclude that the convergence point will be the same as the solution of problem *D*.

Combining the derivatives in Equation *SG* with a stochastic approximation algorithm to update S_j's and τ_j's, we obtain a complete one-timescale algorithm for learning solutions to Problem *D*. This algorithm is specified by Algorithm 4.

Algorithm 4: Subgradiant step for μ

```
for Each Arriving Item τ_{n+1}, θ_{n+1}, p_{n+1} do
    # Update inter-arrival time estimates,
    τ̂_j(n+1) = τ̂_j(n) + a_n [ 1_j(θ_{n+1}) τ_{n+1} − τ̂_j(n) ]
    τ̂(n+1) = τ̂(n) + a_n [ τ_{n+1} − τ̂(n) ]
    # Find the set A*_i(n) and randomly choose one of its components ,
    A*_i(n) = { j ∈ A_i | j ∈ argmin_{j∈A_i} ( μ_j(n) / v_ij ) }
    # Place bid x_j(n) observe 1[p_{n+1} ≤ x_j(n)] and update μ_j,
    μ_j(n+1) = μ_j(n) − a_n [ (1_j(θ_{n+1}) / τ̂(n)) 1[p_{n+1} ≤ μ_j(n)] − Σ_{i=1}^{N} (C_i / v_ij) 1[j ∈ A*_i(n)] ]
    # Keep track of total supply targets
    S_j(n+1) = S_j(n) + a_n 1_j(θ_{n+1}) [ (1 / τ̂_j(n)) 1[p_{n+1} ≤ μ_j(n)] − S_j(n) ]
    # Keep track of total ρ_i
    ρ_i = min_{j∈A_i} ( μ_j / v_ij )
```

Here we omitted a derivative step for updating ρ and instead directly applied the fact that $\rho_i^* = \min_{j\in\mathcal{A}_i}(\frac{\mu_j^*}{v_{ij}})$.

3.4.2 Simulation

To compare the results of the algorithms, we will set all the market and contract variables similar to those used in the last algorithm.

Numerical results are given in Figure 3.5 and 3.6. With reference to Code, we have calculated :

$$\text{bids} = [28.76 \quad 26.56 \quad 27.95 \quad 27.79 \quad 28.76] \tag{3.23}$$

$$\text{supply rates} = [2.85 \quad 12.87 \quad 12.61 \quad 6.58 \quad 3.17] \tag{3.24}$$

$$\rho = [28.76 \quad 26.57 \quad 27.93 \quad 27.93 \quad 27.78 \quad 26.57] \tag{3.25}$$

$$\text{Total cost} = 886.83 \tag{3.26}$$

$$R = \begin{bmatrix} 2.86 & 0 & 0 & 0 & 3.14 \\ 0 & 5.1 & 0 & 0 & 0 \\ 0 & 0 & 2.7 & 0 & 0 \\ 0 & 0 & 9.86 & 0 & 0 \\ 0 & 0 & 0 & 6.55 & 0 \\ 0 & 7.76 & 0 & 0 & 0 \end{bmatrix} \tag{3.27}$$

We finally see from both of these figures that the algorithms have converged after to approximately $10,000$ item arrivals.

By comparing μ and ρ, we can construct the matrix Θ, where $\theta_{ij} = \mu_j - v_{ij}\rho_i$:

$$\Theta = \begin{bmatrix} 0 & 26 & 27 & 27 & 0 \\ 28 & 0 & 27 & 1 & 2 \\ 28 & 26 & 0 & 27 & 0 \\ 0 & 26 & 0 & 27 & 0 \\ 28 & 26 & 0 & 0 & 0 \\ 28 & 0 & 27 & 1 & 2 \end{bmatrix}$$

Drawing from Observation 3.1.2, if $v_{ij} > 0$ and $\theta_{ij} > 0$, then it follows that $R_{ij} = 0$. As a result, we establish:

$$\begin{aligned} \mathcal{A} &= \big(\{1,5\}, \{2,4,5\}, \{3,5\}, \{1,3,5\}, \{3,4,5\}, \{2,4,5\}\big), \\ \mathcal{A}^* &= \big(\{1,5\}, \{2\}, \{3,5\}, \{1,3,5\}, \{3,4,5\}, \{2\}\big). \end{aligned} \tag{3.28}$$

The matrix R contains $MN = 36$ variables. By examining matrix V, we can immediately reduce this to $d = \sum_{i=1}^{N} |\mathcal{A}_i| = 16$ nonzero components. Applying the observation 3.1.2 and considering Θ, we can further reduce the dimensionality to $d^* = \sum_{i=1}^{N} |\mathcal{A}_i^*| = 12$. Ultimately, simulations reveal that the actual number of nonzero components in R is 7.

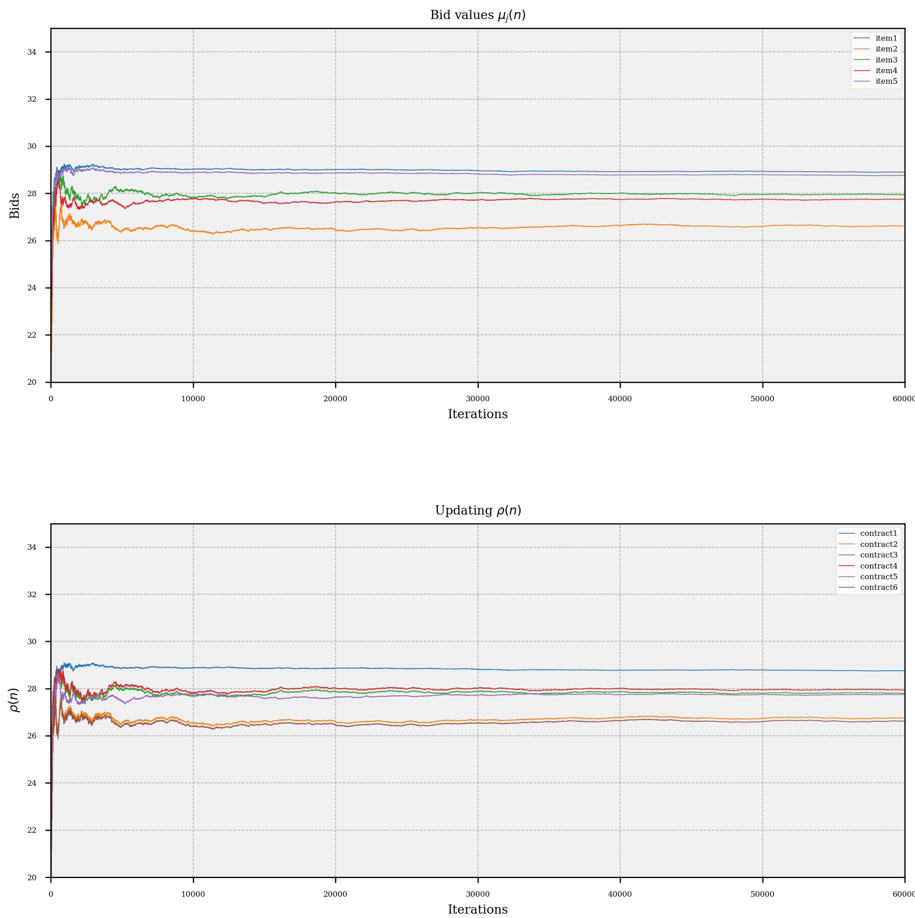

Figure 3.5: Convergence of Bids and supply Rates
Convergence of the bids for the Sub-Gradient Descent. The iterates start converging after 10000 iteration.

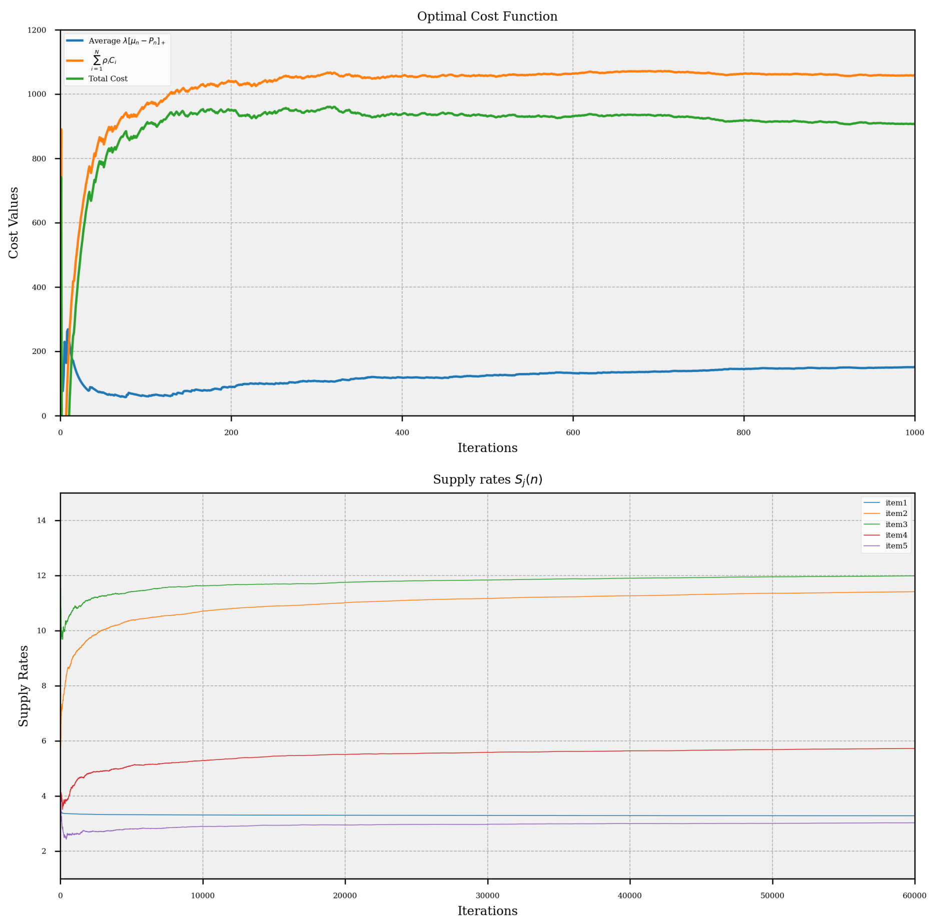

Figure 3.6: Convergence of the Cost Functions and Supply Rates
Convergence of the two component of the cost function and the total cost function for the Sub-Gradient Descent, is shown.

3.4.3 Comparison Between The Algorithms

This section evaluates the performance of the three algorithms presented in this chapter, highlighting their advantages and disadvantages based on variable management, convergence speed, computational demands, and proximity to optimal bids.

Variable Management: The Penalty Term method and Projected Gradient Descent (PGD) algorithms involve $N + M$ variables, as they can not separate updates of μ from ρ. This contrasts with the Subgradient Descent algorithm, which uniquely benefits from variable separation, directly updating μ (equal to the bids, which is our primary variable of interest). Thus, in terms of variable management, Subgradient Descent offers a significant advantage by focusing updates on only N variables.

Convergence Speed and Noise: Our simulations indicate that the Penalty Term algorithm is approximately ten times slower in convergence rate than the others and have the highest noise levels. This algorithm's initial iterations are particularly impacted by the large penalty term, and to manage this we need to limit the bids to a boundary. Between Subgradient Descent and PGD, PGD achieves faster convergence. However, Subgradient Descent results in a smoother final value (in PGD, the noise is projected on the constraint set as well, which will increase the noise level).

Computational Demands: PGD is computationally more intensive due to the necessity of solving a projection convex problem at every step using CXVPY. Conversely, the Penalty Term algorithm's primary computational challenge lies in optimizing the extra hyperparameter α, which must increase over time. The rate of this increase significantly influences the convergence speed, presenting an additional layer of complexity in algorithm tuning.

Proximity to Optimal Bids: When evaluating which algorithm's outcomes most closely approximate the optimal bids, we can compare their cost functions. The Subgradient Descent algorithm exhibits the lowest cost, suggesting its final bids are nearest to the optimal solutions.

In summary, while PGD offers quicker convergence and Subgradient Descent provides a smoother final value with less computational overhead, the Penalty Term algorithm struggles with slower convergence and additional complexity due to hyperparameter tuning. Subgradient Descent emerges as the most effective in approximating optimal bids, striking a balance between efficiency and computational demand.

3.5 Optimal Cost and Acquisition Rates

In the previous sections, we explored three online convex optimization algorithms designed to use item arrival data for updating the bid values and supply rates.

In this section, we present methods for estimating the Optimal Cost and Acquisition Rates. The dual problem formulation eliminated Acquisition Rates as a variable, focusing solely on bid values. However, we need to accurately estimate the Acquisition Rates to distribute the wining items among the contracts, and ensure that each contract receives items in a manner that minimize the overall cost function.

Additionally, knowing the cost function's value at each sequence helps us to check the convergence of these algorithms towards the optimal solution of problem D. It also allows for a comparison between the dual and primal optimal values, to see if the duality gap is in fact zero, as was shown previously in section 3.1.

For estimating the optimal cost, we present an adaptive algorithm, which updates the cost using each new item bid. (similarly the the supply rates)

On the other hand, for estimating the Acquisition Rates, we suggest an offline algorithm that provide estimates of Acquisition Rates at any step, without knowing the previous data.

3.5.1 Stochastic approximation for Optimal Cost

The cost function of the dual problem is $\sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j) - \sum_{i=1}^{N} \rho_i C_i$. Calculating the term $\Lambda_j^*(\mu_j)$ is not possible since the price distribution is unknown. Therefore, we need to find an estimation for it.
We have:

$$\Lambda^*(\mu) = \int_0^{\mu} W(u)\mathsf{d}u = \mu W(\mu) - \int_0^{\mu} uW'(u)\mathsf{d}u$$

Where $W(x)$ denotes the cumulative distribution function (CDF) of the price distribution. i.e.,

$$W(\mu) = \mathbb{P}(p \leq \mu) = \mathbb{E}_p(\mathbb{1}[p \leq \mu]).$$

we can the rewrite the term $\int_0^{\mu} uW'(u)\mathsf{d}u$ as $\int_0^{\mu} xf_p(x)\mathsf{d}x = \int_0^{\infty} x\mathbb{1}[x \leq \mu]f_p(x)\mathsf{d}x$, where f_p is the density function of p distribution. This integral is equal to $\mathbb{E}_p(p\mathbb{1}[p \leq \mu])$. Therefore, we have:

$$\Lambda^*(\mu) = \mu\mathbb{E}_p(\mathbb{1}[p \leq \mu]) - \mathbb{E}_p(p\mathbb{1}[p \leq \mu]),$$

$$\Lambda^*(\mu) = \mathbb{E}_p[(\mu - p)\mathbb{1}[p \leq \mu]] = \mathbb{E}_p[(\mu - p)_+].$$

In the real market, we do not have access to W or Λ, and are only provided with information regarding the type of the arrived items ($\mathbb{1}_j(\theta_{n+1})$), the bid amount for the item ($\mu_j(n)$)), whether we won the bid or not ($\mathbb{1}[p_{n+1} \leq \mu(n)]$) , and in case we won the auction , the price we need to pay, which in second price actions is p_{n+1} (if unsuccessful, p_{n+1} remains unknown).
Consequently, we can determine the function $\mu_j(n) - p_{n+1}$, if we win the bid on item j. In other words, our data at each iteration is represented as:

$$[\mu_j(n) - p_{n+1}]\mathbb{1}_j(\theta_{n+1})\mathbb{1}[p_{n+1} \leq \mu_j(n)] = (\mu_j(n) - p_{n+1})_+\mathbb{1}_j(\theta_{n+1}).$$

Before proceeding further, we will introduce a proposition that will aid us in the estimating the cost.

Proposition 3.5.1 (Cost Function Convergence)**.** *Suppose p_n, $n \in \mathbb{N}$ are Independent and identically distributed random variables, where $p \in \mathcal{L}^2$.*
Additionally, suppose we have the stochastic process μ_n, where μ_n is $\mathcal{F}_n = \sigma(p_1, .., p_n)$ measurable, and $\mu_n \xrightarrow{a.s} \mu^*$.
And the additional condition is $\mathbb{E}\sum_{i=0}^{\infty}(\mu_n - \mu^*)^2 < \infty$ or $\mathbb{E}[\sum_{i=0}^{\infty}(\mu_n - \mu^*)^2|\mathcal{F}_{n-1}] < \infty$. Then:

$$\frac{\sum_{i=1}^{n}(\mu_i - p_i)_+}{n} \xrightarrow{a.s} \mathbb{E}_p[(\mu - p)_+]$$

Using this proposition, we can estimate The cost function $\sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j)$ by expressing it as:

$$\begin{aligned}
\sum_{j=1}^{M} \lambda_j \Lambda_j^*(\mu_j) &= \sum_{j=1}^{M} \lambda \mathbb{P}_{\theta_j} \Lambda_j^*(\mu_j) \\
&= \sum_{j=1}^{M} \lambda \mathbb{E}_{\theta}[\mathbb{E}_{p_j}[\mathbb{1}_j(\theta_{n+1})(\mu_j(n) - p_j(n+1))_+]] \\
&= \lambda \mathbb{E}_{\theta,p_j}[\sum_{j=1}^{M} \mathbb{1}_j(\theta_{n+1})(\mu_j(n) - p_j(n+1))_+] \\
&= \lambda \mathbb{E}_{\theta,p_j}[(\mu_{\theta_{n+1}}(n) - p(n+1))_+],
\end{aligned}$$

Were $\mathbb{P}_{\theta_j}$ represents the probability that the arriving item belongs to type j.

In the previous section, we showed that in all of the algorithms, the bids will converge almost surely to the optimal value. Leveraging this fact, and using Proposition 3.5.1, we can deduce that the empirical average of the variable $(\mu_{\theta_{n+1}}(n) - p(n+1))_+$ will converge to its expectation. Thus, we can estimate the first term of the cost function from the following update step.

$$L_{n+1} = L_n + \frac{\lambda(\mu_{\theta_{n+1}}(n) - p(n+1))_+ - L_n}{n} \tag{3.29}$$

And finally, the entire cost estimation will $L_n - \sum_{i=1}^{N} \rho_i C_i$.

3.5.2 Finding Acquisition Rates with Quadratic Programming

In this section, we discuss how to find Acquisition Rates using Quadratic Programming.

Since we have the optimal bids and corresponding supply rates $(\tilde{S}_j)$, we can estimate the Acquisition Rates. Unlike the bid and supply rates estimation, which was a online convex optimization problem, finding optimal acquisition rates is an offline problem, and do not use the previous bidding data. To find the optimal acquisition rates, we have a best estimation of supply rates, and we only need to solve a set of linear equations.

These equations for the acquisition rates will ensure that both the contract rates and the supply rates constraints are satisfied:

$$\sum_{j=1}^{M} v_{ij} R_{ij} = C_i, \quad \sum_{i=1}^{N} R_{ij} = \tilde{S}_j$$

Here we have $N+M$ equations and NM variables. Therefore, this set of linear equations can have more than one feasible solution. However, we can select the set of R_{ij}s such that the square norm of R is minimized. Furthermore, there must be a condition that all rates are positive.

Overall, the optimization problem can be formulated as follows:

$$\begin{aligned} \underset{R}{\text{minimize}} \quad & \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} R_{ij}^2}{2} \\ \text{subject to} \quad & \sum_{j=1}^{M} v_{ij} R_{ij} = C_i \\ & \sum_{i=1}^{N} R_{ij} = \tilde{S}_j, R_{ij} \geq 0. \end{aligned} \tag{$R-update$}$$

The cost function of this convex optimization problem is strongly convex with respect to R. Additionally, the set of constraints on R is nonempty (as per assumption 3.1.1). Consequently, there exists a unique solution to this optimization problem.

From observation 3.1.2, we knew that if $v_{ij} = 0$ or $v_{ij} > 0, \theta_{ij} > 0$, then we have $R_{ij} = 0$. Therefore, we can limit the set of non-zero R values to a subset on $\mathbb{R}^{NM}$, denoted as I, where $I = \{i, j | v_{ij} > 0, \theta_{ij} = 0\}$. By doing so, we restrict our optimization problem to solve only for R_{ij} where $(i, j) \in I$. This limitation of the constraints can improve algorithm efficiency. And now the algorithm has d (size of set I) variables to optimize, and usually we have $d << NM$, which will speed up the algorithm significantly.

So at each iteration of the bidding process, solving this quadratic optimization problem provides access to the acquisition rates. While this optimization problem can be solved using tools like CVXPY in Python, we introduce a simple algorithmic solution for it as well.

We define sets $S_1 = \{R_{ij} | \sum_{j=1}^{M} v_{ij} R_{ij} = C_i\}$, $S_2 = \{R_{ij} | \sum_{i=1}^{N} R_{ij} = \tilde{S}_j\}$, $S_3 = \mathbb{R}_+^{NM}$. The projection of R_{ij} on these three sets will be:

$$P_{S_1}(R_{ij}) = R_{ij} - \frac{v_{ij}(\sum_{j=1}^{M} v_{ij} R_{ij} - C_i)}{\sum_{j=1}^{M} v_{ij}^2}$$
$$P_{S_2}(R_{ij}) = R_{ij} - \frac{(\sum_{i=1}^{N} R_{ij} - \tilde{S}_j)}{N}$$
$$P_{S_3}(R_{ij}) = (R_{ij})_+$$

Proof. We will prove the correctness of the first projection, and the proof for the other projections follows similarly. First, we need to show that $P_{S_1}(R_{ij}) \in S_1$. We have:

$$\sum_{j=1}^{M} v_{ij} P_{ij} = \sum_{j=1}^{M} v_{ij} R_{ij} - \frac{\sum_{j=1}^{M} v_{ij}^2 (\sum_{j=1}^{M} v_{ij} R_{ij} - C_i)}{\sum_{j=1}^{M} v_{ij}^2} = C_i$$

So, $P_{S_1}(R_{ij})$ satisfies the constraint $\sum_{j=1}^{M} v_{ij} P_{ij} = C_i$, thus $P_{S_1}(R_{ij}) \in S_1$.
Next, we need to show that for any $c_{ij} \in S_1$, $\langle P_{ij} - R_{ij}, P_{ij} - c_{ij} \rangle \leq 0$. We have:

$$\begin{aligned}\langle P_{ij} - R_{ij}, P_{ij} - c_{ij}\rangle &= \langle \frac{-v_{ij}(\sum_{j=1}^{M} v_{ij}R_{ij} - C_i)}{\sum_{j=1}^{M} v_{ij}^2}, P_{ij} - c_{ij}\rangle \\ &= -\sum_{i=1}^{N}\sum_{j=1}^{M} \frac{v_{ij}(\sum_{j=1}^{M} v_{ij}R_{ij} - C_i)}{\sum_{j=1}^{M} v_{ij}^2}(P_{ij} - c_{ij}) \\ &= -\sum_{i=1}^{N} \frac{(\sum_{j=1}^{M} v_{ij}R_{ij} - C_i)}{\sum_{j=1}^{M} v_{ij}^2}\sum_{j=1}^{M}(v_{ij}P_{ij} - v_{ij}c_{ij}) = 0\end{aligned}$$

Therefore, $\langle P_{ij} - R_{ij}, P_{ij} - c_{ij}\rangle \leq 0$, which completes the proof. □

So the update steps will be:

$$R_{ij}(k+1) = [R_{ij}(k) - \frac{v_{ij}(\sum_{j=1}^{M} v_{ij}R_{ij}(k) - C_i)}{\sum_{j=1}^{M} v_{ij}^2} - \frac{(\sum_{i=1}^{N} R_{ij}(k) - \tilde{S}_j)}{N}]_+$$

This method is called method of alternating projections (MAP), and the sequence of alternating projections will converge to the solution of $R - update$.

Algorithm 5: Cost and Acquisition Rates estimation

for *Each Arriving Item* $\tau_{n+1}, \theta_{n+1}, p_{n+1}$ **do**

Update inter-arrival time estimates,

Place bid $x_j(n)$ observe $\mathbb{1}[p_{n+1} \leq x_j(n)]$ and update $\mu_j(n), \rho_i(n)$

Keep track of total supply targets

$S_j(n+1) = S_j(n) + a_n \mathbb{1}_j(\theta_{n+1}) \big[\frac{1}{\hat{\tau}_j(n)} \mathbb{1}[p_{n+1} \leq \mu_j(n)] - S_j(n)\big]$

Update the cost function

$L_{n+1} = L_n + a_n(\lambda \, [\mu_{\theta_{n+1}}(n) - p_{n+1}]_+ - L_n),$

$\text{cost}(n+1) = L_{n+1} - \sum_{i=1}^{N} \rho_i(n+1) C_i$

Update the Acquisition Rates

$R_{ij}^{(0)} = R_{ij}(n),$

for $k = 1 : N_0$ **do**

$R_{ij}^{(k+1)} = [R_{ij}^{(k)} - \frac{v_{ij}(\sum_{j=1}^{M} v_{ij} R_{ij}^{(k)} - C_i)}{\sum_{j=1}^{M} v_{ij}^2} - \frac{(\sum_{i=1}^{N} R_{ij}^{(k)} - S_j(n+1))}{N}]_+$

$R_{ij}(n+1) = R_{ij}^{(N_0+1)}$

Chapter 4

Conclusion

4.1 Summary

In this thesis, we addressed the problem of Real-Time Bidding (RTB) in a second-price Vickrey auction setting with multiple item types and unknown price distributions. We proposed and analyzed various algorithms to tackle this optimization problem.

First, in Chapter 2, we developed a two-time scale Projected Stochastic Approximation algorithm to solve the primal problem. This algorithm efficiently updated the Acquisition Rates while learning the bids.

In Chapter 3, we focused on solving the dual problem using three different algorithms: Penalty Term method, Projected Gradient Descent (PGD), and Subgradient Descent. We discussed the convergence conditions and convergence rates of these algorithms and confirmed their convergence through simulations and and compare their convergence rates as well.

Finally, we compared these algorithms and discussed their advantages and disadvantages. We found that PGD offers faster convergence, while Subgradient Descent provides smoother final values with less computational overhead. Notably, both PGD and Subgradient Descent surpassed the Penalty Term method in performance. Therefore, when considering computational efficiency, the choice between Projected Gradient Descent (PGD) and Subgradient Descent depends on the computational power available and the speed at which final bids need to be adapted. PGD offers faster convergence, making it suitable when computational power is not limited and rapid bid adjustments are necessary. Conversely, Subgradient Descent is preferable in scenarios where computational time per step exceeds the arrival rate of new items, as often encountered in large-scale advertising space sales.

Now, we will compare the primal and dual problems. In terms of formulation, the primal problem involves a convex optimization with NM variables and $N + M$ linear equality constraints, while the dual problem comprises $N+M$ variables with NM inequality constraints. Typically, solving the primal problem appears more straightforward due to the simplicity of equality conditions and the algorithms that can solve them. However, as discussed in Section 3.4, we demonstrated a new dual problem, where we encountered a non-differentiable convex optimization problem with no constraints and only N variables. Moreover, the variable ρ in the dual problem can be eliminated. And that's an advantage since our primary interest lies in μ, which is equivalent to optimal bids.

Another distinction lies in the focus of each problem. The primal problem centers on Acquisition Rates, beneficial for optimizing item distribution among contracts when the items are already won. In contrast, the dual problem solely concentrates on bids, which suffices when the DSP's priority lies in winning items with optimal bids, without necessarily considering efficient distribution among contracts.

Additionally, the dual problem offers the advantage of finding the optimal sets A^*, which reduce the dimensions of non-zero components in Acquisition Rates. This results in solving the optimization problem in a more compact set. Moreover, in the dual problem, the convex optimization problem is offline and do not utilize any previous data, enabling the algorithm to estimate the Acquisition Rates at any step just by knowing the supply rate estimation.

The algorithms themselves, in each case of primal and dual, exhibit differences. In the primal approach, we employed a two-time scale Projected Stochastic Approximation, whereas for the dual, we utilized Subgradient Gradient Descent. The key distinction lies in the properties of the dual problem, which allowed for a one-time scale algorithm, simplifying the process. In contrast, the two-time scale Projected Stochastic Approximation required determination of two functions for each step sequences to ensure efficient operation. Additionally, the two-time scale Projected Stochastic Approximation involved other hyperparameters such as α, which could influence the convergence of the algorithm. The dynamics of these two algorithms are different as well; in the primal problem, bids are updated on the fast time scale, while the supply rates and Acquisition Rates are on the slow time scale. However, in the dual problem, the bids and supply rates are on one timescale, while the Acquisition Rates are updated offline.

Finally, in simulations, we observed that the primal algorithm is slower and noisier. Additionally, in theory, we demonstrated that the convergence rate of the primal algorithm is on the order of $1/n^\epsilon$ (Section 2.2.2), while for the dual algorithm, it was $1/\log(n)$ (Section 3.4.1), which is faster.

4.2 Future work and related problems

In this thesis, our focus was on second-price Vickrey Auctions due to their properties that facilitated the development of our algorithms. However, Real-Time Bidding (RTB) can be analyzed in other auction settings such as first-price auctions, third-price auctions, or All-Pay Auctions [31]. Additionally, exploring bidding theory settings where other bidders have bidding strategies instead of being modeled as random variables could provide valuable insights.

Furthermore, our modeling of the RTB processes as a discrete process, where every bid is independent of the last one, may not fully capture the dynamics of real-world scenarios. Every Bid in practice influence future bids, suggesting the need for modeling the system as a continuous process, which would transform the optimization problem into a continuous optimization problem.

A challenge that may arise for demand side platform (DSP) is when many items with similar arrival rates (and same priority for contracts), arrive simultaneously. In such a scenario, the optimization algorithm may choose the cheaper item over the more expensive one to minimize the total cost function. However, the more expensive item could potentially be more valuable, as it may represent a more visited ad space, leading to higher bids from the other bidders. Losing such ads could lead to dissatisfaction among clients. To address this issue, we can extend the problem to include conditions on the priority of each item for each customer, optimizing the total number of clicks and the value of items received by each customer.

Finally, the techniques developed in this thesis can be applied to similar stochastic problems discussed in related literature [26], such as Budget Constrained Optimal Bidding, Limit Order Book Aware Markowitz Portfolio, Statistical Arbitrage Mining, and The Dark Pool Liquidation Problem. Exploring these applications could further advance our understanding and application of optimization techniques in Real-Time auction networks.

References

[1] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[2] Santiago R Balseiro, Omar Besbes, and Gabriel Y Weintraub. Repeated auctions with budgets in ad exchanges: Approximations and design. *Management Science*, 61(4):864–884, 2015.

[3] Santiago R Balseiro and Ozan Candogan. Optimal contracts for intermediaries in online advertising. *Operations Research*, 65(4):878–896, 2017.

[4] Shalabh Bhatnagar, N Hemachandra, and Vivek Kumar Mishra. Stochastic approximation algorithms for constrained optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, 21(3):15, 2011.

[5] Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency (India), P 19 Green Park Extension, New Delhi 110 016, India, 2008.

[6] Vivek S Borkar and Sean P Meyn. The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.

[7] Jonathan M. Borwein and Adrian S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer-Verlag, New York, 2 edition, 2006.

[8] Jonathan M. Borwein and Jon D. Vanderwerff. *Convex Functions*. Cambridge University Press, 2010.

[9] Stephen Boyd, Almir Mutapcic, and John Duchi. Stochastic subgradient methods. Notes for EE364b, Stanford University, April 12 2018.

[10] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[11] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 661–670, 2017.

[12] Bowei Chen, Shuai Yuan, and Jun Wang. A dynamic pricing model for unifying programmatic guarantee and real-time bidding in display advertising. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9, 2014.

[13] Hana Choi, Carl F. Mela, Santiago R. Balseiro, and Adam Leary. Online display advertising markets: A literature review and future directions. *Information Systems Research*, 31(2):556–575, 2020. Published Online: June 09, 2020.

[14] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[15] Simon S. Du and Wei Hu. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. *arXiv preprint arXiv:1802.01504*, 2018. Focus to learn more.

[16] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2005. Focus to learn more.

[17] Michael P Friedlander and Paul Tseng. Exact regularization of convex programs. *SIAM Journal on Optimization*, 18(4):1326–1350, 2008.

[18] Aritra Ghosh, Saayan Mitra, Somdeb Sarkhel, Jason Xie, Gang Wu, and Viswanathan Swaminathan. Scalable bid landscape forecasting in real-time bidding. *arXiv preprint arXiv:2001.06587*, 2020.

[19] Larry Goldstein. On the choice of step size in the robbins-monro procedure. *Statistics & probability letters*, 6(5):299–303, 1988.

[20] Nicolas Grislain, Nicolas Perrin, and Antoine Thabault. Recurrent neural networks for stochastic control in real-time bidding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2801–2809, 2019.

[21] A. Izmailov and M. Solodov. Convergence rate estimates for penalty methods revisited. *Computational Optimization and Applications*, 85:1–20, 2023.

[22] Chong Jiang, Carolyn L Beck, and Rayadurgam Srikant. Bidding with limited statistical knowledge in online auctions. *ACM SIGMETRICS Performance Evaluation Review*, 41(4):38–41, 2014.

[23] Niklas Karlsson. Control problems in online advertising and benefits of randomized bidding strategies. *European Journal of Control*, 30:31–49, 2016.

[24] Ryan J Kinnear, Ravi R Mazumdar, and Peter Marbach. Optimal real-time bidding policies for contract fulfillment in second price auctions. *arXiv preprint arXiv:2012.10001*, 12 2020.

[25] Ryan J Kinnear, Ravi R Mazumdar, and Peter Marbach. Real-time bidding for time constrained impression contracts in first and second price auctions-theory and algorithms. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(3):1–37, 12 2021.

[26] Ryan J Kinnear, Ravi R Mazumdar, and Peter Marbach. Convexity and duality in optimum real-time bidding and related problems. *arXiv preprint arXiv:2206.12408*, 6 2022.

[27] Harold J. Kushner and Dean S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, volume 26 of *Applied Mathematical Sciences*. Springer, 1978.

[28] Harold J Kushner and G George Yin. *Stochastic Approximation and Recursive Algorithms and Applications*, volume 35 of *Stochastic Modelling and Applied Probability*. Springer, 2003. Access provided by UNIVERSITY OF WATERLOO.

[29] Guanghui Lan and Zhiqiang Zhou. Algorithms for stochastic optimization with functional or expectation constraints. *arXiv preprint arXiv:1604.03887*, 2016. Submitted on 13 Apr 2016 (v1), last revised 2 Oct 2020 (v8).

[30] Yishay Mansour, S. Muthukrishnan, and Noam Nisan. Doubleclick ad exchange auction. *arXiv preprint arXiv:1204.0535*, 2012.

[31] Michael Maschler, Eilon Solan, and Shmuel Zamir. *Game Theory*. Cambridge University Press, 2013.

[32] Arunselvan Ramaswamy and Shalabh Bhatnagar. A generalization of the borkar-meyn theorem for stochastic recursive inclusions. *Mathematics of Operations Research*, 42(3):648–661, 2017.

[33] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, 22(3):400–407, 1951.

[34] Vladislav B Tadic, Arnaud Doucet, and Sumeetpal Singh. Two time-scale stochastic approximation for constrained stochastic optimization and constrained markov decision problems. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 6, pages 4736–4741, Denver, CO, USA, 2003. IEEE.

[35] Erik Tillberg, Peter Marbach, and Ravi Mazumdar. Optimal bidding strategies for online ad auctions with overlapping targeting criteria. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–55, 2020.

[36] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[37] M. Vidyasagar. Convergence of stochastic approximation via martingale and converse lyapunov methods. *Mathematical Control Signals and Systems*, 35:351–374, 2023.

[38] I-J Wang and James C Spall. Stochastic optimization with inequality constraints using simultaneous perturbations and penalty functions. In *Proceedings of the 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 4, pages 3808–3813, Maui, HI, USA, 2003. IEEE.

[39] I-Jeng Wang and James C Spall. A constrained simultaneous perturbation stochastic approximation algorithm based on penalty functions. In *Proceedings of the 1998 IEEE ISIC/CIRNISA Joint Conference*, Gaithersburg, MD, September 14-17 1998. IEEE.

[40] Jun Wang, Weinan Zhang, and Shuai Yuan. Display advertising with real-time bidding (rtb) and behavioural targeting. *Foundations and Trends® in Information Retrieval*, 11(4-5):297–435, 2017.

[41] Erfan Yazdandoost Hamedani and Necdet Serhat Aybat. A primal-dual algorithm with line search for general convex-concave saddle point problems. *arXiv preprint arXiv:1803.01401*, 2018. Focus to learn more.

[42] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, pages 1–8, 2013.

[43] Weinan Zhang, Yifei Rong, Jun Wang, Tianchi Zhu, and Xiaofan Wang. Feedback control of real-time display advertising. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 407–416, New York, NY, United States, 2016. Association for Computing Machinery.

[44] Weinan Zhang and Jun Wang. Statistical arbitrage mining for display advertising. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1465–1474, New York, NY, United States, 2015. Association for Computing Machinery.

[45] Weinan Zhang, Shuai Yuan, and Jun Wang. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1077–1086, 2014.

[46] Weinan Zhang, Shuai Yuan, Jun Wang, and Xuehua Shen. Real-time bidding benchmarking with ipinyou dataset. *arXiv preprint arXiv:1407.7073*, 2014.

APPENDICES

Appendix A

Some Useful Theorems

In this appendix we will mention theorems that we used to proof the convergence of the algorithms in the chapter 2 and 3. The proof of these theorems are all showed in the refrences that has been produced.

A.1 Stochastic Approximation

we summarize some basic results on stochastic approximation, derived from [5, 37].

Theorem A.1.1 (Stochastic Approximation [5, 37])**.** *Consider the random sequence, called a stochastic approximation, beginning with an arbitrary* $x_0 \in \mathbb{R}^d$

$$x_{n+1} = x_n + a_n[h(x_n) + M_{n+1}],\ n \in \mathbb{Z}_+, \tag{A.1}$$

where $a_n \in \mathbb{R}$ *is a deterministic sequence, and* M_n *is a random sequence. Let* $\mathcal{F}_n = \sigma(M_1, M_2, \ldots, M_n)$ *be the* σ*-algebra generated by the* M_n*. Assume the following conditions*

1. *the equation* $h(x) = 0$ *has the unique solution* x^*
2. $h : \mathbb{R}^d \to \mathbb{R}^d$ *is globally Lipschitz-continuous with constant* L*.*
3. *(Robbins-Monro)* $\sum_{n=0}^{\infty} a_n = \infty, \sum_{n=0}^{\infty} a_n^2 < \infty$*.*
4. M_n *is a martingale difference sequence,* i.e., $\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] = 0$*, and square-integrable, i.e.,* $\mathbb{E}[||M_{n+1}||^2 \mid \mathcal{F}_n] \leq \sigma^2(1 + ||x_n - x^*||^2)$ *a.s.*

5. *(Globally Asymptotically Stable (GAS) equilibrium) There exists a* $\mathbb{C}^2$ *Lyapunov function* $V : \mathbb{R}^d \rightarrow \mathbb{R}_+$ *for the ODE* $\dot{x} = h(x)$*, that satisfies the following conditions:*

   - *There exists constant* $a > 0$ *such that* $\forall x \in \mathbb{R}^d$ *,* $V(x) = a\|x - x^*\|^2$.
   - *There exists a function* $\phi \in class\mathcal{B}$ *(*$\phi : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ *and* $\phi(x)$ *attains the value of zero exclusively at* $x = 0$*) such that* $\dot{V} \leq -\phi(\|x - x^*\|)$.

Then, $x_n \overset{a.s.}{\rightarrow} x^\star$ *as* $n \rightarrow \infty$.

Theorem A.1.2 (Borkar-Meyn [6]). *Consider a stochastic approximation with limiting ODE* $\dot{x} = h(x)$ *which satisfies all the first 4 assumptions of Theorem A.1.1 and let* $h_c(x) = \frac{1}{c}h(cx)$*. If there exists a Lipschitz continuous function* $h_\infty : \mathbb{R}^d \rightarrow \mathbb{R}^d$ *such that* $h_c \rightarrow h_\infty$ *as* $c \rightarrow \infty$*, uniformly on compacts, and the ODE* $\dot{x} = h_\infty(x)$ *has the origin as its unique globally asymptotically stable equilibrium, then the associated stochastic approximation will converge.*

A.1.1 Projected Stochastic Approximation

In section 3.3 we used the Projected Gradient Descent algorithm to solve the dual problem. The following theorem shows that this algorithm will converge if the set of inequalities G, be a convex nonempty and closed set. It has been proven that with some extra assumptions on G we can show that the Projected Stochastic Approximation will converge to a feasuble point in G.

Theorem A.1.3 (Projected Stochastic Approximation [5, 27]). *Consider the random sequence* $x_n \in \mathbb{R}^d$*, starting at arbitrary point* $x_0 \in \mathbb{R}^d$ *and generated by,*

$$\begin{aligned} \tilde{x}_{n+1} &= x_n + a_n[h(x_n) + M_{n+1}], \\ x_{n+1} &= \Pi_G(\tilde{x}_{n+1}),\ n \in \mathbb{Z}_+, \end{aligned} \tag{A.2}$$

Where G *is a nonempty convex set and* $\Pi_G(x)$ *is the projection of point* x *on set* G*. (*G *need to be convex for the projection to have a unique solution) This sequece will converge as* $n \rightarrow \infty$*, if we have the following conditions holds.*

1. G *must be a convex closed nonempty set and every point in* G *can be described as* $G = \{x \in \mathbb{R}^d \mid g_i(x) \leq 0, i = 1, ..., s\}$*, where* $g_i : \mathbb{R}^d \rightarrow \mathbb{R}$ *are continuous1y differentiab1e functions.*

2. *at each* $x \in \partial G$ $(G \setminus intG)$*,the gradients of the active constraints are linearly independent.*

3. $h : \mathbb{R}^d \to \mathbb{R}^d$ *is globally Lipschitz-continuous with constant* L*.*

4. $h : \mathbb{R}^d \to \mathbb{R}^d$ *can be writen as* $h = -\nabla_x f(x)$*, where* $f : \mathbb{R}^d \to \mathbb{R}$ *is a is a continuously differentiable function.*

5. *(Robbins-Monro)* $\sum_{n=0}^{\infty} a_n = \infty, \sum_{n=0}^{\infty} a_n^2 < \infty$*.*

6. M_n *is a martingale difference sequence,* i.e., $\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] = 0$*, and square-integrable, i.e.,* $\mathbb{E}[||M_{n+1}||^2 \mid \mathcal{F}_n] \leq \sigma^2(1 + ||x_n - x^*||^2)$ *a.s.*

7. *support of the conditional distribution of* M_{n+1} *given* $\mathcal{F}_n$ *is a closed bounded set* $A(x_n)$ *depending on* x_n*.*

Moreover x_n *will converge to a point which satisfies the Kuhn-Tucker necessary condition for a constrained minimum. where the Kuhn-Tucker points is*

$$\text{KT} = \{x \in \mathbb{R}^d \mid \exists \lambda_i \leq 0, i = 1, ..., s, \text{ such that, } \lambda_i g_i(x) = 0, -h(x) + \sum_{i=1}^{s} \nabla_x g_i(x) = 0\}.$$

For every point $x \in G$ *and vector* $V(x)$ *we can define a function denote a vector field on* G*,*

$$\bar{\Pi}(V(x)) = \lim_{\delta \to 0^+} (\frac{\Pi_G(x + \delta V(x)) - x}{\delta}).$$

By the definition one can see that, for $x \in \text{int}G$*,* $\bar{\Pi}(V(x)) = 0$*. In case of* G *being a convex closed set the limit has a unique solution and* $\bar{\Pi}(V(x))$ *has a finite value for every* $x \in \partial G$*. Finally we have that the corresponding ODE to this sequence is*

$$\dot{x} = \bar{\Pi}(h(x))$$

$$\dot{f}(x) = \langle \nabla f(x) \, , \bar{\Pi}(-\nabla f(x)) \rangle$$

A.1.2 Multiple Timescales and Asynchronous Updates

The stochastic approximation algorithm used in algorithm 1 is more general case of Theorem A.1.1, as there are two separate step-size sequences a_n, b_n, and only components corresponding to the type of the arriving item are updated. Thus, we propose the utilization of asynchronous two timescale Stochastic Approximation.

Theorem A.1.4 (Two Timescale Stochastic Approximation [5])**.** *Consider two random sequence, $x_n \in \mathbb{R}^d, y_n \in \mathbb{R}^k$, called two timescale Stochastic Approximation, with iterations*

$$x_{n+1} = x_n + a_n[h(x_n, y_n) + M^1_{n+1}], \tag{A.3}$$

$$y_{n+1} = y_n + b_n[g(x_n, y_n) + M^2_{n+1}], \tag{A.4}$$

where h, g are Lipschitz continuous functions, the deterministic sequences a_n, b_n both satisfy the Robbins-Monro conditions with $b_n/a_n \to 0$. Suppose that M^1_n, M^2_n are square integrable martingale difference sequences (w.r.t. the σ-algebras generated by the history of all M^1_n, M^2_n) in the sense that $\mathbb{E}[M^i_{n+1} \mid \mathcal{F}_n] = 0$ and $\mathbb{E}[||M^i_{n+1}||^2_2 \mid \mathcal{F}_n] \leq K(1 + ||x(n)||^2 + ||y(n)||^2)$ for some $K \in \mathbb{R}$. Moreover, assume the following conditions:

1. *the ODE $\dot{x}(t) = h(x(t), y)$ has a GAS equilibrium $\lambda(y)$, where $\lambda : \mathbb{R}^k \to \mathbb{R}^k$ is Lipschitz.*

2. *the ODE $\dot{y}(t) = h(\lambda(y), y(t))$ has a GAS equilibrium y^**

Then, $x_n \overset{a.s.}{\to} x^\star$ and $y_n \overset{a.s.}{\to} y^\star$ as $n \to \infty$.

Proposition A.1.1 (Two Timescale Asynchronous Stochastic Approximation)**.** *Consider a two-timescale asynchronous stochastic approximation*

$$x_j(n+1) = x_j(n) + a_n \mathbb{1}_j(\theta_{n+1})[h_j(x(n), y(n)) + M^j_{n+1}],$$
$$y_i(n+1) = y_i(n) + b_n \mathbb{1}_i(\phi_{n+1})[g_i(x(n), y(n)) + N^i_{n+1}],$$

where $i \in [N], j \in [M], \theta_n \in [M], \phi_n \in [N]$ are drawn i.i.d. *from categorical distributions $\mathbb{P}\{\theta_n = j\} = \eta_j > 0$, $\mathbb{P}\{\phi_n = i\} = p_i > 0$. Suppose that these iterations satisfy all the conditions in Theorem A.1.4. then x, y will converge a.s*

Proof. We can re-write the algorithm as

$$x_j(n+1) = x_j(n) + a_n[\eta_j h_j(x(n), y(n)) + \widetilde{M}^j_{n+1}] \; j \in [M]$$
$$y_i(n+1) = y_i(n) + b_n[p_i g_i(x(n), y(n)) + \widetilde{N}^i_{n+1}] \; i \in [N],$$

where

$$\widetilde{M}^j_{n+1} = \mathbb{1}_j(\theta_{n+1}) M^j_{n+1} + \big(\mathbb{1}_j(\theta_{n+1}) - \eta_j\big) h_j(x(n), y(n))$$
$$\widetilde{N}^i_{n+1} = \mathbb{1}_i(\phi_{n+1}) N^i_{n+1} + \big(\mathbb{1}_i(\phi_{n+1}) - p_i\big) g_i(x(n), y(n)).$$

Since θ_n, ϕ_n are drawn independently, we have

$$\mathbb{E}[\big(\mathbb{1}_j(\theta_{n+1}) - \eta_j\big)h_j(x(n), y(n)) \mid \mathcal{F}_n] \tag{A.5}$$
$$= h_j(x(n), y(n))\mathbb{E}[\mathbb{1}_j(\theta_{n+1}) - \eta_j] = 0, \tag{A.6}$$

and similarly for ϕ_n. As well,

$$\mathbb{E}[||\widetilde{M}_{n+1}||_2^2| \ \mathcal{F}_n] = \sum_{j=1}^{M}\Big(\mathbb{E}[\mathbb{1}_j(\theta_{n+1})^2(M_{n+1}^j)^2 \mid \mathcal{F}_n] \tag{A.7}$$
$$+ \mathbb{E}[(\mathbb{1}_j(\theta_{n+1}) - \eta_j)^2 h_j(x(n), y(n))^2|\mathcal{F}_n]\Big) \tag{A.8}$$
$$\leq \sum_{j=1}^{M}\Big(\mathbb{E}[(M_{n+1}^j)^2 + h_j(x(n), y(n))^2|\mathcal{F}_n]\Big) \tag{A.9}$$
$$\overset{(a)}{\leq} (K + L)(1 + ||x(n)||_2^2 + ||y(n)||_2^2) \tag{A.10}$$

where (a) is uses the Lipschitz constant L of h. As well, $\mathbb{E}[\widetilde{M}_{n+1} \mid \mathcal{F}_n] = 0$. The conclusion is now a direct consequence of Theorem A.1.4 □

A.2 Duality

In the proof of Proposition 3.3.1, we rely on the equivalence between the optima solutions of problem (D) and KT points. This equivalence is supported by the following theorem:

Theorem A.2.1 (Karush-Kuhn-Tucker Conditions[10]). *We assume that Consider the optimization problem (P) to be*

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leqslant 0, \ \forall i \in \{1, \ldots, m\} \end{aligned} \tag{A.11}$$

where $f, g_1, \ldots, g_s$ are convex functions from $\mathbb{R}^d$ to $\mathbb{R}$.
Suppose that there exists a Slater point, i.e.:

$$(\exists y \in \mathbb{R}^d)(\forall i \in \{1, \ldots, m\}) \quad g_i(y) < 0.$$

Then there exists $x \in \mathbb{R}^d$ *which solves* (P) *if and only if there exists* $\lambda \in \mathbb{R}^m$ *such that the following hold:*

$$(\forall i \in \{1, \ldots, m\}) \quad g_i(x) \leqslant 0 \tag{A.12}$$

$$(\forall i \in \{1, \ldots, m\}) \quad \lambda_i \geqslant 0 \tag{A.13}$$

$$0 \in \partial f(x) + \sum_{i=1}^{m} \lambda_i \partial g_i(x) \tag{A.14}$$

$$(\forall i \in \{1, \ldots, m\}) \quad \lambda_i g_i(x) = 0 \tag{A.15}$$

Appendix B

Python Codes

This section provides concise listings of select Python programs used in core parts of the thesis. Various non-essential pieces have been omitted for brevity.

B.1 Projected Gradient Descent Method For Dual Problem

The following Python code represents the projection onto the constraint set used in Algorithm 3:

```
coefficients = []
for i in range(N):
    for j in range(M):
        l1 = [0]*(M+N)
        l1[j] = -1
        l1[M+i] = V[i,j]
        coefficients.append(l1)
A = np.array(coefficients)

def project_onto_convex_set(A, N, M, y):
    x = cp.Variable(N+M)
    objective = cp.Minimize(cp.norm(y - x))
```

```
    constraints = [A @ x <= 0]
    # Create the problem instance and solve it
    problem = cp.Problem(objective, constraints)
    problem.solve()
    # Return the projected vector
    projected_vector = x.value
    return projected_vector
```

B.2 Subgradient Method For Dual Problem

Here's the Python code for a Two-Time Scale Subgradient Descent algorithm tailored for the special case when $V \in \{0, 1\}$ and the arrival rates are known:

```
def GD(supply,avtimes,ru,mu,C,V,arrival_rates,arv_times,
arv_types,arr_prices,ans,bns,alpha,cost1,cost_tot):
    #update the cost function
    lamda_t=sum(arrival_rates)
    j = arv_types
    avtimes_new[j] = avtimes[j] + ans * (arv_times - avtimes[j])
    if arr_prices < mu_new[j]:
        cost1_new = cost1-bns*(sum(arrival_rates)*(mu_new[j]-arr_prices)+cost1)
    else:
        cost1_new = cost1 + bns *(0-cost1)
    cost_tot_new = cost1_new
    for i in range(N):
        cost_tot_new = cost_tot_new + C[i] * ru[i]
    # update supply rates
    if arr_prices < mu_new[j]:
        supply_new2[j] =supply_new[j]+ans*(arrival_rates[j]*1-supply_new[j])
    else:
        supply_new2[j] =supply_new[j]+ans*(arrival_rates[j]*0-supply_new[j])
    #update mu
    for j in range(M):
        mu_new2[j] = mu_new[j] - bns * (supply_new2[j])
    for i in range(N):
```

```
        allmu = []
        for j in range(M):
            if V[i, j] == 1:
                allmu.append(mu_new[j])
        g = mu_new * 0
        for j in range(M):
            if V[i, j] == 1:
                if mu_new[j] == min(allmu):
                    g[j] = 1
        mu_new2 = mu_new2 + bns * C[i] * g / np.sum(g)
    #update ru
    for i in range(N):
        muall = []
        for j in range(M):
            if V[i, j] == 1:
                muall.append(mu_new2[j])
        ru_new2[i] = min(muall)
    return avtimes_new, supply_new2, ru_new2, mu_new2, cost1_new, cost_tot_new
```

B.3 Finding Acquisition Rates

The following Python code is for is algorithm 5 to find Acquisition Rates, where supply rates are known.

```
def project_S1(R, C, s, V):
    R_new = np.zeros((N, M))
    for i in range(N):
        for j in range(M):
            R_new[i,j] = R[i,j]- (np.sum(R, axis=0)[j] - s[j]) /N
            - V[i,j] *(np.sum(V*R,axis=1)[i] -C[i]) /np.sum(V*V, axis=1)[i]
    return R_new

def projection(R, C, s, V):
    R_new = np.zeros((N, M))
    R_new = project_S1(R, C, s, V)
```

```python
    for i in range(N):
        for j in range(M):
            R_new[i, j] = max(R_new[i, j], 0)
    return R_new

```