Online Learning Algorithms for Stochastic Inventory and Queueing Systems

by

Weidong Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in The University of Michigan
2019

Doctoral Committee:

Professor Izak Duenyas, Co-Chair
Assistant Professor Cong Shi, Co-Chair
Associate Professor Stefanus Jasin
Assistant Professor Viswanath Nagarajan

Weidong Chen

aschenwd@umich.edu

ORCID iD: 0000-0001-5633-7970

I would like to dedicate my Ph.D. thesis to my beloved parents, my father who always shares his wisdom, and my mother who always takes great care of me.

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisors Professor Cong Shi and Professor Izak Duenyas. I started working with them during my master's program; the work was very interesting, but I had never thought about becoming a Ph.D. student. Professor Shi and Professor Duenyas encouraged me to pursue a Ph.D. degree by sharing lots of their thoughts and experiences and would not hesitate to clear my doubt and raise my confidence level. During my Ph.D. journey, they were my mentors not only on research but also on career and life. Without their help and support, this dissertation would not have been possible. They together combined a perfect mix of working styles and personalities that I genuinely appreciate.

I would also like to thank Professor Viswanath Nagarajan and Professor Stefanus Jasin to be on my Ph.D. committee, and I really appreciate their advice and opinions. My gratitude also goes to Professor Katta Murty, and other professors who shared their wisdom with me, and the wonderful staff members in the department for their assistance and help.

I appreciate the friendship with my office mates Amirhossein Meisami, Nima Salehi Sadghiani, Abdullah Alshelahi (who as well brought plenty of Middle East culture into the office), my colleagues Sentao Miao, Qiyun Pan, Qi He for their help with courses and research, and also Professor Shi's research group members Huanan Zhang, Yuchen Jiang, Hao Yuan for sharing research and career advice.

Finally, I would like to thank my friends, who brought so much joy into my life, especially my roommates Chencheng Zhou, Xinyi Ge, Hao Wu, and my working friends Kaixin Wang, Boyang Wang, Duyi Li, Li Ding.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

The management of inventory and queueing systems lies in the heart of operations research and plays a vital role in many business enterprises. To this date, the majority of work in the literature has been done under complete distributional information about the uncertainties inherent in the system. However, in practice, the decision maker may not know the exact distributions of these uncertainties (such as demand, capacity, lead time) at the beginning of the planning horizon, but can only rely on realized observations collected over time. This thesis focuses on the interplay between learning and optimization of three canonical inventory and queueing systems, and proposes a series of *first* online learning algorithms.

The first system studied in Chapter II is the periodic-review multiproduct inventory system with a warehouse-capacity constraint. The second system studied in Chapter III is the periodic-review inventory system with random capacities. The third system studied in Chapter IV is the continuous-review make-to-stock $M/G/1$ queueing system. We take a *nonparametric* approach that directly works with data and needs not to specify any (parametric) form of the uncertainties. The proposed online learning algorithms are stochastic gradient descent type, leveraging the (sometimes non-obvious) convexity properties in the objective functions. The performance measure used is the notion of cumulative regret or simply regret, which is defined as the cost difference between the proposed learning algorithm and the clairvoyant optimal algorithm (had all the distributional information about uncertainties been

given). Our main theoretical results are to establish the square-root regret rate for each proposed algorithm, which is known to be tight. Our numerical results also confirm the efficacy of the proposed learning algorithms.

The major challenges in designing effective learning algorithms for such systems and analyzing them are as follows. First, in most retail settings, customers typically walk away in the face of stock-out, and therefore the system is unable to keep track of these lost-sales. Thus, the observable demand data is, in fact, the sales data, which is also known as the *censored demand* data. Second, the inventory decisions may impact the cost function over extended periods, due to *complex state transitions* in the underlying stochastic inventory system. Third, the stochastic inventory system has *hard physical constraints*, e.g., positive inventory carry-over, warehouse capacity constraint, ordering/production capacity constraint, and these constraints limit the search space in a dynamic way.

We believe this line of research is well aligned with the important opportunity that now exists to advance *data-driven algorithmic decision-making under uncertainty*. Moreover, it adds an important dimension to the general theory of online learning and reinforcement learning, since firms often face a realistic stochastic supply chain system where system dynamics are complex, constraints are abundant, and information about uncertainties in the system is typically censored. It is, therefore, important to analyze the structure of the underlying system more closely and devise an efficient and effective learning algorithm that can generate better data, which is then feedback to the algorithm to make better decisions. This forms a virtuous cycle.

CHAPTER I

Introduction

Supply chain management concerns the efficient allocation and control of raw materials, finished products, and customer services. It plays a vital role in any successful business enterprise. The 2017 Annual State of Logistics Report shows that the total U.S. business logistics cost is 1.48 trillion, accounting for more than 7.7% of the U.S. gross domestic product (GDP). Among the decisions in supply chain management, inventory control is the first of mind and often the most critical component for any wholesale business. Indeed, the idea of inventory control not only applies to products in the warehouse but also to seats on airplanes, beds in hospitals, drivers for ride-sharing companies, and so on. The goal for inventory control is to strike an optimal balance between under-stocking and over-stocking, i.e., maintaining a sufficient amount of inventory to fulfill customer demand while avoiding excess inventory taking up space in case of expiration, damage, or fund flow related problems. The key challenge lies in how to buffer the uncertainty of future evolution appropriately. Often firms find it hard to forecast the future demand, the unexpected interruption in the production phase, as well as the order or shipping lead time. Moreover, given nowadays complex business environment, firms often need to consider other important factors such as product correlations, strategic customers, and financial risks, when seeking the optimal policy.

Many of the theoretical optimization models in inventory and queueing control aim to capture the complexity of making decisions under uncertainty. In conventional models, the uncertainty about future evolution is usually defined through explicitly specified probability distributions or stochastic processes, which are treated as input data to respective optimization models. However, in most real-life applications, the true underlying distributions are not available or they are too complex to work with. Often, our knowledge is restricted to historical data, simulated data, or information from forecasting and market analysis. The objective of this thesis is to develop efficient and effective algorithms for sequential decision-making problems arising in the context of inventory and queueing control where the input data of the problems are unknown or uncertain at the beginning of the decision period. We aim to provide decision tools for decision-makers to better cope with uncertainty in these stochastic systems by absorbing, analyzing and utilizing data in an online fashion, which can be viewed as a substantial step to meet the challenges presented by the era of Big Data.

To achieve our goals, we will develop efficient and effective nonparametric learning algorithms that can simultaneously learn the input uncertainty in the underlying optimization problems as well as optimize the system-wide objective value on the fly. The algorithms compute policies based only on past observable data in an online manner. One major challenge in constructing such algorithms is that, in practice, the data or samples collected are often censored or inaccurate. For example, firms cannot typically observe their lost-sales since customers simply walk away when they find their desired items out of stock. As a result, the sales data collected are not true samples of demand, and the algorithmic design needs to correct such estimation biases in the long run. In our algorithmic framework, we take a non-parametric approach by not enforcing any parametric assumption on the underlying distributions. Our performance measure is regret-based, which quantifies the difference in objective values between our nonparametric sampling-based policy and the clairvoyant optimal

policy that has access to the true underlying distribution a priori. We will derive both theoretical performance guarantees as well as practical implementation strategies in this thesis. From the methodological point of view, the study of the algorithms will advance the understanding of the tradeoffs between learning and earning in the context of inventory and queueing systems, and the analysis will establish important connections with the general theory of online learning (which typically does not deal with inventory constraints and complex system dynamics).

1.1 Contributions of the Thesis

We study three different stochastic systems. We assume that the firm has no prior distributional information about the uncertainty, and must learn from past data. Our objective is to propose learning algorithms that admit provably tight regret.

In Chapter 2, we propose a nonparametric data-driven algorithm called DDM for the management of stochastic periodic-review multi-product inventory systems with a warehouse-capacity constraint. The demand distribution is not known a priori and the firm only has access to censored demand data. We measure the performance of DDM through regret, the difference between the total expected cost of DDM and that of an oracle with access to the true demand distribution acting optimally. We characterize the rate of convergence guarantee of DDM. More specifically, we show that the average expected T-period cost incurred under DDM converges to the optimal cost at the rate of $O(1/\sqrt{T})$. We also discuss several extensions and conduct numerical experiments to demonstrate the effectiveness of our proposed algorithm.

In Chapter 3, we propose the first nonparametric learning algorithm for single-product, periodic-review, backlogging inventory systems with random production capacity. Different than the current literature on this class of problems, we assume that the firm has neither prior information about the demand distribution nor the capacity distribution and only has access to past demand and supply data (which can be

referred to as censored capacity information). If both the demand and capacity distributions are known at the beginning of the planning horizon, it is well-known that modified base-stock policies are optimal. When such distributional information is not available *a priori* to the firm, we propose a cyclic gradient-descent type of algorithm whose running average cost asymptotically converges to the clairvoyant optimal cost, where the clairvoyant optimal cost corresponds to the case where the firm knows the demand and capacity distributions and applies the optimal policy. We prove that the rate of convergence guarantee of our algorithm is $O(1/\sqrt{T})$, which is theoretically the best possible for this class of problems. We also conduct numerical experiments to demonstrate the effectiveness of our proposed algorithms.

In Chapter 4, we consider a canonical $M/G/1$ make-to-stock queueing system that arises in many practical settings. The decision maker has no prior knowledge about the rate of the Poisson arrival process and the distribution of the production/service time, which must be learned over time from past observations. We propose a stochastic gradient descent algorithm and prove that its average expected cost converges to the clairvoyant optimal cost (had the arrival and service distributions been given) at a square-root convergence rate, which is provably tight for this class of problems. We also conduct numerical experiments to demonstrate the effectiveness of our proposed algorithms.

CHAPTER II

Nonparametric Algorithms for Multiproduct Inventory Systems

2.1 Introduction

The study of stochastic multi-product inventory systems dates back to *Veinott* (1965). Most, if not all, of the papers on stochastic multi-product inventory systems assume that the stochastic future demand is given by a specific exogeneous random variable, and the inventory decisions are made with full knowledge of the future demand distribution. However, in practice, the demand distribution is usually not known a priori. Even with past demand data (often censored) collected, the selection of the most appropriate distribution and its parameters remains difficult (see *Huh and Rusmevichientong* (2009), *Huh et al.* (2011), *Besbes and Muharremoglu* (2013) for more discussions on censored demand in inventory systems).

Model overview and research issue. In our periodic-review multi-product lost-sales inventory system over a finite horizon of T periods, the demands across periods $t = 1, \dots T$ are (i.i.d.) random vectors $\mathbf{D}_t$ (with each component representing a different product), respectively. There is a joint warehouse-capacity constraint M imposed on the total number of products that can be held in inventory. The firm has no access to the true underlying demand distribution a priori, and can only observe

sales data (i.e., censored demand) over time. We develop a nonparametric data-driven adaptive inventory control policy $\pi = (\mathbf{y}_t \mid t \geq 1)$ where the decision $\mathbf{y}_t$ represents the order-up-to level in period t. We measure performance of our proposed policy π through regret denoted by $\mathcal{R}_T \triangleq \mathcal{C}(\pi) - \mathcal{C}(\pi^*)$, where $\mathcal{C}(\pi)$ is the total expected cost of π and $\mathcal{C}(\pi^*)$ is the total expected cost of a *clairvoyant* optimal policy π^* with access to the true underlying demand distribution a priori. The research question is to devise an effective nonparametric data-driven policy π that drives the average regret $\mathcal{R}_T/T$ to zero with a fast convergence rate.

Main results and contributions. We propose a nonparametric data-driven algorithm called DDM for stochastic multi-product inventory systems with a warehouse-capacity constraint. We characterize the rate of convergence guarantee of DDM. More specifically, we show that the average regret $\mathcal{R}_T$ converges to zero at the rate of $O(1/\sqrt{T})$. Our algorithm DDM is a stochastic gradient descent type of algorithm, similar in spirit to *Burnetas and Smith* (2000), *Kunnumkal and Topaloglu* (2008) and *Huh and Rusmevichientong* (2009). The work closest to ours is *Huh and Rusmevichientong* (2009) who studied an uncapacitated inventory system with a single product. The novelty of our work lies in both algorithmic design and performance analysis of DDM. First, unlike the uncapacitated single-product case, the gradient estimator in DDM could be sometimes indeterminable in the presence of a warehouse-capacity constraint on multiple products. Second, the projection step in DDM has to factor in both positive inventory carry-over of all products and the warehouse-capacity constraint. To maintain feasibility of the solution in each step, we solve two additional optimization problems. The optimization problems can be efficiently solved by greedy algorithms, but the solution structure makes the asymptotic performance analysis invariably harder than that in the uncapacitated single-product case (where no optimization procedures are needed). The key technical challenge in our analysis is to derive an upper bound of the distance between the target order-up-to level and

the actual implemented order-up-to level (due to the warehouse-capacity constraint and positive inventory carry-over from previous periods). Note that the upper bound on this distance function is almost immediate in the uncapacitated single-product case while the development of an upper bound is significantly more complex in our multi-product setting. Third, we relate the inventory process to a $GI/G/1$ queue. We then develop a stochastic dominance argument and invoke a classical result on the expected busy period in $GI/G/1$ queue due to *Loulou* (1978).

We compare the computational performance of DDM with several existing parametric and nonparametric approaches in the literature. Our results show that DDM outperforms these benchmark algorithms in terms of both consistency and convergence rate. We also consider two interesting extensions, one with a more general warehouse-capacity constraint where different products may have different dimension or sizes, and the other one with discrete demand and order quantities.

Our work is relevant to the following research streams.

Multi-product stochastic inventory systems. There is a large body of literature devoted to various classes of such problems. In this chapter, we focus our attention on the classical stochastic multi-product inventory systems under a warehouse-capacity constraint, first studied by *Veinott* (1965). He provided conditions that ensure that the base-stock ordering policy is optimal in a periodic-review inventory system with a finite horizon. Subsequently, *Ignall and Veinott* (1969) showed that in the stationary demand case, a myopic ordering policy is optimal under certain mild conditions. *Beyer et al.* (2001, 2002) established the optimality of myopic policies in backlogged systems with separable costs by appealing to the sufficient condition provided by *Ignall and Veinott* (1969), which was further extended by *Choi et al.* (2005) under a relaxed demand assumption. Our work focuses on a nonparametric variant in which the demand distribution is not known a priori.

Nonparametric inventory systems. *Burnetas and Smith* (2000) developed a gradient descent type algorithm for ordering and pricing when inventory is perishable; they showed that the average profit converges to the optimal but did not establish the rate of convergence. *Huh and Rusmevichientong* (2009) proposed gradient descent based algorithms for lost-sales systems with censored demand. Subsequently, *Huh et al.* (2009) proposed algorithms for finding the optimal base-stock policy in lost-sales inventory systems with positive lead time. *Huh et al.* (2011) applied the concept of Kaplan-Meier estimator to devise another data-driven algorithm for censored demand. Other nonparametric approaches in the inventory literature include sample average approximation (SAA) (e.g., *Kleywegt et al.* (2002), *Levi et al.* (2007), *Levi et al.* (2015)) which uses the empirical distribution formed by *uncensored* samples drawn from the true distribution. Concave adaptive value estimation (e.g., *Godfrey and Powell* (2001), *Powell et al.* (2004)) successively approximates the objective cost function with a sequence of piecewise linear functions. The bootstrap method (e.g., *Bookbinder and Lordahl* (1989)) estimates the newsvendor quantile of the demand distribution. The infinitesimal perturbation approach (IPA) is a sampling-based stochastic gradient estimation technique that has been used to solve stochastic supply chain models (see, e.g., *Glasserman* (1991)). *Maglaras and Eren* (2015) employed maximum entropy distributions to solve a stochastic capacity control problem. For parametric approaches, such as Bayesian learning (see, e.g., *Lariviere and Porteus* (1999), *Chen and Plambeck* (2008)) or operational statistics (see, e.g., *Liyanage and Shanthikumar* (2005), *Chu et al.* (2008)) in stochastic inventory systems, we refer readers to *Huh and Rusmevichientong* (2009) for an excellent discussion of the key differences between nonparametric and parametric approaches. This chapter contributes to the literature by studying multi-product inventory systems under a warehouse-capacity constraint, which is significantly more complex to analyze.

Online convex optimization. The aim of online convex optimization is to minimize the cumulative loss function defined over a convex compact set with online learning process since the optimizer does not know the (convex) objective function a priori (see *Hazan* (2016), *Shalev-Shwartz* (2012) for an overview). *Zinkevich* (2003) has shown that the average T-period cost using a gradient descent based algorithm converges to the optimal cost at the rate of $O(1/\sqrt{T})$. This result was further extended by *Flaxman et al.* (2005) in a bandit setting. Under additional technical assumptions, a modified algorithm by *Hazan et al.* (2006) achieves a faster convergence rate $O(\log T/T)$. Our problem differs from the conventional online convex optimization problems in that the target levels (or the iterates) may not be achieved due to policy-dependent dynamic inventory constraints.

Stochastic approximation. The proposed gradient descent type of algorithm also resembles the ones used in the Stochastic Approximation (SA) literature (see *Nemirovski et al.* (2009) and references therein), which should be carefully contrasted with ours. First, SA algorithms aim to solve a single-stage stochastic optimization problem by making successive experiments while the cost of experiments is ignored. On the other hand, our algorithm aims to minimize the cumulative loss suffered along the learning progress for a multi-stage closed-loop stochastic optimization problem. Putting into context, SA focuses on measuring the terminal regret $\mathbb{E}[\Pi(\mathbf{y}_T) - \Pi(\mathbf{y}^*)]$, whereas our algorithm focuses on measuring the cumulative loss over time $\mathbb{E}\left[\sum_{t=1}^{T}(\Pi(\mathbf{y}_t) - \Pi(\mathbf{y}^*))\right]$. Second, in the analysis of robust SA algorithms with general convex costs, the step size is chosen to be $O(1/\sqrt{t})$ to obtain a convergence rate of $O(1/\sqrt{t})$ in the terminal regret criterion by appropriately *averaging* the iterate solutions. The standard robust SA approaches cannot be adapted to our setting where the iterates cannot move "freely" due to policy-driven dynamic inventory constraints.

General notation. For any real vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\mathbf{y} \geq \mathbf{x}$ means component-wise greater or equal to; $\mathbf{x}^+ = (\max\{x^i, 0\})_{i=1}^n$; $|\mathbf{x}| = (|x^i|)_{i=1}^n$; the *join* operator $\mathbf{x} \vee \mathbf{y} = (\max\{x^i, y^i\})_{i=1}^n$; the *meet* operator $\mathbf{x} \wedge \mathbf{y} = (\min\{x^i, y^i\})_{i=1}^n$; for any integers t and s with $t \leq s$, $\mathbf{x}_{[t,s]} = \sum_{j=t}^{s} \mathbf{x}_j$ and $\mathbf{x}_{[t,s)} = \sum_{j=t}^{s-1} \mathbf{x}_j$; $||\cdot||$ or $||\cdot||_2$ means 2-norm; $||\cdot||_1$ means 1-norm. The notation $\triangleq$ means "is defined as".

2.2 Multi-Product Stochastic Inventory Systems

We consider a stochastic T-period n-product inventory system under a warehouse-capacity constraint M (e.g., *Ignall and Veinott* (1969), *Beyer et al.* (2001)). The firm has no knowledge of the true underlying demand distribution a priori, but can observe past sales data (i.e., censored demand data), and make adaptive inventory decisions based on the available information.

Random demand and regularity assumptions. For each period $t = 1, \ldots, T$ and each product $i = 1, \ldots n$, we denote the demand of product i in period t by a random variable D_t^i. For notational convenience, we use $\mathbf{D}_t = (D_t^1, \ldots, D_t^n)$ to denote the random demand vector in period t, and $\mathbf{d}_t = (d_t^1, \ldots, d_t^n)$ to denote their realizations.

Assumption 2.1. *We make the following assumptions and regularity conditions on demand.*

(i). For each product i, D_t^i is i.i.d. across time period t.

(ii). For each product i and for each period t, D_t^i is independent (but not necessarily identically distributed) of D_s^j for all $j \neq i$ and $s = 1, \ldots, T$.

(iii). For each product i and for each period t, D_t^i is a continuous random variable defined on a finite support $[0, M]$, whose CDF $F_{D^i}(\cdot)$ is differentiable and density $F'_{D^i}(x) > 0$ for all $x \in [0, M]$.

(iv). For each product i and for each period t, $\mathbb{E}[D_t^i] \geq l$ for some real number $l > 0$.

Assumptions 2.1(a) and 2.1(b) assume some form of stationarity of demand, which is predominant in the nonparametric learning literature (see, e.g., *Levi et al.* (2007), *Huh et al.* (2009, 2011), *Huh and Rusmevichientong* (2009), *Besbes and Muharremoglu* (2013)). Assumption 2.1(c) ensures the per-period cost function defined in (2.3) is differentiable, finite-valued and strictly (jointly) convex, which guarantees a unique minimizer. Assumption 2.1(d) rules out degenerate demands.

System dynamics and objectives. Let $\mathbf{f}_t$ denote the information collected up to the beginning of period t, which includes all the realized demands and past decisions. A feasible *closed-loop* policy π is a sequence of functions $\mathbf{y}_t = \pi_t(\mathbf{x}_t, \mathbf{f}_t)$, $t = 1, \ldots, T$, mapping beginning inventory $\mathbf{x}_t$ and $\mathbf{f}_t$ (state) into ending inventory $\mathbf{y}_t$ (decision) while satisfying $\mathbf{y}_t \geq \mathbf{x}_t$ and the warehouse-capacity constraint (see *Bertsekas* (2000) for discussions on closed-loop optimization problems). Note that when the demand distribution is known a prior, it suffices to consider policies of the form $\mathbf{y}_t = \pi_t(\mathbf{x}_t)$, due to the assumed across-time independence of demands (see *Bertsekas and Shreve* (2007)).

Given a feasible policy π, we describe the sequence of events below. (Note that $\mathbf{x}_t^\pi$, $\mathbf{y}_t^\pi$ and $\mathbf{q}_t^\pi$'s are functions of π; for ease of presentation, we make their dependence on π implicit.)

(i). At the beginning of period t, the firm observes the starting inventory $\mathbf{x}_t = (x_t^1, \ldots, x_t^n)$.

(ii). The firm decides to order $\mathbf{q}_t = (q_t^1, \ldots, q_t^n) \geq 0$, and the ending inventory $\mathbf{y}_t = \mathbf{x}_t + \mathbf{q}_t$, where $\mathbf{y}_t = (y_t^1, \ldots, y_t^n)$. We assume instantaneous replenishment. The total inventory level is restricted by a warehouse-capacity constraint (see *Ignall and Veinott* (1969)), i.e.,

$$\mathbf{y}_t \in \Gamma \triangleq \left\{ \mathbf{y}_t \in \mathbb{R}_+^n : \sum_{i=1}^n y_t^i \leq M \right\}. \tag{2.1}$$

(iii). The demand $\mathbf{D}_t$ is realized, denoted by $\mathbf{d}_t$, which is satisfied to the maximum extent using on-hand inventory. Unsatisfied demand units are *lost*, and the firm only observes the *sales quantity* (or censored demand), i.e., $\min(d_t^i, y_t^i)$ for each product i in period t. The state transition can be written as $\mathbf{x}_{t+1} = (\mathbf{x}_t + \mathbf{q}_t - \mathbf{d}_t)^+ = (\mathbf{y}_t - \mathbf{d}_t)^+$.

(iv). The production, overage and underage costs at the end of period t is then $\mathbf{c}\cdot\mathbf{q}_t + \mathbf{h}\cdot(\mathbf{y}_t - \mathbf{d}_t)^+ + \mathbf{p}\cdot(\mathbf{d}_t - \mathbf{y}_t)^+$, where $\mathbf{c} = (c^1, \ldots, c^n)$, $\mathbf{h} = (h^1, \ldots, h^n)$ and $\mathbf{p} = (p^1, \ldots, p^n)$ are the per-unit purchasing, holding and lost-sales penalty cost vectors, respectively. We note that the cost minimization model with lost-sales assumes that $\mathbf{p} \geq \mathbf{c}$ (see *Zipkin* (2000)) since the firm loses revenue and goodwill from the sale and the revenue has to be greater than the production cost. (Our approach also works for time-invariant random purchasing cost vector.)

Assuming the salvage value of any left-over product at the end of planning horizon equals its production cost, the total expected cost incurred by π can be written as

$$\begin{aligned} \mathcal{C}(\pi) &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{c}\cdot(\mathbf{y}_t - \mathbf{x}_t) + \mathbf{h}\cdot(\mathbf{y}_t - \mathbf{D}_t)^+ + \mathbf{p}\cdot(\mathbf{D}_t - \mathbf{y}_t)^+\right] - \mathbb{E}[\mathbf{c}\cdot\mathbf{x}_{T+1}], \\ &= -\mathbf{c}\cdot\mathbf{x}_1 + \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{c}\cdot\mathbf{y}_t + (\mathbf{h}-\mathbf{c})\cdot(\mathbf{y}_t - \mathbf{D}_t)^+ + \mathbf{p}\cdot(\mathbf{D}_t - \mathbf{y}_t)^+\right], \qquad (2.2) \end{aligned}$$

where the second equality follows from $\mathbf{x}_{t+1} = (\mathbf{y}_t - \mathbf{d}_t)^+$ and some simple algebra. If the underlying distribution $\mathbf{D}_t$ is given a priori, the stochastic inventory control problem specified above can be formulated using dynamic programming (see *Beyer et al.* (2001)) with state variables $\mathbf{x}_t$, control variables $\mathbf{y}_t$ (with $\mathbf{x}_t \leq \mathbf{y}_t \in \Gamma$), random disturbances $\mathbf{D}_t$, and state transition $\mathbf{x}_{t+1} = (\mathbf{y}_t - \mathbf{d}_t)^+$. It turns out that this problem is in fact "myopically" solvable, which is discussed next.

Clairvoyant optimal policy. We first characterize the clairvoyant optimal policy where the distribution of $\mathbf{D}_t$ is known a priori. We define $\Pi(\cdot)$ to be the per-period

expected cost function,

$$\Pi(\mathbf{a}) = \Pi_t(\mathbf{a}) \triangleq \mathbb{E}\left[\mathbf{c}\cdot\mathbf{a} + (\mathbf{h}-\mathbf{c})\cdot(\mathbf{a}-\mathbf{D}_t)^+ + \mathbf{p}\cdot(\mathbf{D}_t-\mathbf{a})^+\right]. \tag{2.3}$$

Let $\mathbf{y}^*$ be a unique critical (deterministic) vector defined by

$$\mathbf{y}^* \triangleq \arg\min_{\mathbf{a}\in\Gamma:\mathbf{a}\geq\mathbf{0}} \Pi(\mathbf{a}). \tag{2.4}$$

Theorem 2.2. *Under Assumption 2.1, when the demand distribution is known a priori, ordering up to* $\mathbf{y}^*$ *defined in* (2.4) *in each period is optimal, with expected per-period cost* $\Pi(\mathbf{y}^*)$.

Proof. Based on (2.3), we define a *myopic* feasible (closed-loop) policy $\bar{\pi}$ as a sequence of functions $\bar{\mathbf{y}}_t = \bar{\pi}_t(\mathbf{x}_t)$, $t = 1,\ldots,T$, mapping beginning inventory (state) $\mathbf{x}_t$ into ending inventory (decision) $\bar{\mathbf{y}}_t$, which also "myopically" minimizes per-period cost $\Pi_t(\cdot)$ with beginning inventory $\mathbf{x}_t$, i.e.,

$$\bar{\mathbf{y}}_t(\mathbf{x}_t) \triangleq \arg\min_{\mathbf{a}\in\Gamma:\mathbf{a}\geq\mathbf{x}_t} \Pi_t(\mathbf{a}). \tag{2.5}$$

The above feasible policy $\bar{\pi}$ is myopic, because it only optimizes per-period cost in each period (the immediate reward). This is in contrast with standard dynamic programming or approximate dynamic programming approaches. To ease the presentation of establishing optimality of $\bar{\pi}$, following *Ignall and Veinott* (1969), we keep $\mathbf{x}_t$, $\bar{\mathbf{y}}_t$, Π_t time-generic, i.e.,

$$\bar{\mathbf{y}}(\mathbf{x}) \triangleq \arg\min_{\mathbf{a}\in\Gamma:\mathbf{a}\geq\mathbf{x}} \Pi(\mathbf{a}). \tag{2.6}$$

It is important to see that $\bar{\mathbf{y}}(\mathbf{x})$ is the unique minimizer of (2.6), due to Assumption 2.1 ensuring strict (joint) convexity of $\Pi(\mathbf{y})$ over the feasible region, and the fact that

the constraint set is affine (see *Boyd and Vandenberghe* (2004)). □

Lemma 2.3. *The optimization problem defined in* (2.6) *has a unique minimizer* $\bar{\mathbf{y}}(\mathbf{x})$.

Proof. Due to Assumption 2.1, the cost function $\Pi(\cdot)$ is differentiable and finite-valued. The derivatives inside expectation are bounded, and also the expectation is a multiple integration over finite ranges. Hence this guarantees the validity of interchange between differentiation and expectation.

Next we argue that $\Pi(\cdot)$ are strictly (jointly) convex over the feasible region. For all i and j,

$$\frac{\partial^2 \Pi(\mathbf{a})}{\partial (a^i)^2} = (h^i + p^i - c^i) F'_{D^i}(a^i) > 0; \quad \frac{\partial^2 \Pi(\mathbf{a})}{\partial a^i \partial a^j} = 0,$$

where Assumption 1(c) ensures $F'_{D^i}(a^i) > 0$ for all $a^i \in [0, M]$. Hence, the Hessian matrix is positive definte (with all strictly positive eigenvalues) over the entire feasible region, ensuring Π to be strictly (jointly) convex.

Now consider the optimization problem (with a given starting inventory $\mathbf{x}$) defined in (2.6). Since $\Pi(\mathbf{y})$ is strictly (jointly) convex and the constraint set is affine, $\bar{\mathbf{y}}(\mathbf{x})$ is the unique minimizer. (See *Boyd and Vandenberghe* (2004) for discussions of unique minimizer in convex optimization problems and also Example 5.4.). □

Next we shall show that the myopic policy $\bar{\pi}$ defined above is optimal. *Ignall and Veinott* (1969) provided a sufficient condition called *substitute property* (together with two mild regularity assumptions) under which the myopic policy is optimal.

Definition 2.4 (Substitute property)**.** For any inventory levels $\mathbf{x}, \tilde{\mathbf{x}} \in \Gamma$,

$$\text{if } \mathbf{x} \geq \tilde{\mathbf{x}}, \text{ then } \bar{\mathbf{y}}(\mathbf{x}) - \mathbf{x} \leq \bar{\mathbf{y}}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}.$$

Definition 2.5 (Regularity conditions in *Ignall and Veinott* (1969))**.** The two regularity conditions in *Ignall and Veinott* (1969) are: (a) $\mathbf{x} \leq \mathbf{x}' \leq \bar{\mathbf{y}}(\mathbf{x})$ implies

$\bar{\mathbf{y}}(\mathbf{x}) = \bar{\mathbf{y}}(\mathbf{x}')$ for $\mathbf{x}, \mathbf{x}' \in \Gamma$; (b) The state transition permits either pure, partial, or no backlogging (lost-sales).

The regularity condition (a) is satisfied by $\bar{\mathbf{y}}(\mathbf{x})$ being the unique minimizer of (2.6) by Lemma 2.3, and the regularity condition (b) is immediate since we consider a standard lost-sales model.

We can now proceed to establish the optimality of myopic policies for the multi-product lost-sales system by showing that the sufficient condition (substitute property) given above holds for our system.

Proposition 2.6. *Under Assumption 2.1, when the demand distribution is known a priori, the myopic ordering policy defined in* (2.5) *is optimal for the multi-product lost-sales inventory systems.*

To prove Proposition 2.6, we need to derive several important properties of the myopic policy. Now consider the two possible starting inventory levels $\mathbf{x}$ and $\tilde{\mathbf{x}}$, with $\mathbf{x} \geq \tilde{\mathbf{x}}$. For notational (superscript) convenience, we use θ instead of $\mathbf{y}^*$ to be the global minimizer of $\Pi(\cdot)$ over Γ. Recall that $\theta = \mathbf{y}^* \triangleq \arg\min_{\mathbf{a}\in\mathbf{\Gamma}} \Pi(\mathbf{a})$, and also the myopic order-to-up level $\bar{\mathbf{y}}(\mathbf{x}) \triangleq \arg\min_{\mathbf{a}\in\mathbf{\Gamma}:\mathbf{a}\geq\mathbf{x}} \Pi(\mathbf{a})$. For simplicity, we define the boundary of our warehouse storage constraint,

$$\partial\Gamma \triangleq \left\{ \mathbf{y} \in \mathbb{R}^n_+ : \sum_{i=1}^{n} y^i = M \right\}.$$

Note that $\mathbf{y} \in \partial\Gamma$ means that the total order-up-to levels have reached the total storage limit M. If $\mathbf{y} \notin \partial\Gamma$, then the warehouse storage constraint is not tight.

Now denote the j^{th} partial derivative of $\Pi(\cdot)$ by $\Pi'_j(\cdot)$. We then develop some useful properties of the myopic order-up-to levels $\bar{\mathbf{y}}(\cdot)$.

Lemma 2.7. *Let* $\mathbf{x} \in \Gamma$ *and* θ *be the global minimizer of* $\Pi(\cdot)$ *over* Γ*,*

(i). $x^j \geq \theta^j \Rightarrow \bar{y}^j(\mathbf{x}) = x^j$.

(ii). $x^j \leq \theta^j \Rightarrow \bar{y}^j(\mathbf{x}) \leq \theta^j$.

Proof. The proof is straightforward. Statement (i) holds because $x^j \geq \theta^j$ (the starting inventory is higher than the global minimizer) for product j, it is sub-optimal to order any more product j. Statement (ii) holds because if $x^j \leq \theta^j$ (the starting inventory is lower than the global minimizer), it is sub-optimal to raise the inventory above the global minimizer. □

Lemma 2.8. *Let* $\mathbf{x} \in \Gamma$ *and* θ *be the global minimizer of* $\Pi(\cdot)$ *over* Γ*,*

(i). $\theta \in \partial\Gamma \Rightarrow \bar{\mathbf{y}}(\mathbf{x}) \in \partial\Gamma$*;*

(ii). $\bar{\mathbf{y}}(\mathbf{x}) \notin \partial\Gamma, x^j \leq \theta^j \Rightarrow \bar{y}^j(\mathbf{x}) = \theta^j$.

In Lemma 2.8, statement (i) states that if the global minimizer occupies the entire storage space, then the myopic order-up-to levels will also occupy the entire storage space. This is because our myopic policy will always order as much as possible to approach the global minimizer. Statement (ii) states that if the total myopic order-up-to level has not reached the storage limit M, then if $x^j \leq \theta^j$, the myopic policy will raise inventory level for product j to the global minimizer θ^j.

Proof. We prove (i) by contradiction. Suppose that $\theta \in \partial\Gamma$ and $\bar{\mathbf{y}}(\mathbf{x}) \notin \partial\Gamma$, then

$$\sum_{i=1}^{n} \theta^i = M \qquad \text{and} \qquad \sum_{i=1}^{n} \bar{y}^i(\mathbf{x}) < M.$$

It is obvious that there exists at least one j such that $\bar{y}^j(\mathbf{x}) < \theta^j$. Since θ minimizes $\Pi(\cdot)$ over Γ, it is clear that θ either reaches the global minimizer of $\Pi(\cdot)$ over the entire real line $\mathbb{R}$ or is smaller than it due to the storage constraint, so the derivative $\Pi'_j(\theta) \leq 0$. Therefore, since $\Pi(\cdot)$ is strictly convex,

$$\Pi'_j(\bar{\mathbf{y}}(\mathbf{x})) < \Pi'_j(\theta) \leq 0.$$

On the other hand, since $\bar{\mathbf{y}}(\mathbf{x}) \notin \partial\Gamma$ and $\bar{\mathbf{y}}(\mathbf{x})$ is a minimizer of $\Pi(\cdot)$ over set $\{\mathbf{y} \mid \mathbf{y} \geq \mathbf{x}, \mathbf{y} \in \Gamma\}$, it is clear that $\bar{\mathbf{y}}(\mathbf{x})$ either reaches θ or is greater than it because of the initial on-hand inventory, so $\Pi'_j(\bar{\mathbf{y}}(\mathbf{x})) \geq 0$, which results in a contradiction, thereby proving (i).

To prove (ii), we observe from the contraposition of (i), i.e., $\bar{\mathbf{y}}(\mathbf{x}) \notin \partial\Gamma \Rightarrow \theta \notin \partial\Gamma$. Then for any product j, $\bar{y}^j(\mathbf{x})$ is not restricted by the storage constraint, and thus if $\theta^j \geq x^j$, then θ^j can always be reached, implying that $\bar{y}^j(\mathbf{x}) = \theta^j$. This completes the proof. □

Lemma 2.9. $\bar{y}^j(\mathbf{x}) > x^j \Rightarrow \Pi'_j(\bar{\mathbf{y}}(\mathbf{x})) = \min_i \Pi'_i(\bar{\mathbf{y}}(\mathbf{x}))$

Lemma 2.9 states that if a product is ordered, then the marginal cost of any additional ordering must be equal across the products. Intuitively, if the marginal cost of ordering this product is higher than others, we can always reduce the quantity of this product and order more of the other products. The rigorous proof is as follows.

Proof. We prove this result by contradiction. Suppose that there exists an i, $1 \leq i \leq n$, such that $\Pi'_i(\bar{\mathbf{y}}(\mathbf{x})) < \Pi'_j(\bar{\mathbf{y}}(\mathbf{x}))$. Then, for a sufficiently small $\epsilon > 0$, $(\bar{y}^1(\mathbf{x}), ...\bar{y}^j(\mathbf{x}) - \epsilon, ..., \bar{y}^i(\mathbf{x}) + \epsilon, ..., \bar{y}^n(\mathbf{x})) \in \Gamma$, and we have

$$\begin{aligned} & \Pi(\bar{\mathbf{y}}(\mathbf{x})) - \Pi(\bar{y}^1(\mathbf{x}), ...\bar{y}^j(\mathbf{x}) - \epsilon, ..., \bar{y}^i(\mathbf{x}) + \epsilon, ..., \bar{y}^n(\mathbf{x})) \\ = \; & \epsilon(\Pi'_j(\bar{\mathbf{y}}(\mathbf{x})) - \Pi'_i(\bar{\mathbf{y}}(\mathbf{x}))) + o(\epsilon^2) > 0, \end{aligned}$$

which contradicts to the fact that $\bar{\mathbf{y}}(\mathbf{x})$ minimizes $\Pi(\cdot)$ over set $\{\mathbf{y} \mid \mathbf{y} \geq \mathbf{x}, \mathbf{y} \in \Gamma\}$. □

Now, we are ready to prove Proposition 2.6.

Proof. To establish the optimality of myopic policies for the multi-product lost-sales system, it suffices to verify that the substitute property (2.4) holds, i.e., for any inventory levels $\mathbf{x}, \tilde{\mathbf{x}} \in \Gamma$, if $\mathbf{x} \geq \tilde{\mathbf{x}}$, then $\bar{\mathbf{y}}(\mathbf{x}) - \mathbf{x} \leq \bar{\mathbf{y}}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}$.

We know that the myopic order-up-to levels $\bar{y}^j(\mathbf{x}) \geq x^j$ for any product j if $\mathbf{x} \in \Gamma$. Similarly, $\bar{y}^j(\tilde{\mathbf{x}}) \geq \tilde{x}^j$ for any product j if $\tilde{\mathbf{x}} \in \Gamma$. Now if $\bar{y}^j(\mathbf{x}) = x^j$, then we have

$$0 = \bar{y}^j(\mathbf{x}) - x^j \leq \bar{y}^j(\tilde{\mathbf{x}}) - \tilde{x}^j.$$

Thus, it suffices to prove that $\bar{y}^j(\mathbf{x}) \leq \bar{y}^j(\tilde{\mathbf{x}})$, whenever $\bar{y}^j(\mathbf{x}) > x^j$. We have to consider three cases as follows.

Case (a). First, if both $\bar{\mathbf{y}}(\mathbf{x}) \notin \partial\Gamma$ and $\bar{y}^j(\tilde{\mathbf{x}}) \notin \partial\Gamma$, then it follows from Lemma 2.7 and Lemma 2.8 that

$$\bar{y}^j(\mathbf{x}) = \max\{\theta^j, x^j\}, \qquad \bar{y}^j(\tilde{\mathbf{x}}) = \max\{\theta^j, \tilde{x}^j\}, \qquad \forall\, j.$$

Then $\bar{y}^j(\mathbf{x}) = \bar{y}^j(\tilde{\mathbf{x}})$ and the result follows immediately.

Case (b). Second, if $\bar{\mathbf{y}}(\mathbf{x}) \in \partial\Gamma$ but $\bar{y}^j(\tilde{\mathbf{x}}) \notin \partial\Gamma$, then by Lemma 2.7 (ii) and Lemma 2.8 (ii), we have $\bar{y}^j(\mathbf{x}) \leq \theta^j = \bar{y}^j(\tilde{\mathbf{x}})$, and the result also follows immediately. It is impossible for the case where $\bar{\mathbf{y}}(\mathbf{x}) \notin \partial\Gamma$ and $\bar{y}^j(\tilde{\mathbf{x}}) \in \partial\Gamma$ to happen. To see this, if such case exists, then we can always find some j such that for $x^j > \tilde{x}^j$, $\bar{y}^j(\tilde{\mathbf{x}}) > \bar{y}^j(\mathbf{x})$. However, by Lemma 2.7 (ii) and Lemma 2.8 (ii), we know that $\bar{y}^j(\mathbf{x}) \leq \theta^j = \bar{y}^j(\tilde{\mathbf{x}})$, which results in a contradiction.

Case (c). Third, we need to analyze the remaining case where $\bar{\mathbf{y}}(\mathbf{x}) \in \partial\Gamma$ and $\bar{y}^j(\tilde{\mathbf{x}}) \in \partial\Gamma$, i.e.,

$$\sum_{j=1}^{n} \bar{y}^j(\mathbf{x}) = \sum_{j=1}^{n} \bar{y}^j(\tilde{\mathbf{x}}) = M. \tag{2.7}$$

We partition all the products into three sets as follows,

$$I^a = \{k : \bar{y}^k(\mathbf{x}) > x^k\}, I^b = \{k : \bar{y}^k(\mathbf{x}) = x^k \cap \Pi'_k(\bar{y}^k(\mathbf{x}) \leq 0\}, I^c = \{k : \Pi'_k(\bar{y}^k(\mathbf{x}) > 0\}.$$

Note that these three sets are disjoint and the union of them is exhaustive.

Now we focus on the set I^c first and let $j \in I^c$. Then we have $\bar{y}^j(\mathbf{x}) \geq \max\{\tilde{x}^j, \theta^j\}$.

By Lemma 2.7, it is clear that $\bar{y}^j(\tilde{\mathbf{x}}) \leq \max\{\tilde{x}^j, \theta^j\}$. Hence, $\bar{y}^j(\tilde{\mathbf{x}}) - \bar{y}^j(\mathbf{x}) \leq 0$ for all $j \in I^c$. Together with (2.7), we know that

$$\sum_{j \in I^a \cup I^b}^{n} \left(\bar{y}^j(\tilde{\mathbf{x}}) - \bar{y}^j(\mathbf{x})\right) \geq 0.$$

If $\bar{y}^m(\tilde{\mathbf{x}}) = \bar{y}^m(\mathbf{x})$ for all $m \in I^a \cup I^b$, then the result follows immediately. Now consider the case where there exists a product $m \in I^a \cup I^b$ such that $\bar{y}^m(\tilde{\mathbf{x}}) > \bar{y}^m(\mathbf{x})$. This implies that $\bar{y}^m(\tilde{\mathbf{x}}) > \bar{y}^m(\mathbf{x}) \geq x^m \geq \tilde{x}^m \geq 0$. By Lemma 2.9, we have $\Pi'_m(\bar{\mathbf{y}}(\tilde{\mathbf{x}})) = \min_i \Pi'_i(\bar{\mathbf{y}}(\tilde{\mathbf{x}}))$. Moreover, due to the strict convexity of $\Pi(\cdot)$, then we have

$$\min_i \Pi'_i(\bar{\mathbf{y}}(\tilde{\mathbf{x}})) = \Pi'_m(\bar{\mathbf{y}}(\tilde{\mathbf{x}})) > \Pi'_m(\bar{\mathbf{y}}(\mathbf{x})). \tag{2.8}$$

To complete the proof, it suffices to show that for any product $j \in I^a$, $\bar{y}^j(\tilde{\mathbf{x}}) \geq \bar{y}^j(\mathbf{x})$. Now suppose there exists a product $n \in I^a$ such that $\bar{y}^n(\tilde{\mathbf{x}}) < \bar{y}^n(\mathbf{x})$. It is clear that $\bar{y}^n(\mathbf{x}) > \bar{y}^n(\tilde{\mathbf{x}}) \geq 0$. By Lemma 2.9, we have $\Pi'_n(\bar{\mathbf{y}}(\mathbf{x})) = \min_i \Pi'_i(\bar{\mathbf{y}}(\mathbf{x}))$. Moreover, due to the strict convexity of $\Pi(\cdot)$, then we have

$$\min_i \Pi'_i(\bar{\mathbf{y}}(\mathbf{x})) = \Pi'_n(\bar{\mathbf{y}}(\mathbf{x})) > \Pi'_n(\bar{\mathbf{y}}(\tilde{\mathbf{x}})). \tag{2.9}$$

Note that (2.8) implies that $\Pi'_n(\bar{\mathbf{y}}(\tilde{\mathbf{x}})) > \Pi'_m(\bar{\mathbf{y}}(\mathbf{x}))$ but (2.9) implies that $\Pi'_m(\bar{\mathbf{y}}(\mathbf{x})) > \Pi'_n(\bar{\mathbf{y}}(\tilde{\mathbf{x}}))$, which results in a contradiction. This completes the proof. □

Equipped with Proposition 2.6, we are ready to prove Theorem 2.2.

Proof. Proposition 2.6 fully characterizes the structural properties of optimal policies as follows. Let $\mathbf{y}^*$ be a unique critical (deterministic) vector defined by in (2.4). Then a clairvoyant optimal policy π^* is characterized as follows:

(i). If the beginning inventory level of product i is above its individual base-stock level (i.e., the i^{th} component of $\mathbf{y}^*$), then this product is not ordered in the

period.

(ii). If this product i is ordered in the period, the ending inventory level (after ordering) does not exceed its individual base-stock level (i.e., the i^{th} component of $\mathbf{y}^*$).

(iii). If there is enough storage space to bring all products (whose inventory levels are below their individual base-stock levels) up to their base-stock levels, then such an order is optimal. Otherwise, the ending inventory levels takes up all the available storage space.

Thus, the stationary multi-period inventory problem is analytically equivalent to the single-period problem, and ordering up to $\mathbf{y}^*$ in each period is also optimal for this problem. Clearly, once we start below $\mathbf{y}^*$, and order up to $\mathbf{y}^*$, we remain at or below $\mathbf{y}^*$ thereafter; in such a case, the expected cost incurred in each period is $\Pi(\mathbf{y}^*)$. □

2.3 Nonparametric Data-Driven Inventory Control Policies

When the firm has no knowledge of the true underlying distribution of $\mathbf{D}_t$ a priori, we aim to find a provably good adaptive data-driven inventory control policy that makes the total expected system costs close to the optimal strategy. The proposed data-driven algorithm DDM maintains a vector triplet of sequences $(\mathbf{z}_t, \hat{\mathbf{y}}_t, \mathbf{y}_t)_{t\geq 0}$. The first sequence $(\mathbf{z}_t)_{t\geq 0}$ represents the *constraint-free target* inventory levels where the warehouse storage constraint is waived. The second sequence $(\hat{\mathbf{y}}_t)_{t\geq 0}$ represents the *target* inventory levels when the warehouse storage constraint is taken into account. However, the target inventory levels $(\hat{\mathbf{y}}_t)_{t\geq 0}$ may not be always feasible due to warehouse capacity constraint and positive inventory carry-over. Thus, we use the third sequence $(\mathbf{y}_t)_{t\geq 0}$ to represent the *actual implemented* inventory levels after ordering.

We first present a compact description of our data-driven multi-product algorithm (DDM).

Data-Driven Multi-product Algorithm (DDM).

Step 0. (Initialization.) Set the initial inventory levels $\mathbf{y}_0 = \hat{\mathbf{y}}_0 = \mathbf{z}_0$ to be any values within Γ and then set the initial values $t = 0$, $\tau_0 = 0$ and $k = 0$.

For each period $t = 0, \ldots, T-1$, repeat the following steps:

Step 1. (Setting the constraint-free and constrained target inventory levels.)

Case 1: If $\mathbf{y}_t \geq \hat{\mathbf{y}}_t$ (i.e., $y_t^i \geq \hat{y}_t^i$ for all $i = 1, \ldots, n$), the algorithm updates the constraint-free target inventory levels $\mathbf{z}_{t+1}$ by

$$\begin{aligned} \mathbf{z}_{t+1} &= \hat{\mathbf{y}}_t - \eta_t \mathbf{G}_t(\hat{\mathbf{y}}_t), \\ \text{where } \eta_t &= \left(\frac{\gamma M}{\sqrt{n} \cdot \max_i\{p^i - c^i, h^i\}} \right) \frac{1}{\sqrt{t}} \text{ for some } \gamma > 0 \end{aligned} \tag{2.10}$$

for each product $i = 1, \ldots, n$, and the i^{th} component of $\mathbf{G}_t$ is defined as

$$G_t^i(\hat{\mathbf{y}}_t) = \begin{cases} h^i, & \text{if } \hat{y}_t^i > d_t^i, \\ -(p^i - c^i), & \text{if } \hat{y}_t^i \leq d_t^i. \end{cases} \tag{2.11}$$

Note that $\gamma = 1$ for achieving the tightest theoretical bound.

Then the algorithm sets the constrained target inventory levels $\hat{\mathbf{y}}_{t+1}$ by solving

$$\hat{\mathbf{y}}_{t+1} = \arg\min_{\mathbf{w} \in \Gamma} ||\mathbf{w} - \mathbf{z}_{t+1}||_2. \tag{2.12}$$

Record the break point $\tau_k := t$ and increase the value k by 1.

Case 2: Else if $\mathbf{y}_t \not\geq \hat{\mathbf{y}}_t$ (i.e., there exists an i such that $y_t^i < \hat{y}_t^i$), the algorithm keeps both the constraint-free and constrained target inventory levels unchanged, i.e., $\mathbf{z}_{t+1} = \mathbf{z}_t$ and $\hat{\mathbf{y}}_{t+1} = \hat{\mathbf{y}}_t$.

Step 2. (Solving for the actual implemented target inventory levels.)

Define the set J and its complement as

$$J \triangleq \left\{ i : x_{t+1}^i > \hat{y}_{t+1}^i \right\}, \qquad \bar{J} \triangleq \left\{ i : x_{t+1}^i \leq \hat{y}_{t+1}^i \right\}. \tag{2.13}$$

For each product $i \in J$, we set the actual implemented levels

$$y_{t+1}^i = x_{t+1}^i, \quad \text{if } x_{t+1}^i > \hat{y}_{t+1}^i. \tag{2.14}$$

If $\bar{J} \neq \emptyset$, then we set the actual implemented levels $\mathbf{y}_{t+1}$ by solving

$$\min \sum_{i \in \bar{J}} (\hat{y}_{t+1}^i - y_{t+1}^i)^2 \quad \text{s.t.} \quad \sum_{i \in \bar{J}} y_{t+1}^i \leq M - \sum_{j \in J} x_{t+1}^j, \quad y_{t+1}^i \geq x_{t+1}^i, \ \forall\, i \in \bar{J}. \tag{2.15}$$

This concludes the description of the algorithm.

2.3.1 Algorithm Overview of DDM and Properties

Step 1: (Stochastic Gradient Descent). Let $\mathcal{T} = \{\tau_0, \tau_1, \ldots, \tau_m\}$ with $\tau_m \leq T$, which is the set of break points of DDM. In each period $\tau_k + 1$ $(k = 1, \ldots, m)$, we update the constraint-free target levels $\mathbf{z}_{t+1}$ by a stochastic gradient descent step. Conceptually, we update the minimizer along the negative direction of the true gradient of $\Pi(\cdot)$. However, since the true cost function $\Pi(\cdot)$ is not available to us (without knowing the underlying demand distribution), we can only rely on the observed sales data $\mathbf{d}_t$ to provide us an estimator of the true gradient of $\Pi(\hat{\mathbf{y}}_t)$ at the points $\hat{\mathbf{y}}_t$. The estimator $G_t^i(\hat{\mathbf{y}}_t)$ defined in (2.10) can be computed using the sales (censored demand) data observed by the firm in period $t \in \mathcal{T}$. When $t \in \mathcal{T}$, we have $y_t^i \geq \hat{y}_t^i$ for all $i = 1, \ldots, n$. Hence, the event $\{\hat{y}_t^i \leq d_t^i\}$ is equivalent to the case where the ending inventory in period t is at most $y_t^i - \hat{y}_t^i$, which is an observable event; the event $\{\hat{y}_t^i > d_t^i\}$ is equivalent to the case where the ending inventory in period t is strictly greater than $y_t^i - \hat{y}_t^i$, which is also observable. In this case, $\mathbf{G}_t$ defined in (2.11) is

an unbiased estimator of the true gradient $\nabla\Pi(\hat{\mathbf{y}}_t)$ at $\hat{\mathbf{y}}_t$, i.e., $\mathbb{E}[\mathbf{G}_t(\hat{\mathbf{y}}_t)] = \nabla\Pi(\hat{\mathbf{y}}_t)$, where the expectation is taken over the demand in period t. On the other hand, when $t \notin \mathcal{T}$, $G_t^i(\hat{\mathbf{y}}_t)$ may be *indeterminable* because the actual implemented inventory levels could fall below the target order-up-to levels. To be more specific, when $y_t^i < \hat{y}_t^i$ and $y_t^i \leq d_t$, the firm only observes the stockout but not the lost-sales quantity. Therefore, the firm cannot distinguish between $y_t^i \leq d_t < \hat{y}_t^i$ and $y_t^i < \hat{y}_t^i \leq d_t$, and hence cannot determine the value of $G_t^i(\hat{\mathbf{y}}_t)$. In periods when $t \notin \mathcal{T}$, we keep the target order-up-to levels unchanged.

We then carry out a greedy projection of the constraint-free target inventory levels $\mathbf{z}_{t+1}$ onto the warehouse storage constraint set Γ via (2.12), more specifically,

$$\min \sum_{i=1}^{n} (\hat{y}_{t+1}^i - z_{t+1}^i)^2 \quad \text{s.t.} \quad \sum_{i=1}^{n} \hat{y}_{t+1}^i \leq M, \quad \hat{y}_{t+1}^i \geq 0, \ \forall\, i. \tag{2.16}$$

We also make two simple observations that will be useful in Section 2.4. (a) A simple observation leads to the lower and upper bounds of z_{t+1}^i for each product i, i.e., $\hat{y}_t^i - \eta_t h^i \leq z_{t+1}^i \leq \hat{y}_t^i + \eta_t(p^i - c^i)$. In fact, z_{t+1}^i has to hit one of the two boundaries. (b) Another important observation is that when the product i in the first step updates its constraint-free target level z_{t+1}^i through a positive direction, i.e., $z_{t+1}^i = \hat{y}_t^i + \eta_t(p^i - c^i) \geq \hat{y}_t^i \geq 0$, we must have $\hat{y}_{t+1}^i \leq z_{t+1}^i$. To see this, suppose otherwise $\hat{y}_{t+1}^i > z_{t+1}^i$, we can decrease $\hat{y}_{t+1}^i$ to z_{t+1}^i, thereby strictly improving the objective value of (2.16) while maintaining feasibility. On the other hand, when the product i in the first step updates its constraint-free target level z_{t+1}^i through a negative direction, we have $z_{t+1}^i = \hat{y}_t^i - \eta_t h^i \leq \hat{y}_t^i$, Thus, this leads to the following property that will be useful in the performance analysis,

$$\hat{y}_{t+1}^i \leq \hat{y}_t^i + \eta_t(p^i - c^i), \quad \forall\, i = 1, \ldots, n. \tag{2.17}$$

Step 2: (Maintaining Feasibility). The target inventory levels $\hat{\mathbf{y}}_{t+1}$ derived

in the second step may not be achievable or implementable, due to the physical inventory carry-over and the warehouse capacity constraint. We then need to carry out an additional optimization procedure as follows. This step tries to order as many products as possible to reach the target level, and it is easy to solve quantitatively but hard to analyze. First we divide all the products into two groups, namely, the set J and its complement as defined in (2.13). We then have the following two cases.

Case 1. For each product $i \in J$, i.e., the beginning inventory level of product i is already greater than its target level. It is natural to not order any more product i and hence we follow (2.14).

Case 2. Now we focus on the set $\bar{J} \neq \emptyset$. Since the remaining inventory space now becomes $M - \sum_{j \in J} x_{t+1}^j$, we solve the optimization problem (2.15) to determine the actual implemented levels $\mathbf{y}_{t+1}$. Note that the optimization problem is well-defined since

$$M - \sum_{j \in J} x_{t+1}^j = M - \sum_{j \in J} (y_t^j - d_t^j)^+ \geq M - \sum_{j \in J} y_t^j \geq 0,$$

where the inequality follows from the fact that the algorithm keeps $\mathbf{y}_t \in \Gamma$.

The optimization (2.15) attempts to raise our inventory level as close as possible to the target inventory level $\hat{y}_{t+1}^i$ for each product $i \in \bar{J}$; however, it is possible that some of the products in $\bar{J}$ cannot hit the target level due to inventory constraints. Since we minimize the 2-norm type of objective function, it can be readily verified that the optimization (2.15) makes the *shortfalls* defined as $\hat{y}_{t+1}^i - y_{t+1}^i$ as even as possible across the products in the set $\bar{J}$.

Note that if the optimal objective value of (2.15) is equal to 0, then the algorithm goes to Case 1 in the next period and updates the target inventory levels. Otherwise it goes to Case 2 and maintains the target inventory levels; while maintaining these target levels, the inventory levels within J are decreasing and more inventory space is freed over time, and the shortfalls will decrease to zero.

2.4 Performance Analysis of DDM

The regret of our data-driven algorithm, denoted by $\mathcal{R}_T$, is defined as the difference between the optimal clairvoyant cost (given the demand distribution a priori) and the cost incurred by our data-driven algorithm (which learns the demand distribution over time). That is, for any $T \geq 1$,

$$\mathcal{R}_T \triangleq \mathbb{E}\left[\sum_{t=1}^{T} \Pi(\mathbf{y}_t)\right] - \sum_{t=1}^{T} \Pi(\mathbf{y}^*),$$

where $\mathbf{y}_t$ are the actual implemented order-up-to levels of our nonparametric (closed-loop) algorithm DDM, and $\mathbf{y}^*$ is the clairvoyant optimal solution in (2.4).

Theorem 2.10 below states the main result in this chapter.

Theorem 2.10. *Under Assumption 2.1, the average regret $\mathcal{R}_T/T$ of our data-driven algorithm DDM approaches 0 at the rate of $1/\sqrt{T}$. That is, there exists some constant K, such that for any $T \geq 1$,*

$$\frac{1}{T}\mathcal{R}_T \triangleq \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \Pi(\mathbf{y}_t)\right] - \Pi(\mathbf{y}^*) \leq \frac{K}{\sqrt{T}},$$

where $\mathbf{y}_t$ are actual implemented order-up-to levels of our nonparametric (closed-loop) algorithm DDM, and $\mathbf{y}^*$ is the clairvoyant optimal solution in (2.4).

It is known that in the general convex case (without assuming smoothness and strong convexity), this rate of $O(1/\sqrt{T})$ is unimprovable (see, e.g., Theorem 3.2. of *Hazan* (2016)). Our key contribution here is to establish this best possible rate even with inventory and capacity constraints (i.e., the iterates cannot move "freely" due to policy-driven dynamic inventory constraints).

Then the proof of Theorem 2.10 is the direct consequence of the following two key lemmas.

Lemma 2.11. *For any $T \geq 1$, there exists a constant $K_1 \in \mathbb{R}$ such that*

$$\Delta_1(T) = \mathbb{E}\left[\sum_{t=1}^{T} \Pi(\hat{\mathbf{y}}_t) - \sum_{t=1}^{T} \Pi(\mathbf{y}^*)\right] \leq K_1\sqrt{T},$$

where $\hat{\mathbf{y}}_t$ are target order-up-to levels of DDM, and $\mathbf{y}^*$ is the clairvoyant optimal solution in (2.4).

Lemma 2.12. *For any $T \geq 1$, there exists some constant $K_2 \in \mathbb{R}$ such that*

$$\Delta_2(T) = \mathbb{E}\left[\sum_{t=1}^{T} \Pi(\mathbf{y}_t) - \sum_{t=1}^{T} \Pi(\hat{\mathbf{y}}_t)\right] \leq K_2\sqrt{T},$$

where $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$ are actual implemented and target order-up-to levels of DDM, respectively.

2.4.1 Bound on Δ_1 - Online Convex Optimization (Proof of Lemma 2.11)

The proof of Lemma 2.11 builds upon the ideas and techniques used in online convex optimization (see, e.g., *Zinkevich* (2003) and *Flaxman et al.* (2005)). It is shown the cost function $\Pi(\cdot)$ is jointly convex, and $G(\cdot)$ is an unbiased estimator of the true expected gradient of $\Pi(\cdot)$ under censored demand within the set of breakpoints. In addition, this gradient estimator is bounded, i.e., $||G(\cdot)||_2^2 \leq n(\max_i\{p^i - c^i, h^i\})^2$.

Proof. Due to convexity of the cost function $\Pi(\mathbf{y})$, we have

$$\mathbb{E}\left[\Pi(\hat{\mathbf{y}}_t) - \Pi(\mathbf{y}^*)\right] \leq \mathbb{E}\left[\nabla\Pi(\hat{\mathbf{y}}_t)(\hat{\mathbf{y}}_t - \mathbf{y}^*)\right]. \tag{2.18}$$

Note that the subgradient $\nabla\Pi(\hat{\mathbf{y}}_t)$ defines the supporting hyperplane of Π at the point $\hat{\mathbf{y}}_t$.

For any period $t \in \mathcal{T}$, i.e., in the set of break points, we can obtain the upper bound of the second moment difference between our target inventory level and the

optimal target inventory level.

$$\begin{aligned}
\mathbb{E}||\hat{\mathbf{y}}_{t+1} - \mathbf{y}^*||^2 &\leq \mathbb{E}||\mathbf{z}_{t+1} - \mathbf{y}^*||^2 \\
&= \mathbb{E}||\hat{\mathbf{y}}_t - \eta_t G_t(\hat{\mathbf{y}}_t) - \mathbf{y}^*||^2 \\
&= \mathbb{E}||\hat{\mathbf{y}}_t - \mathbf{y}^*||^2 + \eta_t^2 \mathbb{E}||G_t(\hat{\mathbf{y}}_t)||^2 - 2\eta_t \mathbb{E}[G_{t_n}(\hat{\mathbf{y}}_t)(\hat{\mathbf{y}}_t - \mathbf{y}^*)],
\end{aligned} \tag{2.19}$$

where the first inequality follows the optimization (2.12) and the Pythagorean Theorem since

$$||\mathbf{z}_{t+1} - \mathbf{y}^*||^2 = ||\hat{\mathbf{y}}_{t+1} - \mathbf{y}^*||^2 + ||\mathbf{z}_{t+1} - \hat{\mathbf{y}}_{t+1}||^2$$

by property of the 2-norm projection; the first equality follows from the definition of $\mathbf{z}_{t+1}$; the second equality follows from a simple binomial expansion.

We can also re-write $\mathbb{E}[G_t(\hat{\mathbf{y}}_t)(\hat{\mathbf{y}}_t - \mathbf{y}^*)]$ by taking conditional expectation on the value of $\hat{\mathbf{y}}_t$,

$$\begin{aligned}
\mathbb{E}\left[G_t(\hat{\mathbf{y}}_t)(\hat{\mathbf{y}}_t - \mathbf{y}^*)\right] &= \mathbb{E}\left[\mathbb{E}\left[G_t(\hat{\mathbf{y}}_t)(\hat{\mathbf{y}}_t - \mathbf{y}^*)|\hat{\mathbf{y}}_t\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[G_t(\hat{\mathbf{y}}_t)|\hat{\mathbf{y}}_t\right](\hat{\mathbf{y}}_t - \mathbf{y}^*)\right] \\
&= \mathbb{E}\left[\nabla\Pi(\hat{\mathbf{y}}_t)(\hat{\mathbf{y}}_t - \mathbf{y}^*)\right],
\end{aligned} \tag{2.20}$$

where the first equality holds because $\mathbf{y}^*$ does not relate with $\hat{\mathbf{y}}_t$; the last equality follows from the fact that G_t is an unbiased estimator of the true gradient $\nabla\Pi$.

Combining (2.19) and (2.20), it is clear that

$$\mathbb{E}[\nabla\Pi(\hat{\mathbf{y}}_t)(\hat{\mathbf{y}}_t - \mathbf{y}^*)] \leq \frac{1}{2\eta_t}\left(\mathbb{E}||\hat{\mathbf{y}}_t - \mathbf{y}^*||^2 - \mathbb{E}||\hat{\mathbf{y}}_{t+1} - \mathbf{y}^*||^2\right) + \frac{\eta_t}{2}\mathbb{E}||G_t(\hat{\mathbf{y}}_t)||^2. \tag{2.21}$$

Without loss of generality, let $\mathcal{T} = \{\tau_0, \ldots, \tau_k\}$ with $\tau_0 = 0$ and $\tau_k = T$. By the

construction of DDM,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\Pi(\hat{\mathbf{y}}_t)-\sum_{t=1}^{T}\Pi(\mathbf{y}^*)\right] &= \mathbb{E}\left[\sum_{s=0}^{k-1}\sum_{t=\tau_s+1}^{\tau_{s+1}}(\Pi(\hat{\mathbf{y}}_t)-\Pi(\mathbf{y}^*))\right]\\ &\leq \frac{M}{l}\cdot\mathbb{E}\left[\sum_{s=1}^{k}(\Pi(\hat{\mathbf{y}}_{\tau_s})-\Pi(\mathbf{y}^*))\right],\end{aligned}$$

where the inequality follows from the fact that the time between any two consecutive break points cannot exceed the time for a "fictitious" system with M inventory units for each product $i = 1, \ldots, n$ to become empty along every sample path. The expectation of the latter (which is independent of $\hat{\mathbf{y}}_t$) is upper bounded by M/l.

It then suffices to bound the term $\mathbb{E}\left[\sum_{s=1}^{k}(\Pi(\hat{\mathbf{y}}_{\tau_s})-\Pi(\mathbf{y}^*))\right]$. Now, by summing both sides of (2.18) over periods τ_1 to τ_k,

$$\mathbb{E}\left[\sum_{s=1}^{k}(\Pi(\hat{\mathbf{y}}_{\tau_s})-\Pi(\mathbf{y}^*))\right] \leq \sum_{s=1}^{k}\mathbb{E}\left[\nabla\Pi(\hat{\mathbf{y}}_{\tau_s})(\hat{\mathbf{y}}_{\tau_s}-\mathbf{y}^*)\right] \tag{2.22}$$

$$\begin{aligned}&\leq \sum_{s=1}^{k}\left(\frac{1}{2\eta_{\tau_s}}\left(\mathbb{E}||\hat{\mathbf{y}}_{\tau_s}-\mathbf{y}^*||^2-\mathbb{E}||\hat{\mathbf{y}}_{\tau_s+1}-\mathbf{y}^*||^2\right)+\frac{\eta_{\tau_s}}{2}\mathbb{E}||G_{\tau_s}(\hat{\mathbf{y}}_{\tau_s})||^2\right)\\ &= \sum_{s=1}^{k}\left(\frac{1}{2\eta_{\tau_s}}\left(\mathbb{E}||\hat{\mathbf{y}}_{\tau_s}-\mathbf{y}^*||^2-\mathbb{E}||\hat{\mathbf{y}}_{\tau_{s+1}}-\mathbf{y}^*||^2\right)+\frac{\eta_{\tau_s}}{2}\mathbb{E}||G_{\tau_s}(\hat{\mathbf{y}}_{\tau_s})||^2\right)\\ &= \frac{1}{2\eta_{\tau_1}}\mathbb{E}||\hat{\mathbf{y}}_{\tau_1}-\mathbf{y}^*||^2-\frac{1}{2\eta_{\tau_k}}\mathbb{E}||\hat{\mathbf{y}}_{\tau_{k+1}}-\mathbf{y}^*||^2+\frac{1}{2}\sum_{s=2}^{k}\left(\frac{1}{\eta_{\tau_s}}-\frac{1}{\eta_{\tau_{s-1}}}\right)\mathbb{E}||\hat{\mathbf{y}}_{\tau_s}-\mathbf{y}^*||^2\\ &\quad+\sum_{s=1}^{k}\eta_{\tau_s}\frac{\mathbb{E}||G_{\tau_s}(\hat{\mathbf{y}}_{\tau_s})||^2}{2}\\ &\leq 2M^2\left(\frac{1}{2\eta_{\tau_1}}+\frac{1}{2}\sum_{s=2}^{k}\left(\frac{1}{\eta_{\tau_s}}-\frac{1}{\eta_{\tau_{s-1}}}\right)\right)+\frac{n(\max_i\{p^i-c^i,h^i\})^2}{2}\sum_{s=1}^{k}\eta_{\tau_s}\\ &= \frac{M^2}{\eta_{\tau_k}}+\frac{n(\max_i\{p^i-c^i,h^i\})^2}{2}\sum_{s=1}^{k}\eta_{\tau_s},\end{aligned}$$

where the first and second inequalities follows from (2.18) and (2.21), respectively; the first equality holds since $\hat{\mathbf{y}}_{\tau_s+1} = \hat{\mathbf{y}}_{\tau_{s+1}}$ by the construction of DDM; the last

inequality follows from the fact that for any $\mathbf{x}, \mathbf{y} \in \Gamma$,

$$||\mathbf{x} - \mathbf{y}||_2^2 \leq ||\mathbf{x}||_2^2 + ||\mathbf{y}||_2^2 \leq ||\mathbf{x}||_1^2 + ||\mathbf{y}||_1^2 \leq 2M^2.$$

Putting everything together, we have

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=1}^{T} \Pi(\hat{\mathbf{y}}_t) - \sum_{t=1}^{T} \Pi(\mathbf{y}^*)\right] \\ \leq \; & \frac{M}{l}\left(\frac{M^2}{\eta_T} + \frac{n(\max_i\{p^i - c^i, h^i\})^2}{2}\sum_{s=1}^{k}\eta_{\tau_s}\right). \end{aligned} \tag{2.23}$$

Note that we have chosen our step size "optimally" as

$$\eta_t = \left(\frac{\gamma M}{\sqrt{n} \cdot \max_i\{p^i - c^i, h^i\}}\right)\frac{1}{\sqrt{t}} \text{ for some } \gamma > 0,$$

so that

$$\begin{aligned} \sum_{s=1}^{k}\eta_{\tau_s} \leq \sum_{t=1}^{T}\eta_t &= \left(\frac{\gamma M}{\sqrt{n} \cdot \max_i\{p^i - c^i, h^i\}}\right)\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \\ &\leq \left(\frac{\gamma M}{\sqrt{n} \cdot \max_i\{p^i - c^i, h^i\}}\right)2\sqrt{T}. \end{aligned} \tag{2.24}$$

Plugging (2.24) and η_T into (2.23) yields the result with the constant term

$$K_1 = (\gamma + \gamma^{-1})M^2 l^{-1}\sqrt{n} \cdot \max_i\{p^i - c^i, h^i\}.$$

Note that putting $\gamma = 1$ gives the tightest bound. This completes the proof. □

2.4.2 Bound on Δ_2 - Stochastic Dominance and a GI/G/1 Queue (Proof of Lemma 2.12)

The main focus of this chapter is to establish the result in Lemma 2.12. First we derive a bound of the gap between the cost functions associated with the actual

implemented level $\mathbf{y_t}$ and the desired target level $\hat{\mathbf{y}}_t$, using the distance function $|\mathbf{y}_t - \hat{\mathbf{y}}_t|$.

Lemma 2.13. *The difference in cost functions*

$$\mathbb{E}[\Pi(\mathbf{y_t}) - \Pi(\hat{\mathbf{y}}_t)] \leq \mathbb{E}\left[(\mathbf{h} \vee (\mathbf{p} - \mathbf{c})) \cdot |\mathbf{y}_t - \hat{\mathbf{y}}_t|\right].$$

Proof. By the definition of the per-period cost function in (2.3), it follows that

$$\begin{aligned}
\mathbb{E}[\Pi(\mathbf{y}_t) - \Pi(\hat{\mathbf{y}}_t)] &\leq \mathbb{E}\left[\mathbf{c} \cdot (\mathbf{y}_t - \hat{\mathbf{y}}_t)\right] + \mathbb{E}\left[(\mathbf{h} - \mathbf{c}) \cdot (\mathbf{y}_t - \hat{\mathbf{y}}_t)^+\right] + \mathbb{E}\left[\mathbf{p} \cdot (\hat{\mathbf{y}}_t - \mathbf{y}_t)^+\right] \\
&= \mathbb{E}\left[\mathbf{h} \cdot (\mathbf{y}_t - \hat{\mathbf{y}}_t)^+\right] + \mathbb{E}\left[(\mathbf{p} - \mathbf{c}) \cdot (\hat{\mathbf{y}}_t - \mathbf{y}_t)^+\right] \\
&\leq \mathbb{E}\left[(\mathbf{h} \vee (\mathbf{p} - \mathbf{c}) \cdot |\mathbf{y}_t - \hat{\mathbf{y}}_t|\right],
\end{aligned}$$

where the last inequality follows from various operators defined at the end of Section 1. □

Given Lemma 2.13, we need to develop an upper bound on the distance function $|\mathbf{y}_t - \hat{\mathbf{y}}_t|$, which is the crux of our performance analysis. Lemmas 2.14 and 2.15 below play a major role in the development of such an upper bound. Their proof strategy relies heavily on the construction of DDM and also the structural properties of optimization problems (2.15) and (2.16), which is quite involved.

Lemma 2.14 below provides an upper bound on the distance function for products in the set J in which the beginning inventory level already exceeds the target order-up-to level.

Lemma 2.14. *In each period $t+1$, we bound the distance function for all $i \in J \triangleq \left\{i : x^i_{t+1} > \hat{y}^i_{t+1}\right\}$.*

$$\sum_{i \in J} |y^i_{t+1} - \hat{y}^i_{t+1}| \leq \sum_{i \in J} \left|y^i_t - \hat{y}^i_t\right| + \eta_t \left(\sum_{i \in J} h^i + \sum_{j \in \bar{J}} \left(p^j - c^j\right)\right) - \sum_{i \in J} d^i_t.$$

Proof. **Case 1.** We first consider time period $t \in \mathcal{T} = \{\tau_0, \ldots, \tau_k\}$, which belongs to the set of break points in DDM. Due to the construction of DDM, we update the target levels at $t+1$ only if $t \in \mathcal{T}$. For each product $i \in J$, i.e., $x^i_{t+1} > \hat{y}^i_{t+1}$, we have $y^i_{t+1} = x^i_{t+1} > \hat{y}^i_{t+1} \geq 0$ by (2.14). This implies that $y^i_{t+1} > 0$, and by the lost-sales system dynamics, we have

$$x^i_{t+1} = (y^i_t - d_t)^+ = y^i_t - d_t > 0. \tag{2.25}$$

The next key step is to compare the target level $\hat{y}^i_{t+1}$ with the constraint-free target level z^i_{t+1}. First, notice that when $y^i_t - d^i_t > 0$, the algorithm updates the constraint-free target level in a negative direction, i.e.,

$$z^i_{t+1} = \hat{y}^i_t - \eta_t h^i < \hat{y}^i_t. \tag{2.26}$$

Second, by the important property (2.17) of our algorithm, we have

$$\hat{y}^j_{t+1} \leq \hat{y}^j_t + \eta_t \left(p^j - c^j\right), \quad \forall\, j = 1, \ldots, n.$$

Thus, the maximum positive displacement of $\hat{\mathbf{y}}_{t+1}$ from $\hat{\mathbf{y}}_t$ (excluding the set J) is

$$\sum_{j \in \bar{J}} \left(\hat{y}^j_{t+1} - \hat{y}^j_t\right) \;\; \leq \;\; \sum_{j \in \bar{J}} \eta_t \left(p^j - c^j\right). \tag{2.27}$$

Now, to draw a relation between $\hat{y}^i_{t+1}$ and z^i_{t+1}, there are two cases.

Subcase 1a. In the first case where $\sum_{j=1}^n \hat{y}^j_{t+1} \geq \sum_{j=1}^n \hat{y}^j_t$, we must have

$$\sum_{i \in J} \left(z^i_{t+1} - \hat{y}^i_{t+1}\right) < \sum_{i \in J} \left(\hat{y}^i_t - \hat{y}^i_{t+1}\right) \leq \sum_{j \in \bar{J}} \left(\hat{y}^j_{t+1} - \hat{y}^j_t\right) \;\; \leq \;\; \sum_{j \in \bar{J}} \eta_t \left(p^j - c^j\right), \tag{2.28}$$

where the first inequality follows from (2.26); the second inequality follows from

$\sum_{j=1}^{n} \hat{y}_{t+1}^{j} \geq \sum_{j=1}^{n} \hat{y}_{t}^{j}$; and the third inequality follows from (2.27).

Subcase 1b. In the second case where $\sum_{j=1}^{n} \hat{y}_{t+1}^{j} < \sum_{j=1}^{n} \hat{y}_{t}^{j} \leq M$, the warehouse storage constraint is not tight (i.e., the constraint-free target levels are in the interior of Γ), and by the optimization procedure (2.15), $z_{t+1}^{j} = \hat{y}_{t+1}^{j}$ for all $j = 1, \ldots, n$. Thus, we have

$$z_{t+1}^{i} - \hat{y}_{t+1}^{i} = \hat{y}_{t+1}^{i} - \hat{y}_{t+1}^{i} = 0, \quad i = 1, \ldots, n. \tag{2.29}$$

Combining the above two cases and using the relations (2.28) and (2.29), we can then obtain an upper bound for our distance function as follows,

$$\begin{aligned}
\sum_{i \in J} |y_{t+1}^{i} - \hat{y}_{t+1}^{i}| &= \sum_{i \in J} \left(x_{t+1}^{i} - \hat{y}_{t+1}^{i}\right) = \sum_{i \in J} \left(y_{t}^{i} - d_{t}^{i} - \hat{y}_{t+1}^{i}\right) \\
&\leq \sum_{i \in J} \left(y_{t}^{i} - d_{t}^{i} - z_{t+1}^{i}\right) + \sum_{j \in \bar{J}} \eta_{t} \left(p^{j} - c^{j}\right) \\
&= \sum_{i \in J} \left(y_{t}^{i} - \hat{y}_{t}^{i}\right) + \eta_{t} \left(\sum_{i \in J} h^{i} + \sum_{j \in \bar{J}} \left(p^{j} - c^{j}\right) \right) - \sum_{i \in J} d_{t}^{i}, \\
&\leq \sum_{i \in J} \left|y_{t}^{i} - \hat{y}_{t}^{i}\right| + \eta_{t} \left(\sum_{i \in J} h^{i} + \sum_{j \in \bar{J}} \left(p^{j} - c^{j}\right) \right) - \sum_{i \in J} d_{t}^{i},
\end{aligned}$$

where the first equality follows from the fact that $i \in J$ and the construction of our algorithm (2.14); the second equality is due to (2.25); the first inequality follows from (2.28) and (2.29), and the third equality follows from (2.26). Now we have completed the proof for Case 1.

Case 2. We then consider time period $t \notin \mathcal{T}$, which does not belong to the set of break points in DDM. According to the construction of DDM, the target order-up-to levels are kept unchanged, i.e., $\hat{y}_{t+1}^{i} = \hat{y}_{t}^{i}$ for all $i = 1, \ldots, n$ and for all $t \notin \mathcal{T}$. we can

similarly obtain an upper bound for our distance function as follows,

$$\begin{aligned}\sum_{i\in J}|y^i_{t+1}-\hat{y}^i_{t+1}| &= \sum_{i\in J}\left(x^i_{t+1}-\hat{y}^i_{t+1}\right)=\sum_{i\in J}\left(y^i_t-d^i_t-\hat{y}^i_{t+1}\right)\\ &= \sum_{i\in J}\left(y^i_t-d^i_t-\hat{y}^i_t\right)\le\sum_{i\in J}\left|y^i_t-\hat{y}^i_t\right|-\sum_{i\in J}d^i_t,\end{aligned}$$

where the first equality follows from the fact that $i \in J$ and the construction of our algorithm (2.14); the second equality is due to (2.25); the third equality follows from $\hat{y}^i_{t+1} = \hat{y}^i_t$. Now we have completed the proof for Case 2. □

Lemma 2.15 below provides an upper bound on the distance function for products in the complement set $\bar{J}$ in which the beginning inventory level is below the target order-up-to level. If this is the case for all products, i.e., all products belong to $\bar{J}$, then the target levels can always be achieved. If not, we solve (2.15) to re-distribute our target levels such that the difference between the target level and the actual implemented level is as even as possible across different products.

Lemma 2.15. *In each period $t+1$, we bound the distance function for all $i \in \bar{J} \triangleq \left\{i : x^i_{t+1} \le \hat{y}^i_{t+1}\right\}$ as follows. If $J = \emptyset$, we have $\sum_{i\in\bar{J}}|\hat{y}^i_{t+1} - y^i_{t+1}| = 0$. Otherwise, if $J \neq \emptyset$, we have*

$$\sum_{i\in\bar{J}}|\hat{y}^i_{t+1}-y^i_{t+1}|\le\sum_{i\in\bar{J}}\left|\hat{y}^i_t-y^i_t\right|+\eta_t\sum_{i\in\bar{J}}(p^i-c^i)-\sum_{j\in J}d^j_t.$$

Proof. **Case 1.** We first consider time period $t \in \mathcal{T} = \{\tau_0, \ldots, \tau_k\}$, which belongs to the set of break points in DDM. Due to the construction of DDM, we update the target levels at $t+1$ only if $t \in \mathcal{T}$. For each product $i \in \bar{J}$, i.e., $x^i_{t+1} \le \hat{y}^i_{t+1}$, recall that we need to solve the optimization problem (2.15) to determine our actual implemented levels $\mathbf{y}_{t+1}$. That is,

$$\min\sum_{i\in\bar{J}}(\hat{y}^i_{t+1}-y^i_{t+1})^2 \quad \text{s.t.} \quad \sum_{i\in\bar{J}}y^i_{t+1}\le M-\sum_{j\in J}x^j_{t+1}, \quad y^i_{t+1}\ge x^i_{t+1},\ \forall\, i\in\bar{J}.$$

It is straightforward to see that $\hat{y}^i_{t+1} \geq y^i_{t+1}$ for each product $j \in \bar{J}$. To see this, suppose otherwise $\hat{y}^i_{t+1} < y^i_{t+1}$; we can always lower the value of y^i_{t+1} to $\hat{y}^i_{t+1}$ strictly improving the objective value while maintaining feasibility.

Now there are three sub-cases.

Subcase 1a. The simplest case is when $J = \emptyset$, then (2.15) reduces to

$$\min \sum_{i=1}^{n} (\hat{y}^i_{t+1} - y^i_{t+1})^2 \quad \text{s.t.} \quad \sum_{i \in \bar{J}} y^i_{t+1} \leq M, \quad y^i_{t+1} \geq x^i_{t+1}, \; \forall \, i \in \bar{J}.$$

Since $\hat{\mathbf{y}}_{t+1} \in \Gamma$, we have $y^i_{t+1} = \hat{y}^i_{t+1}$ for each product $i = 1, \ldots, n$, and thus the distance function is zero for each product $i = 1, \ldots, n$.

Subcase 1b. The second case is when upon solving $\mathbf{y}_{t+1}$, the warehouse storage constraint is not tight, i.e.,

$$\sum_{i \in \bar{J}} y^i_{t+1} < M - \sum_{j \in J} x^j_{t+1}. \tag{2.30}$$

Then we claim that

$$\sum_{i \in \bar{J}} \hat{y}^i_{t+1} < M - \sum_{j \in J} x^j_{t+1}, \tag{2.31}$$

We argue the claim by contradiction. Suppose otherwise that

$$\sum_{i \in \bar{J}} \hat{y}^i_{t+1} \geq M - \sum_{j \in J} x^j_{t+1} > \sum_{i \in \bar{J}} y^i_{t+1}.$$

Then there must exist a product k such that $y^k_{t+1} < \hat{y}^k_{t+1}$, you can always increase y^k_{t+1} by

$$\epsilon \triangleq M - \sum_{j \in J} x^j_{t+1} - \sum_{i \in \bar{J}} y^i_{t+1} > 0$$

to make the warehouse storage constraint tight, thereby strictly reducing the optimal objective value. This contradicts the optimality of $\mathbf{y}_{t+1}$ in (2.15).

Thus, by (2.30) and (2.31), we have $y^i_{t+1} = \hat{y}^i_{t+1}$ for each product $i \in \bar{J}$, and the

distance function is zero for each product $i = 1, \ldots, n$.

Subcase 1c. The third case is much more involved. That is, upon solving $\mathbf{y}_{t+1}$, the warehouse storage constraint becomes tight, i.e.,

$$\sum_{i \in \bar{J}} y_{t+1}^i = M - \sum_{j \in J} x_{t+1}^j,$$

and the set $J \neq \emptyset$. We can then rewrite the optimization problem (2.15) as follows,

$$\begin{aligned} \min \quad & \sum_{i \in \bar{J}} (\hat{y}_{t+1}^i - y_{t+1}^i)^2 \\ \text{s.t.} \quad & \sum_{i \in \bar{J}} \left(\hat{y}_{t+1}^i - y_{t+1}^i\right) = \sum_{i \in \bar{J}} \hat{y}_{t+1}^i - M + \sum_{j \in J} x_{t+1}^j, \quad y_{t+1}^i \geq x_{t+1}^i, \ \forall\, i \in \bar{J}. \end{aligned}$$

We then bound the distance function as follows,

$$\begin{aligned} \sum_{i \in \bar{J}} \left|\hat{y}_{t+1}^i - y_{t+1}^i\right| &= \sum_{i \in \bar{J}} \hat{y}_{t+1}^i - M + \sum_{j \in J} x_{t+1}^j = \sum_{i \in \bar{J}} \hat{y}_{t+1}^i - M + \sum_{j \in J} (y_t^j - d_t^j)^+ \\ &= \sum_{i \in \bar{J}} \hat{y}_{t+1}^i - M + \sum_{j \in J} (y_t^j - d_t^j) \\ &= \sum_{i \in \bar{J}} \hat{y}_{t+1}^i - \left(M - \sum_{j \in J} y_t^j\right) - \sum_{j \in J} d_t^j \\ &\leq \sum_{i \in \bar{J}} \hat{y}_{t+1}^i - \sum_{i \in \bar{J}} y_t^i - \sum_{j \in J} d_t^j = \sum_{i \in \bar{J}} \left(\hat{y}_{t+1}^i - y_t^i\right) - \sum_{j \in J} d_t^j \\ &\leq \sum_{i \in \bar{J}} \left(\hat{y}_t^i + \eta_t (p^i - c^i) - y_t^i\right) - \sum_{j \in J} d_t^j \\ &\leq \sum_{i \in \bar{J}} \left(\hat{y}_t^i - y_t^i\right) + \sum_{i \in \bar{J}} \eta_t (p^i - c^i) - \sum_{j \in J} d_t^j \\ &\leq \sum_{i \in \bar{J}} \left|\hat{y}_t^i - y_t^i\right| + \eta_t \sum_{i \in \bar{J}} (p^i - c^i) - \sum_{j \in J} d_t^j, \end{aligned}$$

where the first equality is because the warehouse storage constraint becomes tight; the second equality is due to the system dynamics; the third equality is because $j \in J$ implies that $x_{t+1}^j > 0$, and hence the plus sign can be removed; the first inequality is

due to the fact that

$$\sum_{j\in\bar{J}} y_t^j + \sum_{j\in J} y_t^j = \sum_{j=1}^{n} y_t^j \le M;$$

and the second inequality follows from the important property (2.17) of our algorithm. Now we have completed the proof for Case 1.

Case 2. We then consider time period $t \notin \mathcal{T}$, which does not belong to the set of break points in DDM. Due to the construction of DDM, the target order-up-to levels are kept unchanged, i.e., $\hat{y}_{t+1}^i = \hat{y}_t^i$ for all $i = 1, \ldots, n$ and for all $t \notin \mathcal{T}$. Also, if $t \notin \mathcal{T}$, then the optimization problem (2.15) has a nonzero objective value, which suggests that the warehouse storage constraint has to be tight. We then similarly bound the distance function as follows,

$$\begin{aligned}
\sum_{i\in\bar{J}} \left|\hat{y}_{t+1}^i - y_{t+1}^i\right| &= \sum_{i\in\bar{J}} \hat{y}_{t+1}^i - M + \sum_{j\in J} x_{t+1}^j = \sum_{i\in\bar{J}} \hat{y}_{t+1}^i - M + \sum_{j\in J} (y_t^j - d_t^j)^+ \\
&= \sum_{i\in\bar{J}} \hat{y}_{t+1}^i - M + \sum_{j\in J} (y_t^j - d_t^j) \\
&= \sum_{i\in\bar{J}} \hat{y}_{t+1}^i - \left(M - \sum_{j\in J} y_t^j\right) - \sum_{j\in J} d_t^j \\
&\le \sum_{i\in\bar{J}} \hat{y}_{t+1}^i - \sum_{i\in\bar{J}} y_t^i - \sum_{j\in J} d_t^j = \sum_{i\in\bar{J}} \left(\hat{y}_{t+1}^i - y_t^i\right) - \sum_{j\in J} d_t^j \\
&= \sum_{i\in\bar{J}} \left(\hat{y}_t^i - y_t^i\right) - \sum_{j\in J} d_t^j \le \sum_{i\in\bar{J}} \left|\hat{y}_t^i - y_t^i\right| - \sum_{j\in J} d_t^j,
\end{aligned}$$

where we used the same arguments as in Subcase 1c and also the fact that $\hat{y}_{t+1}^i = \hat{y}_t^i$ for all $i = 1, \ldots, n$ if $t \notin \mathcal{T}$. Now we have completed the proof for Case 2. □

With the upper bounds on the distance function in two mutually exclusive sets J and $\bar{J}$ obtained from Lemmas 2.14 and 2.15, we provide an overarching upper bound in Lemma 2.16.

Lemma 2.16. *In each period $t+1$, we bound the sum of distance functions as follows.*

$$\sum_{i=1}^{n} |y_{t+1}^i - \hat{y}_{t+1}^i| \leq \left(\sum_{i=1}^{n} \left| y_t^i - \hat{y}_t^i \right| + \eta_t \left(\sum_{i=1}^{n} \left(h^i + 2\left(p^i - c^i \right) \right) \right) - \min_{j=1,\ldots,n} d_t^j \right)^+ .$$

Proof. By Lemma 2.14, we have

$$\sum_{i \in J} |y_{t+1}^i - \hat{y}_{t+1}^i| \leq \left(\sum_{i \in J} \left| y_t^i - \hat{y}_t^i \right| + \eta_t \left(\sum_{i \in J} h^i + \sum_{j \in \bar{J}} \left(p^j - c^j \right) \right) - \sum_{i \in J} d_t^i \right)^+ ,$$

and by Lemma 2.15, we have

$$\sum_{i \in \bar{J}} |y_{t+1}^i - \hat{y}_{t+1}^i| \leq \left(\sum_{i \in \bar{J}} \left| y_t^i - \hat{y}_t^i \right| + \eta_t \sum_{i \in \bar{J}} (p^i - c^i) - \min_{j=1,\ldots,n} d_t^j \right)^+ .$$

Combining the above two inequalities yields the result. □

Next, we wish to find a stochastic process that can be used to bound the sum of distance functions. It is now convenient to introduce the notion of *stochastic order* and *convex order* (see *Shaked and Shanthikumar* (2007)). Consider two random variables X and Y. X is said to be stochastically smaller than Y (denoted by $X \leq_{st} Y$) if $\mathbb{P}(X > x) \leq \mathbb{P}(Y > x), \forall x \in \mathbb{R}$. Also, X is said to be smaller than Y in the convex order (denoted as $X \leq_{cx} Y$) if $\mathbb{E}[\phi(X)] \leq \mathbb{E}[\phi(Y)]$ for all convex functions $\phi : \mathbb{R} \to \mathbb{R}$, provided the expectations exist. Note that convex order is weaker, i.e., $X \leq_{st} Y \Rightarrow X \leq_{cx} Y$.

Next, corresponding to the sum of distance functions, we consider a stochastic process $(Z_t \mid t \geq 0)$

$$Z_{t+1} = \left[Z_t + \frac{S_t}{\sqrt{t}} - \tilde{D}_t \right]^+, \qquad Z_0 = 0,$$

where $S_t \triangleq \sum_{i=1}^{n} \left(h^i + 2(p^i - c^i) \right)$, and $\tilde{D}_t$ is a random variable satisfying $\tilde{D}_t \leq_{st} D_t^j, \ \forall j$.

Lemma 2.17. *The total expected distance function*

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n}|y_t^i-\hat{y}_t^i|\right]\leq\mathbb{E}\left[\sum_{t=1}^{T}Z_t\right],$$

where Z_{t+1}^i is a stochastic process defined above.

Proof. By Lemma 2.16, for each period $t+1$, the sum of distance functions

$$\sum_{i=1}^{n}|y_{t+1}^i-\hat{y}_{t+1}^i|\leq\left(\sum_{i=1}^{n}\left|y_t^i-\hat{y}_t^i\right|+\eta_t\sum_{i=1}^{n}\left(h^i+2\left(p^i-c^i\right)\right)-\min_{j=1,\ldots,n}d_t^j\right)^+.$$

In addition, we know that $\sum_{i=1}^{n}|y_0^i-\hat{y}_0^i|=0$ (since the policy starts with zero inventory). Thus, by the definition of the stochastic process Z_{t+1}, it is clear that $\sum_{i=1}^{n}|y_{t+1}^i-\hat{y}_{t+1}^i|\leq_{st}Z_{t+1}$. This implies that $\sum_{i=1}^{n}|y_{t+1}^i-\hat{y}_{t+1}^i|\leq_{cx}Z_{t+1}$, and then the result follows immediately. □

We observe that the stochastic process Z_t is very similar to a $GI/G/1$ queue, except that the service time is scaled by $1/\sqrt{t}$ in each period t. Now consider a $GI/G/1$ queue $(W_n \mid n\geq 0)$ defined by the following Lindley's equation: $W_0=0$, and

$$W_{t+1}=[W_t+S_t-\tilde{D}_t]^+, \tag{2.32}$$

where the sequences S_t and $\tilde{D}_t$ consist of independent and identically distributed random variables. Let $\tau_0=0$, $\tau_1=\inf\{t\geq 1:W_t=0\}$ and for $k\geq 1$, $\tau_{k+1}=\inf\{t>\tau_k:W_t=0\}$. Let $B_k=\tau_k-\tau_{k-1}$. The random variable W_t is the waiting time of the t^{th} customer in the $GI/G/1$ queue, where the inter-arrival time between the t^{th} and $t+1^{th}$ customers is distributed as $\tilde{D}_t$, and the service time is distributed as S_t. Then, B_k is the length of the k^{th} busy period. Let $\rho=\mathbb{E}[S_1]/\mathbb{E}[\tilde{D}_1]$ represent the system utilization. It is well-known that in a $GI/G/1$ queue, if $\rho\leq 1$, then the queue is stable and the random variable B_k is independent and identically distributed. Note that this stability condition $\rho\leq 1$ can always be satisfied by appropriately scaling

the units of cost parameters.

We invoke the following result from *Loulou* (1978) to bound $\mathbb{E}[B]$, the expected busy period of a $GI/G/1$ queue with inter-arrival distribution D_n and service distribution S_n.

Theorem 2.18 (Loulou 1978)**.** *Let $X_n = S_n - D_n$, and $\alpha = -\mathbb{E}[X_1]$. Let σ^2 be the variance of X_1. If $\mathbb{E}[X_1^3] = \beta < \infty$, and $\rho < 1$,*

$$\mathbb{E}[B] \leq \frac{\sigma}{\alpha} \exp\left(\frac{6\beta}{\sigma^3} + \frac{\alpha}{\sigma}\right).$$

We can now obtain an upper bound on our expected busy period $\mathbb{E}[B]$ for the stochastic process W_t defined in (2.32), by setting $X_1 = \sum_{i=1}^{n} \left(h^i + 2(p^i - c^i)\right) - \tilde{D}_1$ (whose expectation is negative since $\rho \leq 1$).

With the explicit form of $\mathbb{E}[B]$, Lemma 2.19 gives the upper bound of our distance function below. The idea is to connect the upper-bounding stochastic process (which evolves as a $GI/G/1$ queue) with the expected busy period of this queue (where there exists an explicit upper bound that does not depend on the time horizon T).

Lemma 2.19. *The total expected distance function*

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n} |y_t^i - \hat{y}_t^i|\right] \leq 2\mathbb{E}[B]S\sqrt{T},$$

where $\mathbb{E}[B] \leq \frac{\sigma}{\alpha} \exp\left(\frac{6\beta}{\sigma^3} + \frac{\alpha}{\sigma}\right)$, and $S = \sum_{i=1}^{n} \left(h^i + 2\left(p^i - c^i\right)\right)$.

Proof. By Lemma 2.17, it suffices to show that $\mathbb{E}\left[\sum_{t=1}^{T} Z_t\right] \leq 2\mathbb{E}[B]S\sqrt{T}$. Recall that

$$Z_{t+1} = \left[Z_t + \frac{S_t}{\sqrt{t}} - \tilde{D}_t\right]^+, \qquad Z_0 = 0.$$

Let the random variable $l(t)$ denote the index k in which B_k contains t, and it is clear

that

$$Z_t \leq \sum_{s=1}^{t} \frac{S_s}{\sqrt{s}} \mathbb{1}\left[s \in B_{l(t)}\right] \text{ a.s.}$$

By summing Z_t over periods 1 to T and taking expectation, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} Z_t\right] &\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{s=1}^{t} \frac{S_s}{\sqrt{s}} \mathbb{1}\left[s \in B_{l(t)}\right]\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \frac{S_t}{\sqrt{t}} \sum_{s=1}^{T} \mathbb{1}\left[s \in B_{l(t)}\right]\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \frac{S_t}{\sqrt{t}} B_{l(t)}\right] = \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \mathbb{E}[B_1] S \leq 2\mathbb{E}[B] S \sqrt{T}.
\end{aligned}$$

This completes the proof. □

The proof of Lemma 2.12 then follows from Lemma 2.13 and Lemma 2.19.

Proof. Combining Lemma 2.13 and Lemma 2.19, we have

$$\begin{aligned}
\Delta_2(T) &= \mathbb{E}\left[\sum_{t=1}^{T} \left(\Pi(\mathbf{y}_t) - \Pi(\hat{\mathbf{y}}_t)\right)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} (\mathbf{h} \vee (\mathbf{p} - \mathbf{c})) \cdot |\mathbf{y}_t - \hat{\mathbf{y}}_t|\right] \\
&\leq \max_i\{p^i - c^i, h^i\} \mathbb{E}\left[\sum_{t=1}^{T} |\mathbf{y}_t - \hat{\mathbf{y}}_t|\right] = \max_i\{p^i - c^i, h^i\} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n} |y_t^i - \hat{y}_t^i|\right] \\
&\leq \max_i\{p^i - c^i, h^i\} \left(2\sqrt{T}\mathbb{E}[B] S\right) = \left(2 \max_i\{p^i - c^i, h^i\} \mathbb{E}[B] S\right) \sqrt{T}.
\end{aligned}$$

Recall that $\mathbb{E}[B] \leq \frac{\sigma}{\alpha} e^{\frac{6\beta}{\sigma^3} + \frac{\alpha}{\sigma}}$ and $S = \sum_{i=1}^{n} \left(h^i + 2\left(p^i - c^i\right)\right)$. Setting the constant

$$K_2 = 2 \max_i\{p^i - c^i, h^i\} \frac{\sigma}{\alpha} e^{\frac{6\beta}{\sigma^3} + \frac{\alpha}{\sigma}} \left\{\sum_{i=1}^{n} \left(h^i + 2\left(p^i - c^i\right)\right)\right\}.$$

yields the result. This completes the proof. □

2.5 Extensions

2.5.1 Improving the convergence rate

If we change Assumption 2.1(c) slightly to enforce a uniform lower bound $\delta > 0$ on the density of demand, i.e., $F'_{D^i}(x) \geq \delta > 0$ for all $x \in [0, M]$ and all $i = 1, \ldots, n$, and also change the step size $\eta_t = O(1/t)$ in the algorithm, one can readily show that the cost function is δ-strongly convex, and the rate of convergence of DDM can be improved to $O(\log T/T)$.

2.5.2 Different Product Dimensions or Sizes

Our basic model (defined in Section 2.2) assumes that all products have exactly the same dimension or sizes. However, in general, different products may have different dimension or sizes. Let $v^1, v^2, \ldots, v^n$ denote the sizes of the different products, and

$$\mathbf{y}_t \in \Gamma \triangleq \left\{ \mathbf{y}_t \in \mathbb{R}^n_+ : \sum_{i=1}^{n} v^i y^i_t \leq M \right\}, \tag{2.33}$$

By a simple cost transformation, we show in the following that our algorithm DDM (now defined in terms of transformed variables) and its performance analysis remain the same.

We define new decision variables as follows,

$$\tilde{y}^i_t = v^i y^i_t, \qquad \tilde{x}^i_t = v^i x^i_t, \qquad \tilde{q}^i_t = v^i q^i_t,$$

for $i = 1, \ldots, n$. In addition, we appropriately scale the demand and cost parameters as follows,

$$\tilde{D}^i_t = v^i D^i_t, \qquad \tilde{c}^i = c^i/v^i, \qquad \tilde{h}^i = h^i/v^i, \qquad \tilde{p}^i = p^i/v^i$$

for $i = 1, \ldots, n$. With the above transformation, the cost of a feasible policy π under the new warehouse-capacity constraint (2.33) can be transformed as follows,

$$\begin{aligned} \mathcal{C}(\pi) &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{c}\cdot(\mathbf{y}_t - \mathbf{x}_t) + \mathbf{h}\cdot(\mathbf{y}_t - \mathbf{D}_t)^+ + \mathbf{p}\cdot(\mathbf{D}_t - \mathbf{y}_t)^+\right] - \mathbb{E}[\mathbf{c}\cdot\mathbf{x}_{T+1}], \quad (2.34)\\ &= \mathbb{E}\left[\sum_{t=1}^{T} \tilde{\mathbf{c}}\cdot(\tilde{\mathbf{y}}_t - \tilde{\mathbf{x}}_t) + \tilde{\mathbf{h}}\cdot(\tilde{\mathbf{y}}_t - \tilde{\mathbf{D}}_t)^+ + \tilde{\mathbf{p}}\cdot(\tilde{\mathbf{D}}_t - \tilde{\mathbf{y}}_t)^+\right] - \mathbb{E}[\tilde{\mathbf{c}}\cdot\tilde{\mathbf{x}}_{T+1}]. \end{aligned}$$

Moreover, it is clear that the new constraint defined in (2.33) is equivalent to

$$\tilde{\mathbf{y}}_t \in \Gamma \triangleq \left\{\tilde{\mathbf{y}}_t \in \mathbb{R}_+^n : \sum_{i=1}^{n} \tilde{y}_t^i \le M\right\},$$

which has the same form as in the original constraint defined in (2.1). Hence, this more general model has been reduced to the basic model. Our data-driven algorithm (now defined in terms of transformed variables) and its performance analysis remain the same.

2.5.3 Discrete Demand and Ordering Quantities

In practice, the demand and ordering quantities are often integers. We provide a modified algorithm (denoted by DDM-Discrete) in the following to handle such discrete cases, which achieves the same convergence rate $O(1/\sqrt{T})$ with the aid of lost-sales indicators (i.e., the firm knows whether lost-sales has occurred in each period).

DDM-Discrete

Step 0. (Initialization.) Set the initial inventory levels $\mathbf{y}_0 = \hat{\mathbf{y}}_0 = \bar{\mathbf{y}}_0$ to be any non-negative integer values within Γ and then set the initial values $t = 0$, $\tau_0 = 0$ and $k = 0$.

For each period $t = 0, \ldots, T-1$, repeat the following steps:

Step 1. (Setting the constraint-free and constrained target inventory levels.)

Case 1: If $\mathbf{y}_t \ge \hat{\mathbf{y}}_t$, the algorithm updates the constraint-free target inventory

levels $\mathbf{z}_{t+1}$ by (2.10); however, for each product $i = 1, \ldots, n$, the i^{th} component of $\hat{\mathbf{G}}_t$ is defined as

$$\hat{G}_t^i(\hat{\mathbf{y}}_t) = \begin{cases} -p^i + c_t^i + (h^i + p^i - c^i) \cdot \mathbb{1}(d_t^i \leq \hat{y}_t^i), & \text{if } \hat{y}_t^i = \lfloor \bar{y}_t^i \rfloor, \\ -p^i + c_t^i + (h^i + p^i - c^i) \cdot \mathbb{1}(d_t^i \leq \hat{y}_t^i - 1), & \text{if } \hat{y}_t^i = \lceil \bar{y}_t^i \rceil, \end{cases} \tag{2.35}$$

which is the right derivative of Π at $\lfloor \bar{y}_t^i \rfloor$, i.e., the slope of Π at $\bar{y}_t^i$ for the piece-wise linear cost.

Then the algorithm obtains an intermediate (continuous) target level $\bar{\mathbf{y}}_{t+1}$ by solving $\bar{\mathbf{y}}_{t+1} = \arg\min_{\mathbf{w} \in \Gamma} ||\mathbf{w} - \mathbf{z}_{t+1}||_2$. Then set the constrained (discrete) target inventory levels $\hat{\mathbf{y}}_{t+1}$ by probabilistic rounding. That is, if $\bar{y}_{t+1}^i$ is already an integer, set $\hat{y}_{t+1}^i = \bar{y}_{t+1}^i$; otherwise, we flip a (biased) coin with probability $\hat{y}_{t+1}^i - \lfloor \bar{y}_{t+1}^i \rfloor$ of heads. Set $\hat{y}_{t+1}^i = \lceil \bar{y}_{t+1}^i \rceil$ if the outcome is head and set $\hat{y}_{t+1}^i = \lfloor \bar{y}_{t+1}^i \rfloor$ if the outcome is tail.

Record the break point $\tau_k := t + 1$ and increase the value k by 1.

Case 2: Else if $\mathbf{y}_t \not\geq \hat{\mathbf{y}}_t$, the algorithm keeps both the constraint-free and constrained target inventory levels unchanged, i.e., $\mathbf{z}_{t+1} = \mathbf{z}_t$ and $\hat{\mathbf{y}}_{t+1} = \hat{\mathbf{y}}_t$.

Step 2. (Solving for the actual implemented target inventory levels.)

Define the set J as $J \triangleq \left\{ i : x_{t+1}^i > \hat{y}_{t+1}^i \right\}$. Define set J's complement as $\bar{J} \triangleq \left\{ i : x_{t+1}^i \leq \hat{y}_{t+1}^i \right\}$. For each product $i \in J$, we set the actual implemented levels $y_{t+1}^i = x_{t+1}^i$ if $x_{t+1}^i > \hat{y}_{t+1}^i$.

If $\bar{J} \neq \emptyset$, then we set the actual implemented levels $\mathbf{y}_{t+1}$ by solving

$$\begin{aligned} \min \quad & \sum_{i \in \bar{J}} (\hat{y}_{t+1}^i - y_{t+1}^i)^2 \\ \text{s.t.} \quad & \sum_{i \in \bar{J}} y_{t+1}^i \leq M - \sum_{j \in J} x_{t+1}^j, \; y_{t+1}^i \geq x_{t+1}^i, \; y_{t+1}^i \in \mathbb{Z}^+, \; \forall \, i \in \bar{J}. \end{aligned} \tag{2.36}$$

This concludes the description of the algorithm.

Note that the key differences between DDM-Discrete and DDM are in Step 1 – defining a modified gradient $\hat{\mathbf{G}}_t$, and probabilistic rounding. In order to establish our performance guarantee, we need a *lost-sales indicator* for whether lost-sales has occurred in each period t, thereby determining the value of $\mathbb{1}(\hat{y}_t^i \leq d_t^i)$. Without this indicator, it is not sufficient to obtain an unbiased estimator for the right derivative of our cost function $\Pi(\cdot)$ by observing the past sales quantities. The decision maker no longer has access to a local (stochastic) direction of cost improvement. For example, if the computed target inventory level is 15.5 for some product i and we round it down to 15, even an infinite number of sales observations would not allow the decision maker to obtain an estimate of the slope of $\Pi^i(\cdot)$ at 15.5. This is because if the demand turns out to be exactly 15, the unbiased gradient should be h^i since our target level 15.5 is higher than 15; however, if the demand turns out to be 16, then the unbiased gradient should be $-p^i + c^i$. In both cases, we observe zero inventory but cannot determine if the demand is strictly greater than 15 without a lost-sales indicator.

However, with access to this lost-sales indicator, we can construct such an estimator $\hat{\mathbf{G}}_t$ defined in (2.35), which is unbiased when $t \in \mathcal{T}$. Then the proofs of bounds Δ_1 and Δ_2 are almost identical to the ones used in DDM as long as the warehouse-capacity M is also an integer. Hence we are able to extend our results to the discrete demand and inventory case as stated in the theorem below.

Theorem 2.20. *Assume that the clairvoyant optimal solution in* (2.6) *is unique in the discrete demand case. With access to the lost-sales indicator, the average regret* $\mathcal{R}_T/T$ *of our data-driven algorithm for discrete demand case (DDM-Discrete) approaches* 0 *at the rate of* $1/\sqrt{T}$*. That is, there exists some constant* K*, such that for any* $T \geq 1$*,*

$$\frac{1}{T}\mathcal{R}_T \triangleq \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\Pi(\mathbf{y}_t)\right] - \Pi(\mathbf{y}^*) \leq \frac{K}{\sqrt{T}}.$$

To the best of our knowledge, the availability of the lost-sales indicator has been

assumed in all nonparametric studies that analyze newsvendor-type problems with an unknown discrete demand distribution (see, e.g., *Huh and Rusmevichientong* (2009) and *Besbes and Muharremoglu* (2013)). They showed that active exploration plays a much stronger role in the discrete case (compared to the continuous case). However, the need for active exploration disappears as soon as a lost-sales indicator (that records whether demand was censored or not) becomes available, in addition to the censored demand samples. The access to this indicator allows the decision maker to obtain a noisy signal about the potential need for an upward correction.

2.6 Numerical Experiments

We compare the performance of DDM with several existing parametric and nonparametric approaches in the literature (briefly described below). Our results show that DDM outperforms these benchmark algorithms in terms of both consistency and convergence rate. We first explain the detailed experimental setup and then show numerical results (figures) with benchmarks.

2.6.1 Experimental Setup

For each experiment, we specify a (hindsight) demand distribution with cumulative distribution function $F(\cdot)$. The lost-sales penalty cost p^i for each product i is randomly drawn from the interval $[70, 90]$ and the purchasing cost c^i for each product i is randomly drawn from the interval $[55, 65]$. We then set the holding cost $h^i = 0.02c^i$ for each product i (see, e.g., *Zipkin* (2000)). To compare the cost under each algorithm, we evaluate each algorithm on $N = 200$ randomly generated problem instances. Each problem instance consists of independent demand samples and parameters over a time horizon of 500 periods, unless specified otherwise. For each

algorithm π, we compute the average cost till period t, which is given by

$$\frac{1}{N}\sum_{j=1}^{N}\frac{1}{t}\sum_{s=1}^{t}\tilde{\Pi}_{j,s}(y_{j,s}^{\pi}),$$

where the one-period cost $\tilde{\Pi}_{j,s}(y)$ in period s of the problem instance j is given by

$$\tilde{\Pi}_{j,s}(y)=\sum_{i=1}^{n}c^{i}y_{j,s}^{i,\pi}+(h^{i}-c^{i})(y_{j,s}^{i,\pi}-d_{j,s}^{i})^{+}+p^{i}(d_{j,s}^{i}-y_{j,s}^{i,\pi})^{+},$$

where $d_{j,s}^{i}$ is the demand realization for product i in period s of the problem instance j, and $y_{j,s}^{i,\pi}$ is the corresponding order-up-to level computed by each algorithm π.

2.6.2 Benchmarks and Numerical Results

(i). **Algorithm a1 (Known Distribution): Clairvoyant Optimal Policy.**

(ii). **Algorithm a2 (Uncensored): Uncensored SAA.** This is a sample average approximation (SAA) algorithm with uncensored demand (a hypothetical situation). The target inventory level is the quantile of the empirical demand distribution using uncensored demand data.

(iii). **Algorithm b1 (Parametric): MLE Censored.** Assuming the correct parametric form has been pre-specified, this parametric policy uses censored demand data to construct maximum likelihood estimators (MLE) for the parameters in the demand distribution.

(iv). **Algorithm c1 (Nonparametric): Burnetas-Smith (B-S) Policy.** The B-S policy is a nonparametric policy which was developed by *Burnetas and Smith* (2000).

(v). **Algorithm c2 (Nonparametric): CAVE Policy.** The CAVE policy, developed by *Godfrey and Powell* (2001), is a nonparametric approach by ap-

proximating the underlying objective function using a series of piecewise linear functions.

Comparison with parametric MLE algorithms. The numerical results are presented in Figure 2.1. Our results indicate that DDM performs very well, and is consistent (i.e., it converges to the optimal solution). In contrast, MLE Censored is significantly slower than DDM, and also suffers from inconsistency, i.e., it often fails to converge to the optimal solution. This is due to a spiral-down effect. More specifically, if the initial inventory level is lower than the optimal target level, the censored demand is likely to give an even lower estimate in the next period (because the firm cannot observe the lost-sales quantity). Then the target inventory level will be set lower and lower, resulting in divergent cost. The consistency of MLE Censored hinges on the (almost) perfect initial estimation of target levels, which is often impractical. In fact, in three of the examples in Figure 2.1, the MLE Censored algorithm did not converge; in the only one where it did converge, we actually picked starting target levels close enough to the optimal levels so that it would converge, which of course would not be possible in practice.

Comparison with nonparametric algorithms. The numerical results are presented in Figure 2.2. Our results show that DDM consistently outperforms both the B-S policy and the CAVE policy. We also find out that the B-S policy has an extremely slow convergence rate while the CAVE policy is much faster but still slower than DDM. Figure 2.2 also displays the performance of the Uncensored SAA policy (assuming the uncensored demand information). It is interesting to note that the DDM policy performs very close to the Uncensored SAA policy in all of our examples.

Extreme cases with uneven lost-sales penalty costs. DDM performs consistently very well for extreme cases with some pathological parameters (see Figure 2.3).

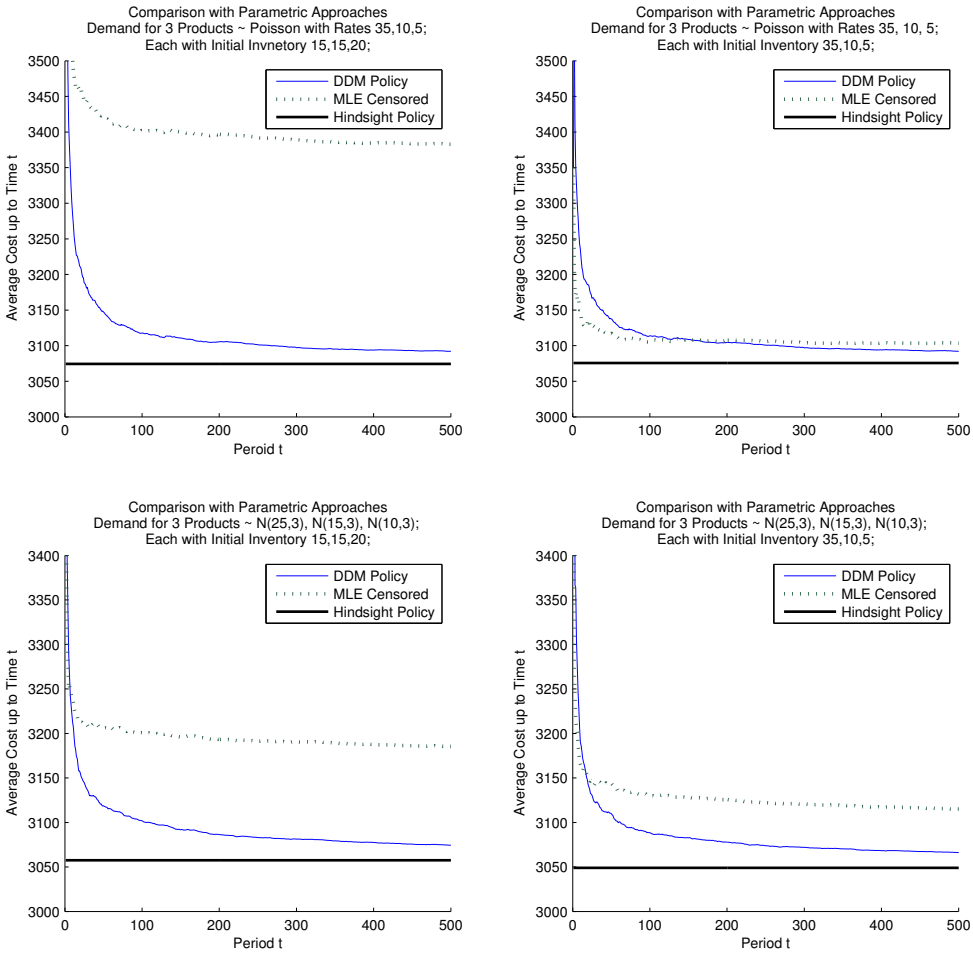

Figure 2.1: Comparison with parametric approaches.

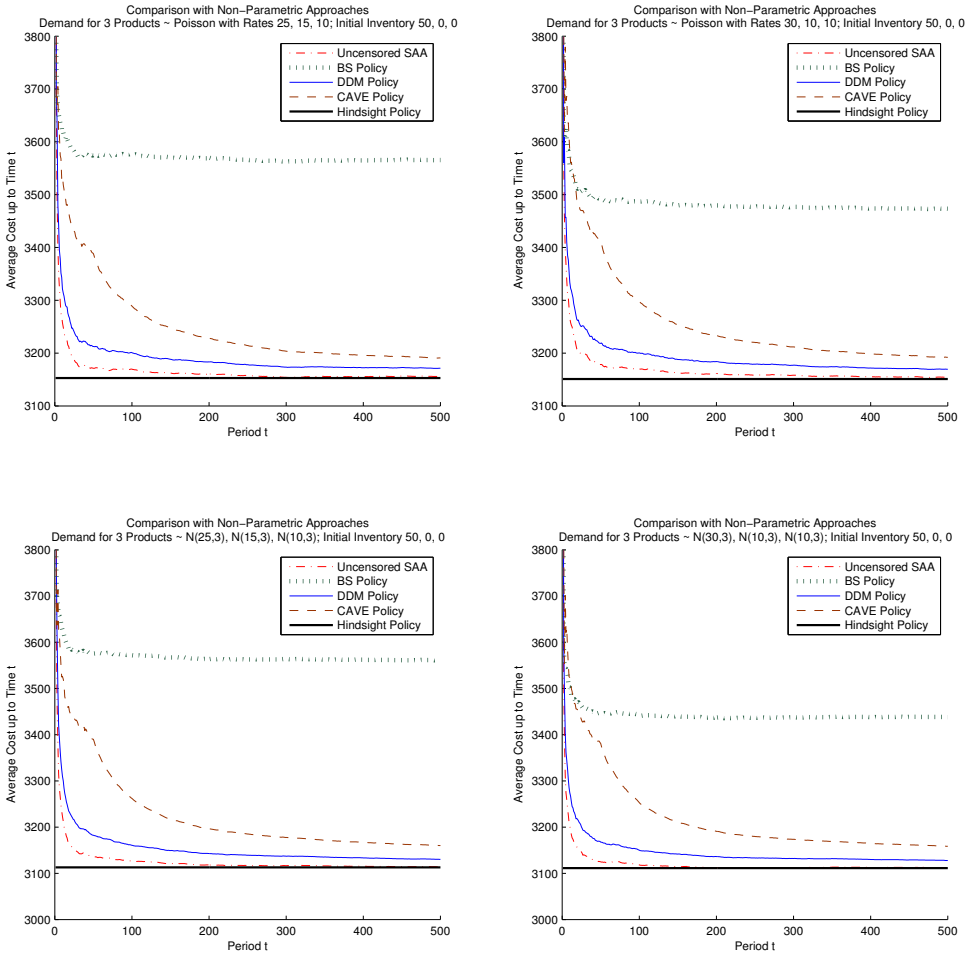

Figure 2.2: Comparison with nonparametric approaches.

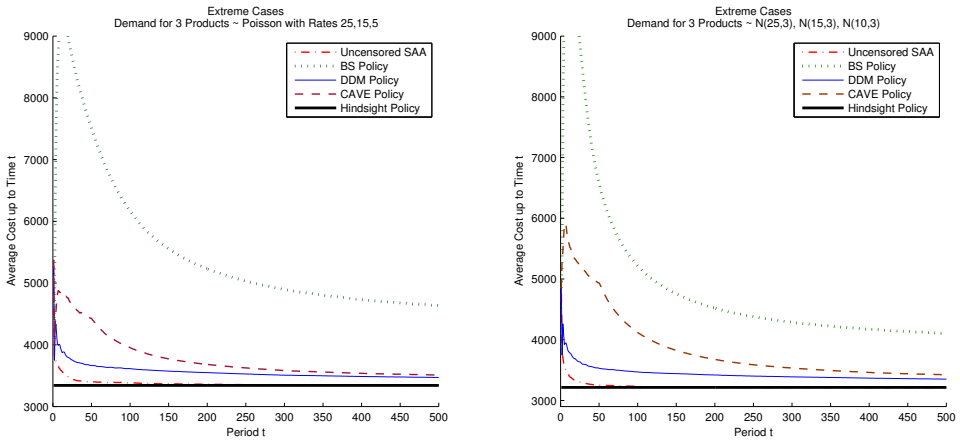

Figure 2.3: Extreme cases with uneven lost-sales penalty costs.

2.7 Concluding Remark

We propose a stochastic gradient descent type algorithm to the stochastic multi-product inventory systems under a warehouse-capacity constraint. We establish the rate of convergence guarantee of our algorithm, i.e., the average expected T-period cost of our policy converges to the optimal cost at the rate of $O(1/\sqrt{T})$. We would like to note that with a slight modification of the newsvendor cost function, we can establish $O(\log T/T)$ convergence of the proposed algorithm.

To close this chapter, we point out three interesting extensions for future research.

(a) *Models with convex ordering cost.* When the ordering cost is convex, *Karlin* (1958) showed the optimal policy to be a *generalized base stock policy* for a single product. That is, there exists a nonnegative function $y(x)$ with $0 \leq dy/dx \leq 1$ such that, in any period, if the starting inventory level is x, order so that the inventory level is brought to $\max(x, y(x))$. The difficulty with extending our results to this convex ordering cost setting is that the clairvoyant optimal solution is no longer a single critical number y^* but a function of x, i.e., $y^*(x)$.

(b) *Models with ordering capacity.* When there is an ordering capacity in a single

product case, *Federgruen and Zipkin* (1986a,b) showed that a modified base-stock policy is optimal under both the average and discounted cost criteria. More specifically, there exists critical value $y^* \geq 0$ such that the manager wants to bring the inventory order-up-to level as close as possible to this y^* at the beginning of each period (i.e., either order up to y^* when possible, or order the full capacity). Despite the simple form of optimal policies, the clairvoyant optimal critical value cannot be *myopically* determined and requires recursive computation via dynamic programming, which adds significant amount of complexity in developing regret bounds.

(c) *Models with setup cost.* When the firm faces fixed costs for ordering that drive lot sizing decisions, computing optimal (s, S) policies (see *Veinott* (1966)) in hind-sight requires a dynamic programming approach and developing a data-driven algorithm for this problem with a provable performance guarantee is likely to be more challenging.

CHAPTER III

Nonparametric Algorithms for Stochastic Inventory Systems with Random Capacity

3.1 Introduction

Capacity plays an important role in a production/inventory system (see *Zipkin* (2000) and *Simchi-Levi et al.* (2014)). The amount of capacity and the variability associated with this capacity affect the production plan as well as the amount of inventory that the firm will carry. As seen from our literature review, there has been a rich and growing literature on capacitated production/inventory systems, and this literature has demonstrated that capacitated systems are inherently more difficult to analyze compared to their uncapacitated counterparts, due to the fact that the capacity constraint makes future costs heavily dependent on the current decision.

To the best of our knowledge, almost all papers on capacitated inventory systems assume that the stochastic future demand that the firm will face and the stochastic future capacity that the firm will have access to are given by exogenous random variables (or random processes), and the inventory decisions are made with full knowledge of future demand and capacity distributions. However, in most practical settings, the firm does not know the demand distribution *a priori*, and has to deduce the demand distribution based on the observed demand while it is producing and selling the prod-

uct. Similarly, when the firm starts producing a new product on a manufacturing line, the firm may have very little idea of the variability associated with this capacity *a priori*. The uncertainty of capacity can be much more significant than the uncertainty in demand in some cases. For instance, Tesla originally stated that it had a line that would be able to build Model 3s at the rate of 5000 per week by the end of June 2017. However, Tesla was never able to reach this production rate at any time in 2017. In fact, during the entire fourth quarter of 2017, Tesla was only able to produce 2425 Model 3s according to *Sparks* (2018). Tesla was finally able to achieve the rate of 5000 produced cars the last week of the second quarter of 2018. However, even at the end of August 2018, Tesla is not able to achieve anywhere near an average 5000 Model 3s production rate per week. Even if we ignore ramp-up issues and assume that Tesla has finally (after one year's delay) achieved "stability", according to Bloomberg's estimate as of September 10, 2018, Tesla was only producing an average of 3857 Model 3s per week in September according to *Randall and Halford* (2018). Even though Tesla may have had more problems than the average manufacturer, significant uncertainty over what production rate can be achieved at a factory is not at all uncommon. In fact, some analysts now question whether this line will ever be able to achieve a consistent production rate of 5000 Model 3s per week displaying the difficulty of estimating the true capacity of a production line.

Another interesting example to look at over time is Apple's launches of its iPhone. When the iPhone 6 was being introduced, there were a large number of articles (see, e.g., *Brownlee* (2014)) indicating that the radical redesign of Apple's smartphone would lead to a short supply of enough devices when it launched due to the increasing difficulty of producing the phone with the new design. In this case, Apple was producing the iPhone already for about 7 years. However, the new generation product was significantly different so that the estimates that Apple had built of its lines' production rates based on the old products were no longer valid. Similarly, as Apple

was about to launch its latest iPhone in October 2018, there were numerous reports about potential capacity problems. *Sohail* (2018) discussed how supply might be constrained at launch due to capacity problems. However, a month and a half after launch, Apple found that sales of its XS and XR models were less than predicted and had to resort to increasing what it offers for trade-in of previous generation iPhone models as an incentive to boost sales. Thus, even in year 11 of production of its product, Apple still has to deal with capacity and demand uncertainty and with each new generation, it has to rediscover its capacity and demand distributions. This is what has motivated us to develop a nonparametric learning algorithm which aids the firm to decide on how many units to produce, while it is learning about its demand and capacity distributions.

3.1.1 Main Result and Contributions

To the best of our knowledge, we develop the first nonparametric learning algorithm, called *the data-driven random capacity algorithm* (DRC for short), for finding the optimal policy in a periodic-review production/inventory system with random capacities where the firm does not have access to both the demand and capacity distributions *a priori*. The performance measure is the standard notion of *regret* in online learning algorithms (see *Shalev-Shwartz* (2012); *Hazan* (2016)), which is defined as the difference between the cost of the proposed learning algorithm and the *clairvoyant optimal* cost, where the clairvoyant optimal cost corresponds to the hypothetical case where the firm knew the demand and capacity distributions *a priori* and applied the optimal policy.

Our main result is to show that the cumulative T-period regret of the DRC algorithm is bounded by $O(\sqrt{T})$, which is also theoretically the best possible for this class of problems. Our work contributes to the active and growing literature in inventory learning literature (as seen from our detailed literature review below). In the

following, we shall highlight the main points of departure of the present chapter from the most closely related prior works.

Our proposed learning algorithm is stochastic gradient descent type, which has been successfully employed for various stochastic inventory systems, started by the seminal work by *Huh and Rusmevichientong* (2009) which studied the classical multi-period stochastic inventory model. *Shi et al.* (2016) then extended their approach to a multi-product setting under a warehouse capacity constraint. The algorithms and their analysis of both *Huh and Rusmevichientong* (2009) and *Shi et al.* (2016) hinged on the myopic optimality of the clairvoyant optimal policy, i.e., it suffices to examine a single-period cost function. However, in the present work, the random production capacity (on how much can be produced) is fundamentally different than the warehouse capacity (on how much can be stored) considered in *Shi et al.* (2016), and it is well-known in the literature that models with random production capacities are much harder to analyze, since the current decisions will impact the cost over an extended period of time (rather than a single period). For example, an under-ordering in one particular period may cause the system to be unable to produce up to the inventory target level over the next multiple periods. Thus, there is a need to carefully re-examine the random capacitated problem with demand and capacity learning.

Apart from the two papers discussed above, there are also three closely related papers to the present chapter, namely, *Zhang et al.* (2018) and *Huh et al.* (2009) and *Zhang et al.* (2019). The former paper developed learning algorithms for the perishable inventory system, and the latter two papers developed learning algorithms for the lost-sales inventory system with positive lead times. It is well-known that both inventory systems need to deal with the lasting impact of current decisions on future cost, due to expanded state vectors and complex system dynamics. On a high level, their main idea is to identify appropriate learning cycles to *de-correlate* the past and

future decisions, and carry out a cyclic stochastic gradient descent procedure based on these learning cycles. There are, however, three points of departure.

First, while the present chapter pursues a similar cyclic updating idea, the cycle in the random capacity problem is very different from the aforementioned inventory systems. It turns out that the "right" cycle identified in our setting is the notion termed *production cycle*, first proposed in *Ciarallo et al.* (1994) used to establish the so-called *extended myopic optimality* for the random capacitated inventory systems. The production cycle is defined as the interval between successive periods in which the policy is able to attain a given base-stock level, in which one can show that the cumulative cost within a production cycle is convex in the base-stock level. Naturally, our DRC algorithm updates base-stock levels in each production cycle. Note that these production cycles (which can be seen as renewal processes) are not *a priori* fixed but are sequentially triggered as demand and supply are realized over time. In our regret analysis, we develop explicit upper bounds on the moments of the length of a random production cycle as well as the stochastic gradient of the cumulative cost within the cycle.

Second, the observed capacity realizations are, in fact, censored. That is, when the plant is able to complete production (i.e., capacity was sufficient in the current period to bring inventory up to the desired level), the actual capacity will not be directly observed. This creates major challenges in the design and analysis of learning algorithms, since *active explorations* are needed to explore the capacity space.

Third, due to random capacity constraints, the firm may not be able to achieve the desired target inventory level as prescribed by the algorithm, and hence we keep track of a virtual (infeasible) bridging system by "temporarily ignoring" the random capacity constraints, which is used to update our target level in the next iteration. The gradient information of this virtual system needs to be correctly obtained from the demand and the censored capacity observed in the real implemented system when

the random capacity constraints are imposed. Also, due to positive inventory carry-over and capacity constraints, we need to ensure that the amount of overage and underage inventory (relative to the desired target level) is appropriately bounded, to achieve the desired rate of convergence of regret.

3.1.2 Relevant Literature

Our work is closely related to two streams of literature: (1) capacitated stochastic inventory systems and (2) nonparametric learning algorithms for stochastic inventory systems.

Capacitated stochastic inventory systems.

There has been a substantial body of literature on capacitated stochastic inventory systems. The dominant paradigm in most of the existing literature has been to formulate stochastic inventory control problems using a dynamic programming framework. This approach is effective in characterizing the structure of optimal policies. We first list paper that considers fixed capacity, *Federgruen and Zipkin* (1986a,b) showed that a modified base-stock policy is optimal under both the average and discounted cost criteria. *Tayur* (1992), *Kapuscinski and Tayur* (1998), and *Aviv and Federgruen* (1997) derived the optimal policy under independent cyclical demands. *Özer and Wei* (2004) showed the optimality of modified base-stock policies in capacitated models with advance demand information. Even for these classical capacitated systems with non-perishable products, the simple structure of their optimal control policies does not lead to efficient algorithms for computing the optimal control parameters. *Tayur* (1992) used the shortfall distribution and the theory of storage processes to study the optimal policy for the case of i.i.d. demands. *Roundy and Muckstadt* (2000) showed how to obtain approximate base-stock levels by approximating the distribution of the shortfall process. *Kapuscinski and Tayur* (1998) proposed a simulation-based technique using infinitesimal perturbation analysis to compute the optimal policy for

capacitated systems with independent cyclical demands. *Özer and Wei* (2004) used dynamic programming to solve capacitated models with advance demand information when the problem size is small. *Levi et al.* (2008) gave a 2-approximation algorithm for this class of problems. *Angelus and Zhu* (2017) identified the structure of optimal policies for capacitated serial inventory systems. All the papers above assume that the firm knows the stochastic demand distribution and the deterministic capacity level.

There has also been a growing body of literature on stochastic inventory systems where both demand and capacity are uncertain. When capacity is uncertain, several papers (e.g., *Henig and Gerchak* (1990); *Federgruen and Yang* (2011); *Huh and Nagarajan* (2010)) assumed that the firm has uncertain yield (i.e., if they start producing a certain number of products, an uncertain proportion of what they started will become finished goods). An alternative approach by *Ciarallo et al.* (1994) and *Duenyas et al.* (1997) assumed that what the firm can produce in a given time interval (e,g., a week) is stochastic (due to for example unexpected downtime, unexpected supply shortage, unexpected absenteeism etc.) and proved the optimality of extended myopic policies for uncertain capacity and stochastic demand under discounted optimal costs scenario. *Güllü* (1998) established a procedure to compute the optimal base stock level for uncertain capacity inventory/production systems. *Wang and Gerchak* (1996) extended the analysis to systems with both random capacity and random yield. *Feng* (2010) addressed a joint pricing and inventory control problem with random capacity and shows that the optimal policy is characterized by two critical values: a reorder point and a target safety stock. More recently, *Chen et al.* (2018b) developed a unified transformation technique which converts a non-convex minimization problem to an equivalent convex minimization problem, and such a transformation can be used to prove the preservation of structural properties for inventory control problems with random capacity. All the papers above assume that the firm knows the stochastic

demand distribution and the stochastic capacity distribution.

Nonparametric learning algorithms for stochastic inventory systems. There has been a recent and growing interest in situations where the distribution of demand is not known *a priori*. Many prior studies have adopted parametric approaches (see, e.g., *Lariviere and Porteus* (1999); *Chen and Plambeck* (2008); *Liyanage and Shanthikumar* (2005); *Chu et al.* (2008)), and we refer interested readers to *Huh and Rusmevichientong* (2009) for a detailed discussion on the differences between parametric and nonparametric approaches.

For nonparametric approaches, *Burnetas and Smith* (2000) considered a repeated newsvendor problem, where they developed an algorithm that converges to the optimal ordering and pricing policy but did not give a convergence rate result. *Huh and Rusmevichientong* (2009) proposed a gradient descent based algorithm for lost-sales systems with censored demand. *Besbes and Muharremoglu* (2013) examined the discrete demand case and showed that active exploration is needed. *Huh et al.* (2011) applied the concept of Kaplan-Meier estimator to devise another data-driven algorithm for censored demand. *Shi et al.* (2016) proposed an algorithm for multi-product systems under a warehouse-capacity constraint. *Zhang et al.* (2018) proposed an algorithm for the perishable inventory system. *Huh et al.* (2009) and *Zhang et al.* (2019) developed learning algorithms for the lost-sales inventory system with positive lead times. *Chen et al.* (2019a, 2015) proposed algorithms for the joint pricing and inventory control problem with backorders and lost-sales, *Chen et al.* (2019b) proposed algorithms for a make-to-stock $M/G/1$ queueing system, respectively. Another popular nonparametric approach in the inventory literature is sample average approximation (SAA) (e.g., *Kleywegt et al.* (2002); *Levi et al.* (2007, 2015)) which uses the empirical distribution formed by *uncensored* samples drawn from the true distribution. Concave adaptive value estimation (e.g., *Godfrey and Powell* (2001); *Powell et al.* (2004)) successively approximates the objective cost function with a

sequence of piecewise linear functions. All the papers surveyed above did not model random capacities in which new learning approaches need to be developed.

3.1.3 Organization and General Notation

The remainder of the chapter is organized as follows. In §3.2, we formally describe the capacitated inventory control problem for random capacity. In §3.3, we show that a target interval policy is optimal for capacitated inventory control problem with salvaging decisions. In §3.4, we introduce the data-driven algorithm for random capacity under unknown demand and capacity distribution. In §3.5, we carry out an asymptotic regret analysis, and show that the average T-period expected cost of our policy differs from the optimal expected cost by at most $O(\sqrt{T})$. In §3.6, we compare our policy performance to the performance of two straw heuristic policies and show that simple heuristic policies used in practice may not work very well. In §3.7, we conclude this chapter and point out plausible future research avenues.

Throughout the chapter, we often distinguish between a random variable and its realizations using capital and lower-case letters, respectively. For any real numbers $a, b \in \mathbb{R}$, $a^+ = \max\{a, 0\}$, $a^- = -\min\{a, 0\}$; the join operator $a \vee b = \max\{a, b\}$; the meet operator $a \wedge b = \min\{a, b\}$.

3.2 Stochastic Inventory Control with Uncertain Capacity

We consider an infinite horizon periodic-review stochastic inventory planning problem with production capacity constraint. We use (time-generic) random variable D to denote random demand, and U to denote random production capacity. The random production capacity may be caused by maintenance or downtime in the production line, lack of materials, etc (see *Zipkin* (2000); *Simchi-Levi et al.* (2014); *Snyder and Shen* (2011)). The demand and the capacity have distribution functions $F_D(\cdot)$ and $F_U(\cdot)$ respectively and density functions $f_D(\cdot)$ and $f_U(\cdot)$ respectively.

At the beginning of our planning horizon, the firm does not know the underlying distributions of D and U. In each period $t = 1, 2, ...$, the sequence of events are as follows:

(a) At the beginning of each period t, the firm observes the starting inventory level x_t before production. (We assume without loss of generality that the system starts empty, i.e., $x_1 = 0$.) The firm also observes the past demand and (censored) capacity realizations up to period $t - 1$.

(b) Then the firm decides the target inventory level s_t. If $s_t \geq x_t$, then it will try to produce $q_t = s_t - x_t$ to bring its inventory level up to s_t. Here, q_t is the target production quantity which may not be achieved due to capacity. During the period, the firm will realize its random production capacity u_t, and therefore its final inventory level will be $s_t \wedge (x_t + u_t)$. We *emphasize* here that the firm will not observe the actual capacity realization u_t if they meet their inventory target s_t. Thus, the firm actually observes the censored capacity $\tilde{u}_t$, i.e., when the production plan cannot be fulfilled at period t, $\tilde{u}_t = u_t$; otherwise, $\tilde{u}_t = (s_t - x_t)^+ \wedge u_t$. On the other hand, if $s_t < x_t$, then the firm will salvage $-q_t = x_t - s_t$ units. Notice that in our model, we allow for negative q_t, which represents salvaging. We denote the inventory level after production or salvaging as $y_t = s_t \wedge (x_t + u_t)$. If the firm decides to bring its inventory level up, it incurs a production cost $c(y_t - x_t)^+$ and if it decides to bring its inventory level down, it receives a salvage value $\theta(x_t - y_t)^+$, where c is the per-unit production cost and θ is the per-unit salvage value. We assume that $\theta \leq c$.

(c) At the end of the period t, after production is completed, the demand D_t is realized, and we denote its realization by d_t, which is satisfied to the maximum extent using on-hand inventory. Unsatisfied demands are *backlogged*, which means that the firm can observe full demand realization d_t in period t. The state transition

can be written as $x_{t+1} = s_t \wedge (x_t + u_t) - d_t = y_t - d_t$. The overage and underage costs at the end of period t is $h(y_t - d_t)^+ + b(d_t - y_t)^+$, where h is the per unit holding cost and b is the per unit backlogging cost.

Following the system dynamics described above, we write the single-period cost as a function of s_t and x_t as follows.

$$\begin{aligned}\Omega(x_t, s_t) &= c(s_t \wedge (x_t + U_t) - x_t)^+ - \theta(x_t - s_t \wedge (x_t + U_t))^+ \\ &\quad + h\left(s_t \wedge (x_t + U_t) - D_t\right)^+ + b\left(D_t - s_t \wedge (x_t + U_t)\right)^+ \\ &= c(y_t - x_t)^+ - \theta(x_t - y_t)^+ + h(y_t - D_t)^+ + b(D_t - y_t)^+.\end{aligned}$$

Let f_t denote the cumulative information collected up to the beginning of period t, which includes all the realized demands d, observed (censored) capacities u, and past ordering decisions s up to period $t-1$. A feasible *closed-loop* control policy π is a sequence of functions $s_t = \pi_t(x_t, f_t), t = 1, 2, ...,$ mapping the beginning inventory x_t and f_t into the ending inventory decision s_t. The objective is to find an efficient and effective adaptive inventory control policy π, or a sequence of inventory targets $\{s_t\}_{t=1}^{\infty}$, which minimizes the long-run average expected cost

$$\limsup_{T\to\infty} \frac{1}{T} \cdot \mathbb{E}\left[\sum_{t=1}^{T} \Omega\left(x_t, s_t\right)\right].$$

3.3 Clairvoyant Optimal Policy

To facilitate the design of a learning algorithm, we first study the clairvoyant scenario by assuming that the distributions of demand and production capacity were given *a priori*. Furthermore, we assume that the actual production capacity in each period is observed by the firm, i.e., there is no capacity censoring in this clairvoyant case. The clairvoyant case is useful as it serves as a lower bound on the cost achievable by the learning model. For the case where the firm can only raise its inventory

(without any salvage decisions), *Ciarallo et al.* (1994) showed that a *produce-up-to* policy is optimal. In this chapter, a *minor contribution* is to extend their policy by enabling the firm to salvage extra goods with salvage price θ at the beginning of each period before the demand is realized. The firm incurs production cost c per-unit good if it decides to produce and receives a salvage value of θ (i.e., incurring a salvage cost $-\theta$) per-unit good if it decides to salvage, and $c \geq \theta$.

We shall introduce a *target interval policy*, and show that it is optimal for both the finite-horizon model and the infinite-horizon model. A target interval policy can be characterized by two threshold values (s_l^*, s_u^*) such that if the starting inventory level $x < s_l^*$, we produce to bring inventory level up to s_l^*, if $x > s_u^*$, we salvage down to s_u^*, and if $s_l^* \leq x \leq x_u^*$, we do nothing.

Assumption 3.1. *We make the following assumptions on the demand and capacity distributions.*

(a) The demands $D_1, \ldots, D_T$ and the capacities $U_1, \ldots, U_T$ are independently and identically distributed (i.i.d.) continuous random variables, respectively. Also, the demand D_t and the capacity U_t are independent across all time periods $t \in \{1, \ldots T\}$.

(b) The (time generic) demand and capacity D and U have a bounded support $[0, \bar{d}]$ and a bounded support $[0, \bar{u}]$, respectively. We also assume that $\mathbb{E}[U] > \mathbb{E}[D]$ to ensure the system stability.

(c) The (clairvoyant) optimal produce-up-to level s_l^* lies in a bounded interval $[0, \bar{s}]$, i.e., $s_l^* \in [0, \bar{s}]$.

Assumption 3.1(a) assumes the stationarity of the underlying production and inventory system to be jointly learned and optimized over time. Assumption 3.1(b) ensures the stability of the system, i.e., the system can clear all the backorders from

time to time. Assumption 3.1(c) assumes that the firm knows an upper bound (potentially a loose one) on the optimal ordering levels. These assumptions are mild and standard in inventory learning literature (see, e.g., *Huh and Rusmevichientong* (2009); *Huh et al.* (2009); *Zhang et al.* (2019, 2018)). We also remark here that an important future research direction is to incorporate non-stationarity of the demand and capacity processes, which would require a significant methodological breakthrough.

Symbol	Type	Description
c	Parameter	Production cost.
θ	Parameter	Salvage cost.
h	Parameter	Per unit holding cost.
b	Parameter	Per unit backlogging cost.
D_t, d_t	Parameter	Random demand and its realization in period t.
F_D, f_D	Parameter	Demand probability and density function.
U_t, u_t	Parameter	Random production capacity and its realization in period t.
F_U, f_U	Parameter	Capacity probability and density function.
s_l^* or s^*	State	Clairvoyant target product-up-to level after ordering.
s_u^*	State	Clairvoyant target salvage-down-to level after salvaging.
x_t	State	Beginning inventory level in period t.
y_t	State	Ending inventory level in period t.
s_t	Control	Target inventory level after ordering/salvaging in period t.
q_t	Control	Ordering/salvaging quantity in period t.

Table 3.1: Summary of Major Notation

3.3.1 Optimal Policy for the Single Period Problem with Salvaging Decisions

We first use a single-period problem to illustrate the idea of target interval policy, and then extend it to the multi-period problem with salvage decisions.

Proposition 3.2. *A target interval policy is optimal for the single period problem.*

Proposition 3.2 shows that the optimal policy for the single period problem is characterized by two critical numbers (s_l^*, s_u^*),. More precisely, the optimal policy can be described as follows:

(i). When $s_l^* \leq x \leq s_u^*$, the firm decides to do nothing.

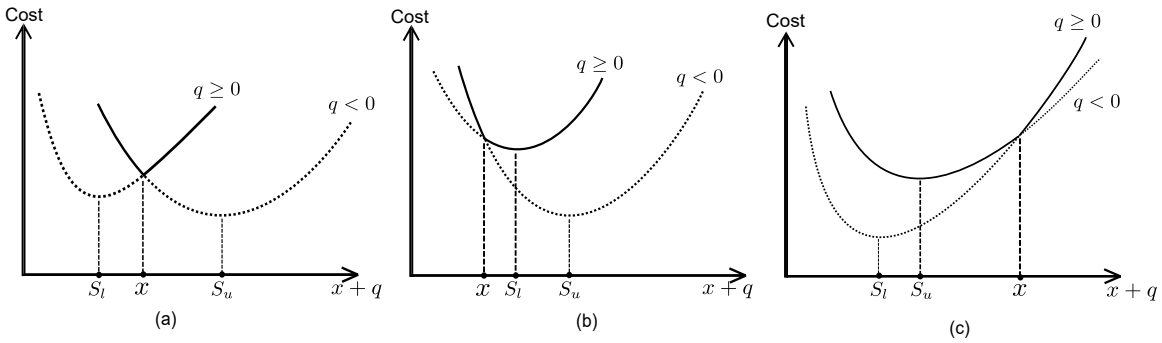

Figure 3.1: Illustration of a target interval policy

(ii). When $x < s_l^*$, the firm decides to produce to bring inventory up to s_l^* as close as possible.

(iii). When $s_u^* < x$, the firm decides to salvage and bring inventory down to s_u^*.

The three situations discussed above can be readily illustrated in Figure 3.1. The two curves are labeled "$q \geq 0$" and "$q < 0$", respectively. The solid curve is the effective cost function $\Omega(y)$, which consists of curve "$q \geq 0$" for $s \geq x$, and curve "$q < 0$" for $s < x$.

Proof of Proposition 3.2:

Proof. To prove the target interval policy, we write the optimal single-period cost function as follows.

$$\mathbb{E}\left[\Omega(x,s)\right] = \min\left\{\min_{s\geq x}\mathbb{E}\left[\Omega_+(x,s)\right], \min_{s<x}\mathbb{E}\left[\Omega_-(x,s)\right]\right\}, \tag{3.1}$$

where

$$\mathbb{E}\left[\Omega_+(x,s)\right] = c\cdot(1-F_U(s-x))(s-x) + c\cdot\int_0^{s-x} r f_U(r)dr$$
$$+ (1-F_U(s-x))\left[\int_s^{\infty} b(z-s)f_D(z)dz + \int_0^{s} h(s-z)f_D(z)dz\right]$$
$$+ \int_0^{s-x}\int_{x+r}^{\infty} b(z-x-r)f_D(z)dz f_U(r)dr + \int_0^{s-x}\int_0^{x+r} h(x+r-z)f_D(z)dz f_U(r)dr, \tag{3.1a}$$

$$\mathbb{E}\left[\Omega_-(x,s)\right] = \theta\cdot(s-x) + \left[\int_s^{\infty} b(z-s)f_D(z)dz + \int_0^{s} h(s-z)f_D(z)dz\right]. \tag{3.1b}$$

Notice that we produce up to s when $s \geq x$, and salvage down to s when $s < x$.

We shall explain that in (3.1a) we condition on the event $s \leq (x+U)$, which has a probability of $(1-F_U(s-x))$, we have $s \wedge (x+U) = s$ and apply the standard newsvendor integral $\mathbb{E}[s-D]^+ + \mathbb{E}[D-s]^+ = \int_0^s (s-z)dz + \int_s^{\infty}(z-s)dz$. Similarly conditioning on the event $s > (x+U)$, which has a probability of $F_U(s-x) = \int_0^{s-x} f_U(r)dr$, we have $s \wedge (x+U) = x+U$ and also apply the standard newsvendor integral. Since we allow for salvaging, the target level s can always be achieved in (3.1b).

To show a *target interval policy* is optimal, we first show that (3.1a) and(3.1b) have global minimizers s_l^* and s_u^*, respectively. Then, we show that $0 \leq s_l^* \leq s_u^* < \infty$. Finally, we discuss different strategies based on different starting inventory levels to imply that a *target interval policy* is optimal.

By applying the Leibniz integral rule, the first partial derivative of (3.1a) with

respect to s is

$$\begin{aligned}
&\frac{\partial}{\partial s}\mathbb{E}\left[\Omega_+(x,s)\right] \\
= \;& (1-F_U(s-x))\left[c+\int_s^{\infty}\frac{\partial}{\partial s}b(z-s)f_D(z)dz+\int_0^{s}\frac{\partial}{\partial s}h(s-z)f_D(z)dz\right].
\end{aligned}$$

It can be easily solved that the solution to the first-order optimality, denoted by s_l^*, is

$$s_l^* = F_D^{-1}\left(\frac{b-c}{h+b}\right)$$

and

$$c+\int_{s_l^*}^{\infty}\frac{\partial}{\partial s}b(z-s)f_D(z)dz+\int_0^{s_l^*}\frac{\partial}{\partial s}h(s-z)f_D(z)dz=0. \tag{3.2}$$

Then it is straightforward to see that $\partial\mathbb{E}\left[\Omega_+(x,s)\right]/\partial s < 0$ for $s < s_l^*$, and we have$\partial\mathbb{E}\left[\Omega_+(x,q)\right]/\partial q > 0$ for $s > s_l^*$. Thus, we conclude that s_l^* is the global minimum of $\mathbb{E}\left[\Omega_+(x,s)\right]$.

Moreover, the second partial derivative of (3.1a) with respect to s is

$$\begin{aligned}
&\frac{\partial^2}{\partial^2 s}\mathbb{E}\left[\Omega_+(x,s)\right] \\
= \;& cf_U(s-x)+(1-F_U(s-x))\left[\int_s^{\infty}\frac{\partial^2}{\partial^2 s}b(z-s)f_D(z)dz\right. \\
&\left.+\int_0^{s}\frac{\partial^2}{\partial^2 s}h(s-z)f_D(z)dz+f_D(s)(h+b)\right] \\
&-f_U(s-x)\left[\int_s^{\infty}\frac{\partial}{\partial s}b(z-s)f_D(z)dz+\int_0^{s}\frac{\partial}{\partial s}h(s-z)f_D(z)dz\right] \\
= \;& (1-F_U(s-x))\left[(h+b)f_D(s)\right]-f_U(s-x)\left[(h+b)F_D(s)-b+c\right].
\end{aligned}$$

It is easy to see when $s \leq s_l^*$,

$$(1 - F_U(s-x))\left[(h+b)f_D(s)\right] > 0 \quad \text{and} \quad f_U(s-x)\left[(h+b)F_D(s) - b + c\right] \leq 0.$$

Therefore, when $s \leq s_l^*$, $\partial^2 \mathbb{E}\left[\Omega_+(x,s)\right]/\partial s^2 \geq 0$, which suggests that $\mathbb{E}\left[\Omega_+(x,s)\right]$ is convex in $s \leq s_l^*$.

Similarly, the first partial derivative of (3.1b) with respect to s is

$$\frac{\partial}{\partial s}\mathbb{E}\left[\Omega_-(x,s)\right] = \theta + \int_s^{\infty} -bf_D(z)dz + \int_0^s hf_D(z)dz \tag{3.3}$$

and it is straightforward to check

$$\frac{\partial^2}{\partial^2 s}\mathbb{E}\left[\Omega_-(x,s)\right] \geq 0,$$

which implies that $\mathbb{E}\left[\Omega_-(x,s)\right]$ is convex in s. Let s_u^* be the solution to the first-order condition $\partial \mathbb{E}\left[\Omega_-(x,s)\right]/\partial s = 0$, and then the solution s_u^* is the global minimum of $\mathbb{E}\left[\Omega_-(x,s)\right]$.

Since $\theta \leq c$, by comparing (3.2) and (3.3), we have $s_l^* \leq s_u^*$. The optimal strategy is as follows.

(i). When $s_l^* \leq x \leq s_u^*$, the firm decides to do nothing.

(ii). When $x < s_l^*$, the firm decides to produce up to s_l^* (as much as possible).

(iii). When $s_u^* < x$, the firm decides to salvage down to s_u^*.

The three cases discussed above can be readily illustrated in Figure 3.1. We sketch (3.1a) and (3.1b) as functions of $s = x + q$. The two curves are labeled "$q \geq 0$" and "$q < 0$", respectively. We note that (3.1a) and (3.1b) intersect at $q = 0$, as discussed earlier. The solid curve is the effective cost function $\Omega(s)$, which consists of the curve "$q \geq 0$" for $s \geq x$, and the curve "$q < 0$" for $s < x$. □

3.3.2 Optimal Policy for the Multi-Period Problem with Salvaging Decisions

Next, we derive the optimal policy for the multi-period problem with salvaging decisions.

Proposition 3.3. *(a) A target interval policy is optimal in any period $t = 1, \ldots, T$ for the multi-period problem with salvaging, where T is the planning horizon.*

(b) A target interval policy is optimal for both the infinite horizon discounted and average cost problems with salvaging decisions.

Proof of Proposition 3.3:

Proof. We first prove Proposition 3.3(a). Define $G_t^*(x_t)$ be the optimal cost from period t to period T with starting inventory x_t, then the optimality equation for the system can be written as follows.

$$G_t^*(x_t) \equiv \min \left\{ \min_{s_t \geq x_t} G_{t+}(x_t, s_t), \min_{s_t < x_t} G_{t-}(x_t, s_t) \right\}, \tag{3.4}$$

where

$$G_{t+}(x_t, s_t) = \mathbb{E}\left[\Omega_+(x_t, s_t)\right] + \int_0^\infty \int_0^{s_t - x_t} G_{t+1}^*(x_t + r - z) f_U(r) dr f_D(z) dz$$

$$+ (1 - F_U(s_t - x_t)) \int_0^\infty G_{t+1}^*(s_t - z) f_D(z) dz, \tag{3.4a}$$

$$G_{t-}(x_t, s_t) = \mathbb{E}\left[\Omega_-(x_t, s_t)\right] + \int_0^\infty G_{t+1}^*(s_t - z) f_D(z) dz, \tag{3.4b}$$

where $\mathbb{E}\left[\Omega_+(x_t, s_t)\right]$ and $\mathbb{E}\left[\Omega_-(x_t, s_t)\right]$ represent the cost functions of period t with the produce-up-to decision and the salvage-down-to decision, respectively, as in Proposition 3.2.

Our goal is to prove that a target interval policy is optimal for any period t, i.e., there exist two threshold levels $s_{t,l}$ and $s_{t,u}$ such that the optimal target level s_t^* satisfies

$$s_t^* = \begin{cases} s_{t,l}, & x_t < s_{t,l}, \\ x_t, & s_{t,l} \leq x_t \leq s_{t,u}, \\ s_{t,u}, & x_t > s_{t,u}. \end{cases}$$

Lemma 3.4. *If $G_{t+1}^*(\cdot)$ is convex, then $G_t^*(\cdot)$ is also convex. Also, a target interval policy is optimal in period t.*

Proof. Proof. We first show that a target interval policy is optimal in period t. The cost function for period t consists of (3.4a) and (3.4b). When $s_t \geq x_t$, the cost function is (3.4a), and when $s_t < x_t$, the cost function is (3.4b). Since $G_{t+1}^*(\cdot)$ and $\mathbb{E}\left[\Omega_-(x_t, s_t)\right]$ are convex in s_t, then we have that (3.4b) is convex in s_t and we let $s_{t,u}$ be the global minimum for (3.4b). For (3.4a), the first-order condition is

$$\begin{aligned} & \frac{\partial}{\partial s_t} G_{t+}(x_t, s_t) \\ = \; & \frac{\partial}{\partial s_t} \mathbb{E}\left[\Omega_+(x_t, s_t)\right] + (1 - F_U(s_t - x_t)) \int_0^\infty G_{t+1}^{*\prime}(s_t - z) f_D(z) dz = 0 \end{aligned} \tag{3.5}$$

Let $s_{t,l}$ be the solution to (3.5). Following the same arguments as in Proposition 3.2 and the convexity of $G_{t+1}^*(\cdot)$ and $\mathbb{E}\left[\Omega_+(x_t, s_t)\right]$ for $s_t \leq s_{t,l}$, we conclude that $s_{t,l}$ is the global minimum for (3.4a). Also, since $\theta \leq c$, we have that $s_{t,l} \leq s_{t,u}$. Thus, a target interval policy is optimal by following the three cases discussed in the single-period problem in Proposition 3.2.

Next, we show that $G_t^*(x_t)$ is convex in x_t. Given $s_{t,l}$ and $s_{t,u}$, we can readily write $G_t^*(x_t)$ with respect to the starting inventory x_t as follows.

$$G_t^*(x_t) = \min\left\{\min_{s_t \geq x_t} G_{t+}(x_t, s_t), \min_{s_t < x_t} G_{t-}(x_t, s_t)\right\} =$$

$$
\begin{cases}
\mathbb{E}\left[\Omega_+(x_t, s_{t,l})\right] & \\
\qquad + \int_0^{\infty} \int_0^{s_{t,l}-x_t} G^*_{t+1}(x_t + r - z) f_U(r) dr f_D(z) dz & \\
\qquad + (1 - F(s_{t,l} - x_t)) \int_0^{\infty} G^*_{t+1}(s_{t,l} - z_t) f_D(z) dz, & x_t < s_{t,l}, \quad (3.6a) \\
\int_{x_t}^{\infty} b(z - x_t) f_D(z) dz + \int_0^{x_t} h(x_t - z) f_D(z) dz & \\
\qquad + \int_0^{\infty} G^*_{t+1}(x_t - z) f_D(z) dz, & s_{t,l} \le x_t \le s_{t,u}, \quad (3.6b) \\
\mathbb{E}\left[\Omega_-(x_t, s_{t,u})\right] + \int_0^{\infty} G^*_{t+1}(s_{t,u} - z) f_D(z) dz, & s_{t,u} < x_t, \quad (3.6c)
\end{cases}
$$

where $s_{t,l}$ and $s_{t,u}$ are the global minima defined earlier.

By the Leibniz integral rule, the second derivatives of (3.6a), (3.6b), and (3.6c) with respect to x_t are

$$
\frac{\partial^2}{\partial^2 x_t} G^*_t(x_t) =
$$

$$
\begin{cases}
\frac{\partial^2}{\partial^2 x_t} \mathbb{E}\left[\Omega_+(x_t, s_{t,l})\right] & \\
\qquad + \int_0^{\infty} \int_0^{s_{t,l}-x_t} G^{*''}_{t+1}(x_t + r - z) f_U(r) dr f_D(z) dz, & x_t < s_{t,l}, \quad (3.7a) \\
(h + b) f_D(x_t) + \int_0^{\infty} G^{*''}_{t+1}(x_t - z) f_D(z) dz, & s_{t,l} \le x_t \le s_{t,u}, \quad (3.7b) \\
\frac{\partial^2}{\partial^2 x_t} \mathbb{E}\left[\Omega_-(x_t, s_{t,u})\right] + \int_0^{\infty} G^{*''}_{t+1}(s_{t,u} - z) f_D(z) dz, & s_{t,u} < x_t. \quad (3.7c)
\end{cases}
$$

Because $\mathbb{E}\left[\Omega_+(x_t, s_{t,l})\right]$ and $\mathbb{E}\left[\Omega_-(x_t, s_{t,u})\right]$ are convex (which has been derived in Proposition 3.2), and $G^{*''}_{t+1}(\cdot)$ is positive (by the inductive assumption), we have that (3.7a), (3.7b), and (3.7c) are all positive. This means that $G^*_t(x_t)$ is convex on these

three intervals separately. It remains to show that $G_t^*(x_t)$ is convex on the entire domain by carefully checking the connecting points between these intervals. We have

$$\begin{aligned}
\lim_{\delta\to 0^-} \frac{G_t^*(s_{t,l}) - G_t^*(s_{t,l}-\delta)}{\delta} &= (h+b)F_D(s_{t,l}) - b + \int_0^\infty G_{t+1}^{*\prime}(s_{t,l}-z)f_D(z)dz, \\
\lim_{\delta\to 0^+} \frac{G_t^*(s_{t,l}+\delta) - G_t^*(s_{t,l})}{\delta} &= (h+b)F_D(s_{t,l}) - b + \int_0^\infty G_{t+1}^{*\prime}(s_{t,l}-z)f_D(z)dz, \\
\lim_{\delta\to 0^-} \frac{G_t^*(s_{t,u}) - G_t^*(s_{t,u}-\delta)}{\delta} &= (h+b)F_D(s_{t,u}) - b + \int_0^\infty G_{t+1}^{*\prime}(s_{t,u}-z)f_D(z)dz, \\
\lim_{\delta\to 0^+} \frac{G_t^*(s_{t,u}+\delta) - G_t^*(s_{t,u})}{\delta} &= (h+b)F_D(s_{t,u}) - b + \int_0^\infty G_{t+1}^{*\prime}(s_{t,u}-z)f_D(z)dz.
\end{aligned}$$

Thus, we can see that the first derivatives at the connecting points are the same, and therefore $G_t^*(\cdot)$ is continuously differentiable and convex on the entire domain. □

By definition, we know that $G_{T+1}^*(x_{T+1}) = -\theta(x_{T+1})$ is convex. Thus, from Lemma 3.4 and applying induction, we conclude that the target interval policy is optimal for any period $t = 1, \ldots, T$. This proves Proposition 3.3(a).

We then prove Proposition 3.3(b). The single-period cost and derivative are exactly the same for both the produce-up-to and salvage-down-to cases. The optimality equation for infinite horizon case can be written as

$$J(x) = \min\left\{\min_{s\geq x} G_+(x,s), \min_{s<x} G_-(x,s)\right\}.$$

where

$$G_+(x,s) = \mathbb{E}\left[\Omega_+(x,s)\right] + \alpha \int_0^\infty \int_0^{s-x} J(x+r-z) f_U(r) dr f_D(z) dz$$

$$+ \alpha(1-F(s-x)) \int_0^\infty J(s-z) f_D(z) dz, \tag{3.8a}$$

$$G_-(x,s) = \mathbb{E}\left[\Omega_-(x,s)\right] + \alpha \int_0^\infty J(s-z) f_D(z) dz, \tag{3.8b}$$

where $0 \leq \alpha < 1$ is the discount factor. Our goal is to prove that a target interval policy is optimal, i.e., there are two threshold levels s_l^* and s_u^* such that the optimal target level is s_l^* when $x < s_l^*$ and s_u^* when $x > s_u^*$ and x otherwise. Similar to Lemma 3.4, we can show that $J(x)$ is convex in the starting inventory x. The remainder argument is identical to that of Proposition 3.3(a). For the infinite horizon average cost problem, it suffices to check the set of conditions in *Schäl* (1993), ensuring the limit of the discounted cost optimal policy is the average optimal policy as the discount factor $\alpha \to 1^-$. Verifying these conditions is standard, and we omit the details for brevity.

We have shown that if the firm has the option to salvage extra goods at the beginning of each period, then it will choose to salvage extra goods if the starting inventory is high enough. In the full-information problem, we can immediately conclude that in the infinite horizon problem, the salvage decision will only be made in the first period when the initial starting inventory is higher than s_u^*. This is because after salvaging down to s_u^* in the first period, the inventory level will gradually be consumed down below s_l^* and after that, the inventory level will never exceed s_l^* again, due to the stationary demand assumption. Thus, the optimal produce-up-to level s_l^* is the same as the optimal produce-up-to level, denoted by s^*, in *Ciarallo et al.* (1994) without salvaging options, and an extended myopic policy described therein is also optimal

for the infinite horizon average cost setting. In the remainder of this chapter, we will use s_l^* and s^* interchangeably.

However, we must *emphasize* here that in the learning version of the problem, since we do not know the demand and capacity distributions (and of course s_l^* or s^*), we need to *actively explore* the inventory space, and salvaging decisions will be made in our nonparametric online learning algorithm (more frequently in the beginning phase).

3.4 Nonparametric Learning Algorithms

As discussed in §3.1, in many practical scenarios, the firm does not know the distribution of demand D nor the distribution of production capacity U at the beginning of the planning horizon. Instead, the firm has to rely on the observable demand and capacity realizations over time to make adaptive production decisions. More precisely, in each period t, the firm can observe the realized demand d_t as well as the observed production capacity $\tilde{u}_t$. In our model, while d_t is the true demand realization (since the demands are backlogged), the observed production capacity $\tilde{u}_t$ is, in fact, *censored*. More explicitly, the censored capacity $\tilde{u}_t = (s_t - x_t)^+ \wedge u_t$. That is, suppose the firm wants to raise the starting inventory level x_t to some target level s_t. If the true realized production capacity $u_t > (s_t - x_t)^+$, then the firm cannot observe the uncensored capacity realization u_t. Our objective is to find an efficient and effective learning production control policy whose long-run average cost converges to the clairvoyant optimal cost (had the distributional information of both the random demand and the random capacity been given *a priori*) at a provably tight convergence rate.

3.4.1 The Notion of Production Cycles

It is well-known in the literature that the optimal policy for a capacitated inventory system cannot be solved myopically, i.e., the control that minimizes a single-period

cost is not optimal. Moreover, when capacities are random, the per-period cost function is non-convex (see, e.g., *Ciarallo et al.* (1994) and *Chen et al.* (2018b)). Thus, the standard online stochastic gradient descent algorithms cannot be readily applied to solve our model.

To overcome this difficulty, we partition the set of time periods into carefully designed learning cycles, and update our production target levels from cycle to cycle, instead of from period to period. We first formally define these learning cycles. Given that we produce up to the target level s_t in some period t and then use the same target level s_t for all subsequent periods, we define a *production cycle* as the set of successive periods starting from period t to the next period in which we are able to produce up to s_t again. Mathematically, let τ_j denote the starting period of the j^{th} production cycle. Then, for any given initial target level $s_1 \in [0, \bar{s}]$, we have

$$\tau_1 = 1, \quad \tau_j = \min\left\{ t \geq \tau_{j-1} + 1 \;\middle|\; x_t + u_t \geq s_{\tau_{j-1}} \right\}, \text{ for all } j \geq 2.$$

For convenience, we call s_{τ_j} the *cycle target level* for production cycle j. We let l_j be the cycle length of the j^{th} production cycle, i.e., $l_j = \tau_{j+1} - \tau_j$.

Figure 3.2 gives a simple graphical example of a production cycle. Suppose the target production level $s_5 = 30$ and the realized capacity levels $u_t = 15$ for $t = 5, \ldots, 9$. In periods $6, 7, 8$, we are not able to attain the target level s_5 even if we produce the full capacity in these periods, whereas we are able to do so in period 9. Therefore this production cycle runs from period 5 to period 9. Note that in period 9, we could only observe the censored capacity $\tilde{u}_9 = 11$ (instead of the true realized capacity $u_9 = 15$), because we only need to produce 11 to attain the target level.

The definition of these production cycles is motivated by the idea of *extended myopic policies*, which we shall discuss next. In the full-information (clairvoyant) case with stationary demand, the structural results in §3.3 imply that if the system

starts with initial inventory s^* (for simplicity we drop the subscript from the optimal produce-up-to level s_l^*), then the optimal policy is a modified base-stock policy, i.e., in each period t,

$$y_t = \begin{cases} s^*, & \text{if } x_t + u_t \geq s^*, \\ x_t + u_t, & \text{if } x_t + u_t < s^*. \end{cases}$$

In this case, our definition of production cycles reduces to

$$\tau_1 = 1, \quad \tau_j = \min \left\{ t \geq \tau_{j-1} + 1 \;\middle|\; y_t = s^* \right\}, \text{ for all } j \geq 2.$$

In other words, the optimal system forms a sequence of production cycles whose cycle target levels are all set to be s^*, which is also illustrated at the top portion of Figure 3.3. *Ciarallo et al.* (1994) showed that the extended myopic policy, which is obtained by merely minimizing the expected total cost within a single production cycle, is optimal. This motivates us to design a nonparametric learning algorithm that updates the modified base-stock levels in a cyclic way, in which the sequence of production cycle costs in our system will eventually converge to the production cycle cost of the optimal system. We *emphasize* again that the (clairvoyant) optimal system needs not salvage since s^* can be computed with known demand and capacity distributions, whereas our system needs to actively explore the inventory space to recover s^* and thus salvaging can happen frequently in the beginning phase of the learning algorithm.

3.4.2 The Data-Driven Random Capacity Algorithm (DRC)

With the definition of production cycles, we shall describe our data-driven random capacity algorithm (DRC for short). The DRC algorithm keeps track of two systems in parallel, and also ensures that both systems share the same production cycles as in the optimal system (which uses the same optimal base-stock level s^* in every period).

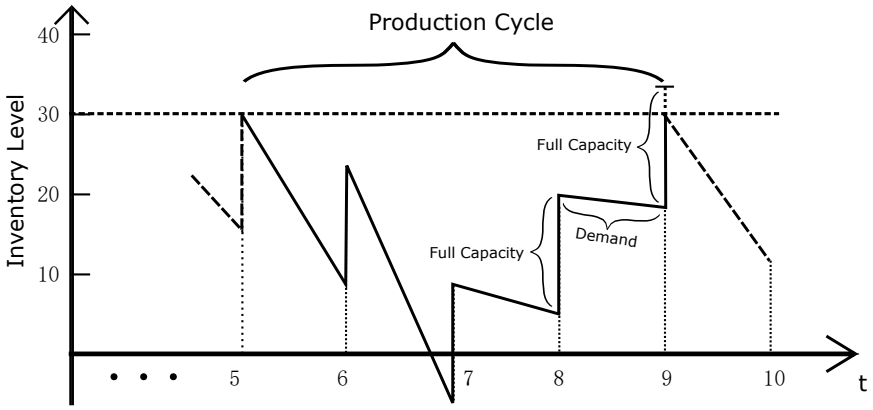

Figure 3.2: An illustration of a production cycle

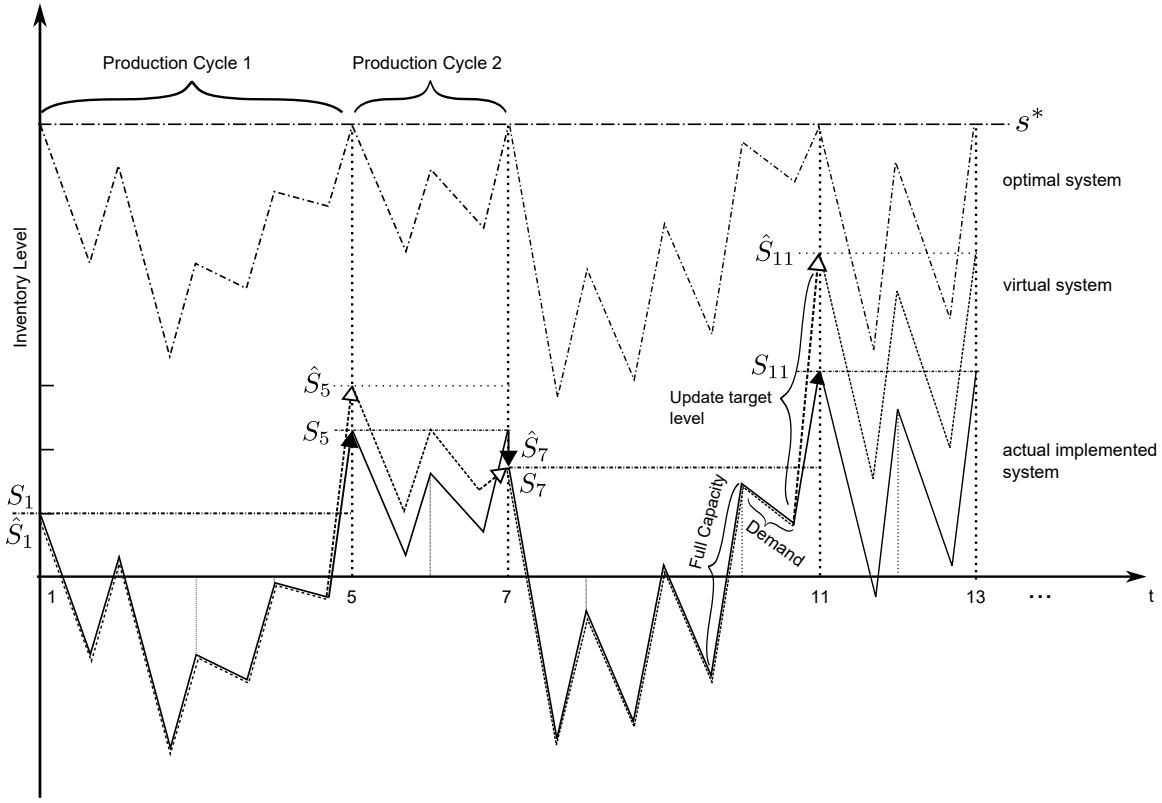

Figure 3.3: An illustration of the algorithmic design

The optimal system is depicted using dash-dot lines shown at the top of Figure 3. The optimal system starts at optimal base-stock level s^*, and uses s^* as target level in every period.

The first system that the DRC algorithm keeps track of is a virtual (or ideal) system, which starts from an arbitrary inventory level $\hat{s}_1$. The DRC algorithm maintains a triplet $(\hat{s}_t, \hat{y}_t, \hat{x}_t)$ in each period t, where $\hat{s}_t$ is the virtual target level, $\hat{y}_t$ is the virtual inventory level, and $\hat{x}_t$ is the virtual starting inventory level. At the beginning of each production cycle j, namely, in period τ_j, the DRC algorithm computes the (desired) virtual cycle target level $\hat{s}_{\tau_j}$, and *artificially adjusts* the virtual inventory level $\hat{y}_{\tau_j} = \hat{s}_{\tau_j}$ by temporarily ignoring the random capacity constraint in that period. For all subsequent periods $t \in [\tau_j + 1, \tau_{j+1} - 1]$ within production cycle j, the DRC algorithm sets the virtual target production level $\hat{s}_t = \hat{s}_{\tau_j}$ and runs the virtual system as usual (facing the same demands and random capacity constraints as in the actual implemented system), i.e., $\hat{y}_t = \hat{s}_t \wedge (\hat{x}_t + u_t)$ and $\hat{x}_{t+1} = \hat{y}_t - d_t$. Figure 3.3 gives an example of the evolution of a virtual system, as depicted using dotted lines.

The second system is the actual implemented system, which starts from an arbitrary inventory level $s_1 = \hat{s}_1$. The DRC algorithm maintains a triplet (s_t, y_t, x_t) in each period t, where s_t is the target production level, y_t is the actual attained inventory level, and x_t is the actual starting inventory level. Different than the virtual system described above, at the beginning of each production cycle j, namely, in period τ_j, the DRC algorithm tries to reach the (desired) virtual target level $\hat{s}_{\tau_j}$ but may fail to do so due to random capacity constraints. The resulting inventory level y_{τ_j} may possibly be lower than $\hat{s}_{\tau_j}$. Nevertheless, to keep the production cycle synchronized with that of the optimal system, we simply set the cycle target level $s_{\tau_j} = y_{\tau_j}$, and keep the target production level the same within the production cycle, i.e., $s_t = s_{\tau_j}$ for all $t \in [\tau_j, \tau_{j+1} - 1]$. Figure 3.3 gives an example of the evolution of an actual implemented system (as depicted using solid lines).

We now present the detailed description of the DRC algorithm.

The Data-Driven Random Capacity Algorithm (DRC)

Step 0. (*Initialization.*) In the first period $t = 1$, set the initial inventory $x_1 \in [0, \bar{s}]$ arbitrarily. We set both the target level and the virtual target level the same as the initial inventory, i.e., $s_1 = \hat{s}_1 = x_1$. Then we also have the actual attained inventory level $y_1 = x_1$ and the virtual inventory level $\hat{y}_1 = \hat{x}_1 = x_1$. Initialize the counter for production cycles $j = 1$, and set $t = \tau_1 = 1$.

Step 1. (*Updating the Virtual System.*)

The algorithm updates the virtual target level in period $t + 1$ by

$$\hat{s}_{t+1} = \begin{cases} \mathbf{P}_{[0,\bar{s}]}\left(\hat{s}_{\tau_j} - \eta_j \cdot \sum_{k=\tau_j}^{t} \mathcal{G}_k(\hat{s}_{\tau_j})\right), & \text{if } t = \tau_j, \\ \hat{s}_{\tau_j}, & \text{if } t > \tau_j, \end{cases}$$

$$\text{where} \qquad \mathcal{G}_k(\hat{s}_{\tau_j}) = \begin{cases} h, & \text{if } \hat{s}_{\tau_j} \wedge (\hat{x}_k + u_k) \geq d_k, \\ -b, & \text{otherwise.} \end{cases}$$

Note that the projection operator $\mathbf{P}_{[0,\bar{s}]}(x) = \max\{0, \min\{x, \bar{s}\}\}$. The step-size is chosen to be

$$\eta_j = \frac{\gamma}{\sqrt{\sum_{k=1}^{j} l_k}}, \quad \text{where} \quad l_k = \tau_{k+1} - \tau_k,$$

where $\gamma > 0$ is a constant (to be optimized later for the tightest theoretical regret bound).

The evolution of the virtual system is given as follows,

$$\hat{y}_t = \begin{cases} \hat{s}_{\tau_j} - \sum_{i=\tau_j}^{t-1} d_i + \sum_{i=\tau_j+1}^{t} u_i, & \text{for } t > \tau_j, \\ \hat{s}_{\tau_j}, & \text{for } t = \tau_j, \end{cases} \qquad \text{and} \qquad \hat{x}_{t+1} = \hat{y}_t - d_t.$$

Step 2. (*Updating the Actual Implemented System.*)

We have the following cases when updating the actual implemented system based on $\hat{s}_t$.

(i). If $\hat{s}_{t+1} \geq s_{\tau_j}$, then we try to produce up to $\hat{s}_{t+1}$, and the actual inventory level y_{t+1} will be

$$y_{t+1} = \begin{cases} \hat{s}_{t+1}, & \text{if } x_{t+1} + u_{t+1} \geq \hat{s}_{t+1}, \\ x_{t+1} + u_{t+1}, & \text{if } x_{t+1} + u_{t+1} < \hat{s}_{t+1}. \end{cases}$$

(a) If $s_{\tau_j} \leq y_{t+1} \leq \hat{s}_{t+1}$, we start a new production cycle $j+1$, by setting the starting period of this new cycle $\tau_{j+1} = t+1$. Correspondingly, we set the virtual cycle target level $\hat{s}_{\tau_{j+1}} = \hat{s}_{t+1}$, and the actual implemented cycle target level $s_{\tau_{j+1}} = y_{t+1}$. We then increase the value of j by one.

(b) If $y_{t+1} < s_{\tau_j}$, we are still in the same production cycle j, and thus we set $s_{t+1} = s_{\tau_j}$.

(ii). If $\hat{s}_{t+1} < s_{\tau_j}$, then we first try to produce up to s_{τ_j} (instead of $\hat{s}_{t+1}$) , and the actual inventory level y_{t+1} will be

$$y_{t+1} = \begin{cases} s_{\tau_j}, & \text{if } x_{t+1} + u_{t+1} \geq s_{\tau_j}, \\ x_{t+1} + u_{t+1}, & \text{if } x_{t+1} + u_{t+1} < s_{\tau_j}. \end{cases}$$

(a) If $y_{t+1} = s_{\tau_j}$, we salvage our inventory level down to $y_{t+1} = \hat{s}_{t+1}$. We then start a new production cycle $j+1$, by setting the starting period of this new cycle $\tau_{j+1} = t+1$. Correspondingly, we set the virtual cycle target level $\hat{s}_{\tau_{j+1}} = \hat{s}_{t+1}$, and the actual implemented cycle target level $s_{\tau_{j+1}} = \hat{s}_{t+1}$. We then increase the value of j by one.

(b) If $y_{t+1} < s_{\tau_j}$, we are still in the same production cycle j, and thus we set $s_{t+1} = s_{\tau_j}$.

We then increase the value of t by one, and go to Step 1. If $t = T$, terminate the algorithm.

3.4.3 Overview of the DRC Algorithm

In Step 1, we update the virtual system using the online stochastic gradient descent method. In each period t of any given cycle j, the DRC algorithm tries to minimize the total expected cost associated with production cycle j by updating the virtual target level using a gradient estimator $\sum_{k=\tau_j}^{t} \mathcal{G}_k(\hat{s}_{\tau_j})$ of the total cost accrued from period τ_j to period t. We shall show in Lemma 3.14 below that $G_j(\hat{s}_{\tau_j}) = \sum_{k=\tau_j}^{\tau_{j+1}-1} \mathcal{G}_k(\hat{s}_{\tau_j})$ is the sample-path cycle cost gradient of production cycle j. Note that $G_j(\hat{s}_{\tau_j})$ is the sample-path cycle cost gradient for the *virtual system*. However, we could only observe the demand and censored capacity information in the *actual implemented system*, and the key question is that whether this information is sufficient to evaluate this $G_j(\hat{s}_{\tau_j})$ correctly.

Lemma 3.5. *The sample-path cycle cost gradient of the virtual system* $G_j(\hat{s}_{\tau_j}) = \sum_{k=\tau_j}^{\tau_{j+1}-1} \mathcal{G}_k(\hat{s}_{\tau_j})$ *for every cycle* $j \geq 1$ *can be evaluated correctly, using the observed demand and censored capacity information of the actual implemented system.*

Proof. It suffices to show that for each period $k = \tau_j, \ldots, \tau_{j+1} - 1$, the cost gradient estimator $\mathcal{G}_k(\hat{s}_{\tau_j})$ can be evaluated correctly. We have the following two cases.

(a) If $k = \tau_j$, i.e., the production cycle j starts in period k, we must have $x_k + \tilde{u}_k \geq s_{\tau_j - 1}$ by our definition of production cycle. In addition, we observe the full capacity $\tilde{u}_i = u_i$ in period $i = \tau_{j-1} + 1, \ldots, k - 1$ but only observe the censored capacity $\tilde{u}_k \leq u_k$ in period k.

(1) if $s_k = \hat{s}_k$, by the system dynamics we have

$$\hat{s}_k = s_k = x_k + \tilde{u}_k \leq \hat{x}_k + \tilde{u}_k \leq \hat{x}_k + u_k,$$

where the first inequality holds because by our algorithm design, we always have $s_{\tau_{j-1}} \leq \hat{s}_{\tau_{j-1}}$ for all $j = 2, 3, \ldots$, and then

$$x_k = s_{\tau_{j-1}} - \sum_{i=\tau_{j-1}}^{\tau_j - 1} d_i + \sum_{i=\tau_{j-1}+1}^{\tau_j - 1} u_i \leq \hat{s}_{\tau_{j-1}} - \sum_{i=\tau_{j-1}}^{\tau_j - 1} d_i + \sum_{i=\tau_{j-1}+1}^{\tau_j - 1} u_i = \hat{x}_k.$$

Hence, the event $\left\{\hat{s}_{\tau_j} \wedge (\hat{x}_k + u_k) \geq d_k\right\}$ is equivalent to $\left\{\hat{s}_{\tau_j} \geq d_k\right\}$, and therefore we can evaluate $\mathcal{G}_k(\hat{s}_{\tau_j})$ correctly.

(2) if $s_k < \hat{s}_k$, we have produced full capacity and therefore observe the full capacity $\tilde{u}_k = u_k$. Then the event $\left\{\hat{s}_{\tau_j} \wedge (\hat{x}_k + u_k) \geq d_k\right\}$ is equivalent to $\left\{\hat{s}_{\tau_j} \wedge (\hat{x}_k + \tilde{u}_k) \geq d_k\right\}$, and therefore we can evaluate $\mathcal{G}_k(\hat{s}_{\tau_j})$ correctly.

(b) On the other hand, if $k \in [\tau_j + 1, \tau_{j+1} - 1]$, i.e., then we are still in the current production cycle j. In this case, we always produce at full capacity, and therefore we observe the full capacity $\tilde{u}_k = u_k$. Then the event $\left\{\hat{s}_{\tau_j} \wedge (\hat{x}_k + u_k) \geq d_k\right\}$ is equivalent to $\left\{\hat{s}_{\tau_j} \wedge (\hat{x}_k + \tilde{u}_k) \geq d_k\right\}$, and therefore we can evaluate $\mathcal{G}_k(\hat{s}_{\tau_j})$ correctly.

Combining the above two cases yields the desired the result. □

In Step 2, we compare $\hat{s}_{t+1}$ and s_{τ_j} to decide how to update the actual implemented system. We have two cases. The first case is when $\hat{s}_{t+1} \geq s_{\tau_j}$. We want to produce up to the new target level $\hat{s}_{t+1}$ instead of s_{τ_j}. If the actual implemented inventory level $y_{t+1} \geq s_{\tau_j}$, we know that the current production cycle ends because we have achieved at least s_{τ_j}, and then we shall start the next production cycle. In order to perfectly align the production cycle with that of the optimal system when $\hat{s}_{t+1} \geq y_{t+1} \geq s_{\tau_j}$, we should set the next cycle target level $s_{\tau_{j+1}} = y_{t+1}$. Otherwise, we produce at full capacity, and stay in the same production cycle, which is also synchronized with the optimal production cycle. The second case is when $\hat{s}_{t+1} < s_{\tau_j}$. We first produce up to the current cycle target level s_{τ_j} to check whether we can start the next production

cycle. If s_{τ_j} is achieved, we shall start the next production cycle and salvage the inventory level down to $y_{t+1} = \hat{s}_{t+1}$ and also set the new cycle target level $s_{\tau_{j+1}} = \hat{s}_{t+1}$. Otherwise, we produce at full capacity, and stay in the same production cycle, which is also synchronized with the optimal production cycle.

The central idea here is to *exactly align* the production cycles of the actual implemented system (as well as the virtual bridging system) with those of the (clairvoyant) optimal system, even while updating our cycle target level at the beginning of each production cycle. As illustrated in Figure 3.3, the optimal system knows s^* *a priori* and keeps using the target level s^* (i.e., the optimal modified base-stock level) in every period t. Whenever the target level s^* is achieved, we start the next production cycle. However, in the learning problem, the firm does not know s^* and needs to constantly update the cycle target level at the beginning of each production cycle. Due to the discrepancy between the new and the previous target levels, it is crucial to design an algorithm that can determine whether the current production cycle ends, and whether we should adopt the new target level in the very same period. Figure 3.4 shows the possible scenarios. The scenarios $1(a)$, $1(b)$ and $1(c)$ show the case when $\hat{s}_{t+1} \geq s_{\tau_j}$. In this case, we always raise the inventory to $\hat{s}_{t+1}$ as much as possible. If $\hat{s}_{t+1}$ is achieved, we know that the production cycle ends. Even if $\hat{s}_{t+1}$ is not achieved, we know that we produce at full capacity and then can readily determine whether the production cycle ends (by checking if we reach at least s_{τ_j}). The scenarios $2(a)$, $2(b)$ and $2(c)$ show the case when $\hat{s}_{t+1} < s_{\tau_j}$. In this case, we always raise the inventory to s_{τ_j} as much as possible to determine whether the production cycle ends (by checking if we reach exactly s_{τ_j}). We salvage the inventory level down to $\hat{s}_{t+1}$ only if the production cycle ends.

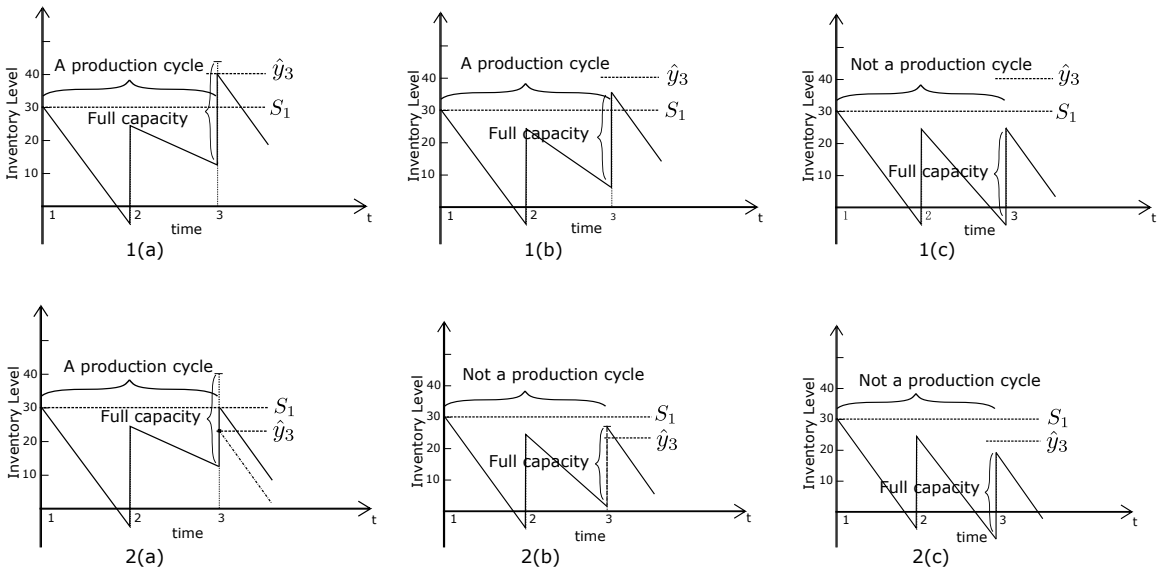

Figure 3.4: A schematic illustration of all possible scenarios

3.5 Performance Analysis of the DRC Algorithm

We carry out a performance analysis of our proposed DRC algorithm. The performance measure is the natural notion of regret, which is defined as the difference between the cost incurred by our nonparametric learning algorithm DRC and the clairvoyant optimal cost (where the demand and production capacity distribution are both known *a priori*). That is, for any $T \geq 1$,

$$\mathcal{R}_T = \mathbb{E}\left[\sum_{t=1}^{T} \Omega(x_t, s_t) - \Omega(x_t, s^*)\right],$$

where s_t is the target level prescribed by the DRC algorithm for period t, and s^* is the clairvoyant optimal target level. Theorem 3.6 below states the main result of this chapter.

Theorem 3.6. *For stochastic inventory systems with demand and capacity learning, the cumulative regret R_T of the data-driven random capacity algorithm (DRC) is bounded by $O(\sqrt{T})$. In other words, the average regret R_T/T approaches to 0 at the*

rate of $O(1/\sqrt{T})$.

Remark 3.7. We first define $\mu = \mathbb{E}[U] - \mathbb{E}[D]$, the difference between expected capacity and expected demand. We define $\upsilon = 2\mu^2/(\bar{u}+\bar{d})^2$ and $X_1 = (h \vee b) l_1 - \sum_{t=\tau_1+1}^{\tau_2} U_t + \sum_{t=\tau_1}^{\tau_2-1} D_t$, and then further define $\alpha = -\mathbb{E}[X_1]$ and $\sigma^2 = Var[X_1]$ and $\beta = \mathbb{E}[X_1^3]$.

The optimal constant γ in the step size is given by

$$\gamma = \frac{\bar{s}}{\sqrt{(h \vee b)^2 \left(\frac{1}{\upsilon} + \frac{2}{\upsilon^2} + \frac{2}{\upsilon^3}\right) + 2(h \vee b)^2 \frac{\bar{s}}{\mu} \frac{\sigma}{\alpha} e^{\frac{6\beta}{\sigma^3} + \frac{\alpha}{\sigma}} + 2(c+\theta)(h \vee b) \frac{\sigma}{\alpha} e^{\frac{6\beta}{\sigma^3} + \frac{\alpha}{\sigma}}}},$$

and the associated constant K in the regret bound of Theorem 3.6 is given by

$$K = \bar{s} \sqrt{(h \vee b)^2 \left(\frac{1}{\upsilon} + \frac{2}{\upsilon^2} + \frac{2}{\upsilon^3}\right) + 2(h \vee b)^2 \frac{\bar{s}}{\mu} \frac{\sigma}{\alpha} e^{\frac{6\beta}{\sigma^3} + \frac{\alpha}{\sigma}} + 2(c+\theta)(h \vee b) \frac{\sigma}{\alpha} e^{\frac{6\beta}{\sigma^3} + \frac{\alpha}{\sigma}}}.$$

The proposed DRC algorithm is the first nonparametric learning algorithm for random capacitated inventory systems, which achieves a square-root regret rate. Moreover, this square-root regret rate is *unimprovable*, even for the repeated newsvendor problem without inventory carryover and with infinite capacity, which is a special case of our problem.

Proposition 3.8. *Even in the case of uncensored demand, there exist problem instances such that the expected regret for any learning algorithm is lower bounded by* $\Omega(\sqrt{T})$.

Proof. The proof of Proposition 3.8 is identical to that of Proposition 1 in *Zhang et al.* (2018) for the repeated uncapacitated newsvendor problem. □

The remainder of this chapter is to establish the regret upper bound in Theorem 3.6. For each $j \geq 1$, if we adopt the cycle target level s_{τ_j} and also artificially set the initial inventory level $x_{\tau_j} = s_{\tau_j}$, we can then express the cost associated with the

production cycle j as

$$\begin{aligned}
\Theta(s_{\tau_j}) &= \sum_{t=\tau_j+1}^{\tau_{j+1}} c\left(s_{\tau_j} \wedge (x_t + U_t) - x_t\right)^+ \qquad (3.9)\\
&\quad + \sum_{t=\tau_j}^{\tau_{j+1}-1} \left[h\left(s_{\tau_j} \wedge (x_t + U_t) - D_t\right)^+ + b\left(D_t - s_{\tau_j} \wedge (x_t + U_t)\right)^+\right] \\
&= \sum_{t=\tau_j+1}^{\tau_{j+1}-1} cU_t + c(s_{\tau_j} - x_{\tau_{j+1}}) \\
&\quad + \sum_{t=\tau_j}^{\tau_{j+1}-1} \left[h\left(s_{\tau_j} \wedge (x_t + U_t) - D_t\right)^+ + b\left(D_t - s_{\tau_j} \wedge (x_t + U_t)\right)^+\right] \\
&= \sum_{t=\tau_j+1}^{\tau_{j+1}-1} cU_t + c\left(\sum_{t=\tau_j}^{\tau_{j+1}-1} D_t - \sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t\right) \\
&\quad + \sum_{t=\tau_j}^{\tau_{j+1}-1} \left[h\left(s_{\tau_j} \wedge (x_t + U_t) - D_t\right)^+ + b\left(D_t - s_{\tau_j} \wedge (x_t + U_t)\right)^+\right],
\end{aligned}$$

where the second equality comes from the fact that we always produce at full capacity within a production cycle, except for the last period in which we are able to reach the target level. The third equality follows from expressing

$$x_{\tau_{j+1}} = x_{\tau_j} + \sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t - \sum_{t=\tau_j}^{\tau_{j+1}-1} D_t = s_{\tau_j} + \sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t - \sum_{t=\tau_j}^{\tau_{j+1}-1} D_t.$$

Now, we use J to denote the total number of production cycles before period T, including possibly the last incomplete cycle. (If the last cycle is not completed at T, then we truncate the cycle and also let $\tau_{J+1} - 1 = T$, i.e., $s_{\tau_{J+1}} = s_{\tau_J}$). By the

construction of the DRC algorithm, we can write the cumulative regret as

$$\begin{aligned}
\mathcal{R}_T &= \mathbb{E}\left[\sum_{t=1}^{T}\Omega(x_t,s_t)-\Omega(x_t,s^*)\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{J}\Theta(s_{\tau_j})+\sum_{j=1}^{J}\left(c\left(s_{\tau_{j+1}}-s_{\tau_j}\right)^+ +\theta\left(s_{\tau_j}-s_{\tau_{j+1}}\right)^+\right)\right. \\
&\qquad \left.-\sum_{j=1}^{J}\sum_{t=\tau_j}^{\tau_{j+1}-1}\Omega(x_t,s^*)\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{J}\Theta(s_{\tau_j})-\sum_{j=1}^{J}\sum_{t=\tau_j}^{\tau_{j+1}-1}\Omega(x_t,s^*)\right] \\
&\qquad +\mathbb{E}\left[\sum_{j=1}^{J}\left(c\left(s_{\tau_{j+1}}-s_{\tau_j}\right)^+ +\theta\left(s_{\tau_j}-s_{\tau_{j+1}}\right)^+\right)\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{J}\Theta(\hat{s}_{\tau_j})-\sum_{j=1}^{J}\sum_{t=\tau_j}^{\tau_{j+1}-1}\Omega(x_t,s^*)\right]+\mathbb{E}\left[\sum_{j=1}^{J}\Theta(s_{\tau_j})-\sum_{j=1}^{J}\Theta(\hat{s}_{\tau_j})\right] \\
&\qquad +\mathbb{E}\left[\sum_{j=1}^{J}\left(c\left(s_{\tau_{j+1}}-s_{\tau_j}\right)^+ +\theta\left(s_{\tau_j}-s_{\tau_{j+1}}\right)^+\right)\right],
\end{aligned}$$

where on the right-hand side of the fourth equality, the first term is the production cycle cost difference between using the virtual target level $\hat{s}_{\tau_j}$ and using the clairvoyant optimal target level s^*. The second term is the production cycle cost difference between using the actual implemented target level s_{τ_j} and using the virtual target level $\hat{s}_{\tau_j}$. The third term is the cumulative production and salvaging costs incurred by adjusting the production cycle target levels.

To prove Theorem 3.6, it is clear that it suffices to establish the following set of results.

Proposition 3.9. *For any $J \geq 1$, there exists a constant $K_1 \in \mathbb{R}^+$ such that*

$$\mathbb{E}\left[\sum_{j=1}^{J}\Theta(\hat{s}_{\tau_j})-\sum_{j=1}^{J}\sum_{t=\tau_j}^{\tau_{j+1}-1}\Omega(x_t,s^*)\right] \leq K_1\sqrt{T}.$$

Proposition 3.10. *For any $J \geq 1$, there exists a constant $K_2 \in \mathbb{R}^+$ such that*

$$\mathbb{E}\left[\sum_{j=1}^{J} \Theta(s_{\tau_j}) - \sum_{j=1}^{J} \Theta(\hat{s}_{\tau_j})\right] \leq K_2\sqrt{T}.$$

Proposition 3.11. *For any $J \geq 1$, there exists a constant $K_3 \in \mathbb{R}^+$ such that*

$$\mathbb{E}\left[\sum_{j=1}^{J} \left(c\left(s_{\tau_{j+1}} - s_{\tau_j}\right)^+ + \theta\left(s_{\tau_j} - s_{\tau_{j+1}}\right)^+\right)\right] \leq K_3\sqrt{T}.$$

3.5.1 Several Key Building Blocks for the Proof of Theorem 3.6

Before proving Propositions 3.9, 3.10, and 3.11, we first establish some key preliminary results.

Recall that the production cycle defined in §3.4.1 is the interval between successive periods in which the policy is able to attain a given base-stock level. We first show that the cumulative cost within a production cycle is convex in the base-stock level.

Lemma 3.12. *The production cycle cost $\Theta(s)$ is convex in s along every sample path.*

Proof. It suffices to analyze the first production cycle cost (with $x_1 = s_1$)

$$\begin{aligned}\Theta(s_1) \;=\; & \sum_{t=2}^{\tau_2-1} cU_t + c\left(\sum_{t=1}^{\tau_2-1} D_t - \sum_{t=2}^{\tau_2-1} U_t\right) \\ & + \sum_{t=1}^{\tau_2-1}\left[h\left(s_1 \wedge (x_t + U_t) - D_t\right)^+ + b\left(D_t - s_1 \wedge (x_t + U_t)\right)^+\right].\end{aligned}$$

Taking the first derivative of $\Theta(s_1)$ w.r.t. s_1, we have

$$\Theta'(s_1) = \sum_{t=1}^{\tau_2-1}\left(h(\xi_t^+(s_1)) - b(\xi_t^-(s_1))\right), \tag{3.10}$$

$$\text{where} \qquad \xi_t^+(s_1) = \mathbb{1}\left\{s_1 - \sum_{t'=1}^{t} D_{t'} + \sum_{t'=2}^{t} U_{t'} \geq 0\right\}$$

$$\text{and} \qquad \xi_t^-(s_1) = \mathbb{1}\left\{s_1 - \sum_{t'=1}^{t} D_{t'} + \sum_{t'=2}^{t} U_{t'} < 0\right\}$$

are indicator functions of the positive inventory left-over and the unsatisfied demand at the end of period t, respectively.

For any given $\delta > 0$, we have

$$\Theta'(s_1+\delta) = \sum_{t=1}^{\tau_2-1} \left[h\left(\xi^+(s_1+\delta)\right) - b\left(\xi^-(s_1+\delta)\right)\right].$$

It is clear that when the target level increases, the positive inventory left-over will also increase, i.e, $\xi^+(s_1+\delta) \geq \xi^+(s_1)$. Similarly, we also have $\xi^-(s_1+\delta) \leq \xi^-(s_1)$. Therefore, we have $\Theta'(s_1+\delta) \geq \Theta'(s_1)$ for any value of s_1, and thus $\Theta(\cdot)$ is convex. □

Given the convexity result, our DRC algorithm updates base-stock levels in each production cycle. Note that these production cycles (as renewal processes) are not *a priori* fixed but are sequentially triggered as demand and capacity realize over time. Therefore, we need to develop an upper bound on the moments of a random production cycle. The proof of Lemma 3.13 relies on building an upward drifting random walk with U_t as upward step and D_t as downward step, wherein the chance of hitting a level below zero is exponentially small due to concentration inequalities. Since the ending of a production cycle corresponds to the situation where the random walk hits zero, the second moment of its length of the current production cycle can be bounded.

Lemma 3.13. *The second moment of the length of a production cycle* $\mathbb{E}\left[l_j^2\right]$ *is bounded for all cycle* j*.*

Proof. By the definition of a production cycle in §3.4.1, we have

$$\begin{aligned} & \mathbb{P}\{l_j = l\} \\ = \; & \mathbb{P}\left\{ U_{\tau_j+1} - D_{\tau_j} < 0, \ldots, \sum_{t=\tau_j+1}^{\tau_j+l-1} U_t - \sum_{t=\tau_j}^{\tau_j+l-2} D_t < 0, \sum_{t=\tau_j+1}^{\tau_j+l} U_t - \sum_{t=\tau_j}^{\tau_j+l-1} D_t \geq 0 \right\}. \end{aligned}$$

Since D_t and U_t are both i.i.d., so is l_j. Let M_k be an upward drifting random walk, more precisely, $M_k = \sum_{t=1}^{k} (U_t - D_t)$. Then we have, by letting $\mu = \mathbb{E}[U_t - D_t]$ and $\upsilon = 2\mu^2 / \left(\bar{u} + \bar{d}\right)^2$,

$$\begin{aligned}
\mathbb{E}\left[l_j^2\right] &= \sum_{k=1}^{\infty} k^2 \mathbb{P}\left(M_1 < 0, \ldots, M_{k-1} < 0, M_k \geq 0\right) \\
&\leq \sum_{k=1}^{\infty} k^2 \mathbb{P}\left(M_{k-1} - (k-1)\mu < -(k-1)\mu\right) \\
&\leq \sum_{k=1}^{\infty} k^2 \exp\left(-\frac{2(k-1)\mu^2}{\left(\bar{u} + \bar{d}\right)^2}\right) \\
&\leq \int_0^{\infty} (k+1)^2 \exp\left(-\frac{2k\mu^2}{\left(\bar{u} + \bar{d}\right)^2}\right) dk = \frac{1}{\upsilon} + \frac{2}{\upsilon^2} + \frac{2}{\upsilon^3} < \infty
\end{aligned}$$

where the second inequality follows from the Hoeffding's inequality. □

We also need to develop an upper bound on the cycle cost gradient.

Lemma 3.14. *For any $j \geq 1$, the function $G_j(s) = \sum_{t=\tau_j}^{\tau_{j+1}-1} \mathcal{G}_t(s)$ is the sample-path cycle cost gradient of production cycle j, where s is the cycle target level. Moreover, $G_j(\cdot)$ has a bounded second moment, i.e., $\mathbb{E}\left[G_j^2(s)\right] < \infty$ for any s.*

Proof. From the definition of $G_j(s)$ and (3.10), it is clear that

$$G_j(s) = \sum_{t=\tau_j}^{\tau_{j+1}-1} \mathcal{G}_t(s) = \sum_{t=\tau_j}^{\tau_{j+1}-1} \left[h(\xi_t^+(s)) - b(\xi_t^-(s))\right] = \Theta'(s).$$

Moreover, we have

$$\begin{aligned}
\mathbb{E}\left[G_j^2(s)\right] &= \mathbb{E}\left[\left(\sum_{t=1}^{\tau_2-1} \left(h(\xi_t^+(s_1)) - b(\xi_t^-(s_1))\right)\right)^2\right] \\
&\leq \mathbb{E}\left[(h \vee b)^2 l_j^2\right] = (h \vee b)^2 \mathbb{E}\left[l_j^2\right] < \infty,
\end{aligned}$$

where the last inequality follows from Lemma 3.13. □

3.5.2 Proof of Proposition 3.9

Proposition 3.9 provides an upper bound on the production cycle cost difference between using the virtual target level $\hat{s}_{\tau_j}$ and using the clairvoyant optimal target level s^*. The proof follows a similar argument used in the general stochastic approximation literature *Nemirovski et al.* (2009) as well as the online convex optimization literature *Hazan* (2016). The main point of departure is due to the *a priori* random cycles, and therefore the proof relies crucially on Lemmas 3.13 and 3.14 previously established.

By optimality of s^*, we have $\mathbb{E}\left[\Omega(s^*, s^*)\right] = \inf_x \left\{\mathbb{E}\left[\Omega(x, s^*)\right]\right\}$, i.e., s^* minimizes the expected single period cost. Also notice that the length of a production cycle is independent of the cycle target level being implemented. Thus, we have

$$\begin{aligned}\mathbb{E}\left[\sum_{j=1}^{J}\Theta(\hat{s}_{\tau_j}) - \sum_{j=1}^{J}\sum_{t=\tau_j}^{\tau_{j+1}-1}\Omega(x_t, s^*)\right] &\leq \mathbb{E}\left[\sum_{j=1}^{J}\Theta(\hat{s}_{\tau_j}) - \sum_{j=1}^{J}\sum_{t=\tau_j}^{\tau_{j+1}-1}\Omega(s^*, s^*)\right] \\ &= \mathbb{E}\left[\sum_{j=1}^{J}\left(\Theta(\hat{s}_{\tau_j}) - \Theta(s^*)\right)\right].\end{aligned} \tag{3.11}$$

By the sample path convexity of $\Theta(\cdot)$ shown in Lemma 3.12, we have

$$\begin{aligned}\mathbb{E}\left[\sum_{j=1}^{J}\left(\Theta(\hat{s}_{\tau_j}) - \Theta(s^*)\right)\right] &\leq \sum_{j=1}^{J}\mathbb{E}\left[\nabla\Theta(\hat{s}_{\tau_j})(\hat{s}_{\tau_j} - s^*)\right] \\ &= \sum_{j=1}^{J}\mathbb{E}\left[G_j(\hat{s}_{\tau_j})(\hat{s}_{\tau_j} - s^*)\right].\end{aligned} \tag{3.12}$$

By the definition of $\hat{s}_{\tau_{j+1}}$ in the DRC algorithm,

$$\begin{aligned}\mathbb{E}\left(\hat{s}_{\tau_{j+1}} - s^*\right)^2 &\leq \mathbb{E}\left(\hat{s}_{\tau_j} - \eta_j G_j(\hat{s}_{\tau_j}) - s^*\right)^2 \\ &= \mathbb{E}\left(\hat{s}_{\tau_j} - s^*\right)^2 + \mathbb{E}\left(\eta_j G_j(\hat{s}_{\tau_j})\right)^2 - \mathbb{E}\left[2\eta_j G_j(\hat{s}_{\tau_j})(\hat{s}_{\tau_j} - s^*)\right] \\ &= \mathbb{E}\left(\hat{s}_{\tau_j} - s^*\right)^2 + \mathbb{E}[\eta_j]\mathbb{E}\left(G_j(\hat{s}_{\tau_j})\right)^2 - 2\mathbb{E}[\eta_j]\mathbb{E}\left[G_j(\hat{s}_{\tau_j})(\hat{s}_{\tau_j} - s^*)\right],\end{aligned}$$

where the second equality holds because the step-size η_j is independent of $\hat{s}_{\tau_j}$ and $G_j(\hat{s}_{\tau_j})$. Thus,

$$\begin{aligned} & \mathbb{E}\left[G_j(\hat{s}_{\tau_j})(\hat{s}_{\tau_j} - s^*)\right] \\ \leq \; & \frac{1}{2\mathbb{E}[\eta_j]}\left(\mathbb{E}\left(\hat{s}_{\tau_j} - s^*\right)^2 - \mathbb{E}\left(\hat{s}_{\tau_{j+1}} - s^*\right)^2\right) + \frac{1}{2}\mathbb{E}\left[\eta_j\left(G_j(\hat{s}_{\tau_j})\right)^2\right]. \end{aligned} \tag{3.13}$$

Combining (3.12) and (3.13), we have

$$\begin{aligned} & \sum_{j=1}^{J}\mathbb{E}\left[\nabla\Theta(\hat{s}_{\tau_j})(\hat{s}_{\tau_j} - s^*)\right] \\ \leq \; & \sum_{j=1}^{J}\left(\frac{1}{2\mathbb{E}[\eta_j]}\left(\mathbb{E}\left(\hat{s}_{\tau_j} - s^*\right)^2 - \mathbb{E}\left(\hat{s}_{\tau_{j+1}} - s^*\right)^2\right) + \frac{1}{2}\mathbb{E}\left[\eta_j\left(G_j(\hat{s}_{\tau_j})\right)^2\right]\right) \\ = \; & \frac{1}{2\mathbb{E}[\eta_1]}\mathbb{E}\left(\hat{s}_{\tau_1} - s^*\right)^2 - \frac{1}{2\mathbb{E}[\eta_j]}\mathbb{E}\left(\hat{s}_{\tau_{j+1}} - s^*\right)^2 \\ & \quad + \frac{1}{2}\sum_{j=2}^{J}\left(\frac{1}{\mathbb{E}[\eta_j]} - \frac{1}{\mathbb{E}[\eta_{j-1}]}\right)\mathbb{E}\left(\hat{s}_{\tau_j} - s^*\right)^2 + \sum_{j=1}^{J}\frac{\mathbb{E}\left[\eta_j\left(G_j(\hat{s}_{\tau_j})\right)^2\right]}{2} \\ \leq \; & 2\bar{s}^2\left(\frac{1}{2\mathbb{E}[\eta_1]} + \frac{1}{2}\sum_{j=2}^{J}\left(\frac{1}{\mathbb{E}[\eta_j]} - \frac{1}{\mathbb{E}[\eta_{j-1}]}\right)\right) + \frac{\mathbb{E}[\left(G_j(\hat{s}_{\tau_j})\right)^2]}{2}\sum_{j=1}^{J}\mathbb{E}[\eta_j] \\ = \; & \frac{\bar{s}^2}{\mathbb{E}[\eta_J]} + \frac{\mathbb{E}[\left(G_j(\hat{s}_{\tau_j})\right)^2]}{2}\sum_{j=1}^{J}\mathbb{E}[\eta_j] \\ \leq \; & K_1\sqrt{T}, \end{aligned}$$

where the last inequality holds due to Lemma 3.14 (bounded second moment of $G(\cdot)$) and

$$\sum_{j=1}^{J}\mathbb{E}[\eta_j] = \gamma\sum_{j=1}^{J}\mathbb{E}\left[1/\sqrt{\sum_{i=1}^{j} l_i}\right] \leq \gamma\sum_{t=1}^{T} 1/\sqrt{t} \leq 2\gamma\sqrt{T}.$$

3.5.3 Proof of Proposition 3.10

Proposition 3.10 provides an upper bound on the production cycle cost difference between using the actual implemented target level s_{τ_j} and using the virtual target

level $\hat{s}_{\tau_j}$. The main idea of this proof on a high level is to set up an upper bounding stochastic process that resembles the waiting time process of a **GI/GI/1** queue. A similar argument appeared *Huh and Rusmevichientong* (2009) and *Shi et al.* (2016). There are two differences. First, the mapping to the waiting time process is more involved in the presence of random capacities. In the above two papers, the resulting level is always higher than the target level, whereas the resulting level could be either higher or lower than the target level in our setting. Second, the present chapter needs to bound the difference in cycle target levels (relying on Lemmas 3.13 and 3.14), rather than per-period target levels.

By the definition of production cycle cost (3.9), we have

$$\begin{aligned}
& \mathbb{E}\left[\Theta(s_{\tau_j}) - \Theta(\hat{s}_{\tau_j})\right] \\
= \; & \mathbb{E}\left[\sum_{t=\tau_j}^{\tau_{j+1}-1} \left[h\left(s_{\tau_j} \wedge (x_t+U_t) - D_t\right)^+ + b\left(D_t - s_{\tau_j} \wedge (x_t+U_t)\right)^+\right]\right. \\
& \qquad \left. - \sum_{t=\tau_j}^{\tau_{j+1}-1} \left[h\left(\hat{s}_{\tau_j} \wedge (x_t+U_t) - D_t\right)^+ + b\left(D_t - \hat{s}_{\tau_j} \wedge (x_t+U_t)\right)^+\right]\right] \\
\leq \; & \mathbb{E}\left[\sum_{t=1}^{l_j-1} (h \vee b)|s_{\tau_j} - \hat{s}_{\tau_j}|\right] \leq \mathbb{E}[l_j](h \vee b)|s_{\tau_j} - \hat{s}_{\tau_j}|,
\end{aligned}$$

where the second inequality holds due to the Wald's Theorem using the fact that l_j is independent of s_{τ_j} and $\hat{s}_{\tau_j}$, and the first inequality follows from the fact that for any $t \in [\tau_j, \tau_{j+1} - 1]$, we have

$$\begin{aligned}
& \mathbb{E}\left[\left[h\left(s_{\tau_j} \wedge (x_t+U_t) - D_t\right)^+ + b\left(D_t - s_{\tau_j} \wedge (x_t+U_t)\right)^+\right]\right. \\
& \qquad \left. - \left[h\left(\hat{s}_{\tau_j} \wedge (x_t+U_t) - D_t\right)^+ + b\left(D_t - \hat{s}_{\tau_j} \wedge (x_t+U_t)\right)^+\right]\right] \\
\leq \; & \mathbb{E}\left[h\left(s_{\tau_j} \wedge (x_t+U_t) - \hat{s}_{\tau_j} \wedge (x_t+U_t)\right)^+\right. \\
& \qquad \left. + b\left(\hat{s}_{\tau_j} \wedge (x_t+U_t) - s_{\tau_j} \wedge (x_t+U_t)\right)^+\right] \\
\leq \; & (h \vee b)\left|s_{\tau_j} - \hat{s}_{\tau_j}\right|.
\end{aligned}$$

Thus, to prove Proposition 3.10, it suffices to prove

$$\mathbb{E}\left[\sum_{j=1}^{J}\Theta(s_{\tau_j})-\sum_{j=1}^{J}\Theta(\hat{s}_{\tau_j})\right]\leq\mathbb{E}[l_j](h\vee b)\mathbb{E}\left[\sum_{j=1}^{J}|s_{\tau_j}-\hat{s}_{\tau_j}|\right]\leq O(\sqrt{T}).$$

Next, we consider an auxiliary stochastic process $(Z_j \mid j \geq 0)$ defined by

$$Z_{j+1}=\left[Z_j+\frac{\gamma\lambda_j}{\sqrt{\sum_{t=1}^{j}l_t}}-\nu_j\right]^+, \tag{3.14}$$

where the random variables $\lambda_j = (h \vee b)l_j$, and $\nu_j = \sum_{t=\tau_j+1}^{\tau_{j+1}} U_t - \sum_{t=\tau_j}^{\tau_{j+1}-1} D_t$, and $Z_0 = 0$. Moreover, since we know that in period τ_{j+1}, the production cycle ends, we must have

$$\nu_j=\sum_{t=\tau_j+1}^{\tau_{j+1}}U_t-\sum_{t=\tau_j}^{\tau_{j+1}-1}D_t\geq 0.$$

Now we want to relate $|\hat{s}_{\tau_j} - s_{\tau_j}|$ to the stochastic process defined above. We can see from the DRC algorithm that the only situation when the virtual target level cannot be achieved is when $\hat{s}_{\tau_j} > s_{\tau_j}$. When $\hat{s}_{\tau_j} \leq s_{\tau_j}$, we can salvage extra inventory and achieve the virtual target level. Therefore, we relate $|\hat{s}_{\tau_j} - s_{\tau_j}|$ with the stochastic process Z_j.

Lemma 3.15. *For any $j \geq 1$,*

$$\mathbb{E}\left[\sum_{j=1}^{J}|s_{\tau_j}-\hat{s}_{\tau_j}|\right]\leq\mathbb{E}\left[\sum_{j=1}^{J}Z_j\right],$$

where $\{Z_j, j \geq 1\}$ is the stochastic process we define above.

Proof. All the stochastic comparisons within this proof are with probability one. When $\hat{s}_{\tau_{j+1}} < x_{\tau_{j+1}} + U_{\tau_{j+1}}$, we have $\hat{s}_{\tau_{j+1}} - s_{\tau_{j+1}} = 0 \leq Z_{j+1}$. When $\hat{s}_{\tau_{j+1}} > x_{\tau_{j+1}} + U_{\tau_{j+1}}$, we have $s_{\tau_{j+1}} = x_{\tau_{j+1}} + U_{\tau_{j+1}} = s_{\tau_j} - \sum_{t=\tau_j}^{\tau_{j+1}-1} D_t + \sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t + U_{\tau_j+1}$. Therefore,

we have

$$
\begin{aligned}
& \left|\hat{s}_{\tau_{j+1}} - s_{\tau_{j+1}}\right| \\
= \;& \hat{s}_{\tau_{j+1}} - s_{\tau_{j+1}} = \mathbf{P}_{[0,\bar{s}]}\left(\hat{s}_{\tau_j} - \eta_j G_j(\hat{s}_{\tau_j})\right) - s_{\tau_{j+1}} \leq \left|\mathbf{P}_{[0,\bar{s}]}\left(\hat{s}_{\tau_j} - \eta_j G_j(\hat{s}_{\tau_j})\right)\right| - s_{\tau_{j+1}} \\
\leq \;& \left|\hat{s}_{\tau_j} - \eta_j G_j(\hat{s}_{\tau_j})\right| - s_{\tau_j} + \left(\sum_{t=\tau_j}^{\tau_{j+1}-1} D_t - \sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t\right) - U_{\tau_{j+1}} \\
\leq \;& \left|\hat{s}_{\tau_j} - s_{\tau_j} - \eta_j G_j(\hat{s}_{\tau_j})\right| + \left(\sum_{t=\tau_j}^{\tau_{j+1}-1} D_t - \sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t\right) - U_{\tau_{j+1}} \\
\leq \;& \left|\hat{s}_{\tau_j} - s_{\tau_j}\right| + \left|\eta_j G_j(\hat{s}_{\tau_j})\right| - \left(\sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t - \sum_{t=\tau_j}^{\tau_{j+1}-1} D_t\right) \\
\leq \;& \left|\hat{s}_{\tau_j} - s_{\tau_j}\right| + \eta_j (h \vee b) \cdot l_j - \left(\sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t - \sum_{t=\tau_j}^{\tau_{j+1}-1} D_t\right),
\end{aligned}
$$

where the first equality holds because following the DRC algorithm, we always have $s_{\tau_j} \leq \hat{s}_{\tau_j}$. The third inequality holds because s_{τ_j} is always nonnegative. This is because the virtual target level is truncated to be nonnegative all the time, and we update the actual implemented target level when the production cycle ends, which means after the previous actual implemented target level is achieved. Since $s_1 \geq 0$, $s_{\tau_j} \geq 0$ for all j. The fourth inequality holds because of the triangular inequality and the last inequality holds because $|G_j(\hat{s}_{\tau_j})| \leq (h \vee b) \cdot l_j$.

Therefore, from the above claim we have

$$
\left|s_{\tau_{j+1}} - \hat{s}_{\tau_{j+1}}\right| \leq \left[\left|s_{\tau_j} - \hat{s}_{\tau_j}\right| + \eta_j (h \vee b) l_j - \left(\sum_{t=\tau_j+1}^{\tau_{j+1}-1} U_t - \sum_{t=\tau_j}^{\tau_{j+1}-1} D_t\right)\right]^+.
$$

Comparing to (3.14), we have

$$
\eta_j (h \vee b) l_j \leq \frac{\gamma \lambda_j}{\sqrt{\sum_{t=1}^{j} l_t}},
$$

and since $s_1 - \hat{s}_1 = 0$, it follows, from the recursive definition of Z_j, that $|s_{\tau_{j+1}} - \hat{s}_{\tau_{j+1}}| \leq Z_{j+1}$ holds with probability one. Summing up both sides of the inequality completes the proof. □

We observe that the stochastic process Z_j is very similar to the waiting time in a **GI/GI/1** queue, except that the service time is scaled by $\gamma/\sqrt{\sum_{i=1}^{j} l_i}$ in each production cycle j. Now consider a **GI/GI/1** queue $(W_j \mid j \geq 0)$ defined by the following Lindley's equation: $W_0 = 0$, and

$$W_{j+1} = [W_j + \lambda_j - \nu_j]^+ , \tag{3.15}$$

where the sequences λ_j and ν_j consist of independent and identically distributed random variables (only dependent upon the distributions of D and U). Let $\varphi_0 = 0$, $\varphi_1 = \inf\{t \geq 1 : W_j = 0\}$ and for $t \geq 1$, $\varphi_{t+1} = \inf\{t > \varphi_t : W_j = 0\}$. Let $B_t = \varphi_t - \varphi_{t-1}$. The random variable W_j is the waiting time of the j^{th} customer in the **GI/GI/1** queue, where the inter-arrival time between the j^{th} and $j+1^{th}$ customers is distributed as ν_j, and the service time is distributed as λ_j. Then, B_t is the length of the t^{th} busy period. Let $\rho = \mathbb{E}[\lambda_1]/\mathbb{E}[\nu_1]$ represent the system utilization. Note that if $\rho < 1$, then the queue is stable, and the random variable B_t is independent and identically distributed.

We invoke the following result from *Loulou* (1978) to bound $\mathbb{E}[B_t]$, the expected busy period of a $GI/G/1$ queue with inter-arrival distribution ν and service time λ.

Lemma 3.16 (*Loulou* (1978))**.** *Let $X_j = \lambda_j - \nu_j$, and $\alpha = -\mathbb{E}[X_1]$. Let σ^2 be the variance of X_1. If $\mathbb{E}[X_1]^3 = \beta < \infty$, and $\rho < 1$,*

$$\mathbb{E}[B_1] \leq \frac{\sigma}{\alpha} \exp\left(\frac{6\beta^3}{\sigma^3} + \frac{\alpha}{\sigma}\right).$$

For each $n \geq 1$, let the random variable $i(n)$ denote the index t such that B_t

contains n. This means that the n^{th} customer is within the $B_{i(n)}$ busy period. Since B_t is i.i.d., we know that $\mathbb{E}[B_{i(n)}] = \mathbb{E}[B_t] = \mathbb{E}[B_1]$.

Lemma 3.17. *For any period $t \geq 1$, we have*

$$\mathbb{E}\left[\sum_{j=1}^{J} Z_j\right] \leq 2\gamma(h \vee b)\mathbb{E}[B_1]\sqrt{T}.$$

Proof. As defined above, the stochastic process $Z_{j+1} = \left[Z_j + \frac{\gamma\lambda_j}{\sqrt{\sum_{i=1}^{j} l_i}} - \nu_j\right]^+$. Since Z_j can be interpreted as the waiting time in the **GI/GI/1** queueing system, we can rewrite Z_j as

$$Z_j = \sum_{j'=1}^{j}\left(\frac{\gamma\lambda_{j'}}{\sqrt{\sum_{i=1}^{j'} l_i}} - \nu_{j'}\right)\mathbb{1}\left[j' \in B_{i(j)}\right] \leq \sum_{j'=1}^{j}\frac{\gamma\lambda_{j'}}{\sqrt{\sum_{i=1}^{j'} l_i}}\mathbb{1}\left[j' \in B_{i(j)}\right]. \quad (3.16)$$

We then bound the total waiting time of sequence Z_j by only considering the cumulative service times as follows.

$$\begin{aligned}
\mathbb{E}\left[\sum_{j=1}^{J} Z_j\right] &= \mathbb{E}\left[\sum_{j=1}^{J}\sum_{j'=1}^{j}\frac{\gamma\lambda_{j'}}{\sqrt{\sum_{i=1}^{j'} l_i}}\mathbb{1}[j' \in B_{i(j)}]\right] \\
&\leq \mathbb{E}\left[\sum_{j=1}^{J}\sum_{j'=1}^{J}\frac{\gamma(h \vee b)l_{j'}}{\sqrt{\sum_{i=1}^{j'} l_i}}\mathbb{1}[j' \in B_{i(j)}]\right] \\
&\leq \mathbb{E}\left[\sum_{j'=1}^{J}\frac{\gamma(h \vee b)l_{j'}}{\sqrt{\sum_{i=1}^{j'} l_i}}\sum_{j=1}^{J}\mathbb{1}[j' \in B_{i(j)}]\right] \\
&= \mathbb{E}\left[\sum_{j'=1}^{J}\frac{\gamma(h \vee b)l_{j'}}{\sqrt{\sum_{i=1}^{j'} l_i}}B_{i(j')}\right] \leq \mathbb{E}\left[\sum_{t=1}^{T}\frac{\gamma(h \vee b)}{\sqrt{t}}B_{i(t)}\right],
\end{aligned}$$

where the last inequality holds because

$$\sum_{j'=1}^{J}\frac{l_{j'}}{\sqrt{\sum_{i=1}^{j'} l_i}} \leq \sum_{t=1}^{T}\frac{1}{\sqrt{t}}, \quad \text{where} \quad T = \sum_{j'=1}^{J} l_{j'}.$$

Thus, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{j=1}^{J} Z_j\right] &\leq \mathbb{E}\left[\sum_{t=1}^{T} \frac{\gamma(h \vee b)}{\sqrt{t}} B_{i(t)}\right] \\
&= \gamma(h \vee b)\mathbb{E}\left[\sum_{t=1}^{T} \frac{1}{\sqrt{t}}\right] \mathbb{E}[B_{i(t)}] \\
&\leq 2\gamma(h \vee b)\sqrt{T}\mathbb{E}[B_1], \qquad (3.17)
\end{aligned}$$

where the last inequality follows from the fact that $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\sqrt{T} - 1$. Combining (3.16) and (3.17) completes the proof. □

Combining Lemmas 3.15 and 3.17, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{j=1}^{J} \Theta(s_{\tau_j}) - \sum_{j=1}^{J} \Theta(\hat{s}_{\tau_j})\right] &\leq \mathbb{E}\left[\sum_{j=1}^{J} \gamma(h \vee b)(\hat{s}_{\tau_j} - s_{\tau_j})\right] \\
&\leq \gamma(h \vee b)\mathbb{E}[l_1]\mathbb{E}\left[\sum_{j=1}^{J} Z_j\right] \\
&\leq 2\gamma(h \vee b)^2\mathbb{E}[l_1]\mathbb{E}[B]\sqrt{T},
\end{aligned}$$

where both $\mathbb{E}[B]$ and $\mathbb{E}[l_1]$ are bounded constants. This completes the proof for Proposition 3.10.

3.5.4 Proof of Proposition 3.11

Proposition 3.11 provides an upper bound on the cumulative production and salvaging costs incurred by adjusting the production cycle target levels. The main idea of this proof on a high level is to use the fact that the cycle target levels of the actual implemented system are getting closer to the ones of the virtual system over time, and each change in the cycle target level can be sufficiently bounded, resulting in an

upper bound on the cumulative production and salvaging costs.

$$\begin{aligned}
& \mathbb{E}\left[\sum_{j=1}^{J} c\left(s_{\tau_{j+1}} - s_{\tau_j}\right)^+\right] \leq \mathbb{E}\left[\sum_{j=1}^{J} c\left(\hat{s}_{\tau_{j+1}} - s_{\tau_j}\right)^+\right] \\
= & \mathbb{E}\left[\sum_{j=1}^{J} c\left(\mathbf{P}_{[0,\bar{s}]}\left(\hat{s}_{\tau_j} - \eta_j \cdot G_j(\hat{s}_{\tau_j})\right) - s_{\tau_j}\right)^+\right] \\
\leq & \mathbb{E}\left[\sum_{j=1}^{J} c\left(\left(\hat{s}_{\tau_j} - \eta_j \cdot G_j(\hat{s}_{\tau_j})\right) - s_{\tau_j}\right)^+\right] \\
\leq & \mathbb{E}\left[\sum_{j=1}^{J} c\left|\hat{s}_{\tau_j} - s_{\tau_j}\right| + \sum_{j=1}^{J} c\left|\eta_j \cdot G_j(\hat{s}_{\tau_j})\right|\right] \leq K_4\sqrt{T},
\end{aligned}$$

where K_4 is some positive constant. The result trivially holds if $s_{\tau_{j+1}} \leq s_{\tau_j}$. Now, consider the case where $s_{\tau_{j+1}} > s_{\tau_j}$, i.e., the firm produces. The first inequality holds because if the firm produces, we must have $s_{\tau_{j+1}} \leq \hat{s}_{\tau_{j+1}}$ by the construction of DRC. The second inequality holds because $s_{\tau_j} \geq 0$. The third inequality holds by the triangular inequality. The last inequality is due to the fact that $\sum_{j=1}^{J}\left|\hat{s}_{\tau_j} - s_{\tau_j}\right| \leq O(\sqrt{T})$ from Proposition 3.10, and

$$\sum_{j=1}^{J} c\left|\eta_j \cdot G_j(\hat{s}_{\tau_j})\right| \leq c\gamma(h \vee b)\sum_{j=1}^{J}\frac{l_j}{\sqrt{\sum_{i=1}^{j} l_i}} \leq 2c\gamma(h \vee b)\sqrt{T}.$$

Similarly,

$$\begin{aligned}
& \mathbb{E}\left[\sum_{j=1}^{J} \theta(s_{\tau_j} - s_{\tau_{j+1}})^+\right] = \mathbb{E}\left[\sum_{j=1}^{J} \theta(s_{\tau_j} - \hat{s}_{\tau_{j+1}})^+\right] \\
= & \mathbb{E}\left[\sum_{j=1}^{J} \theta\left(s_{\tau_j} - \mathbf{P}_{[0,\bar{s}]}\left(\hat{s}_{\tau_j} - \eta_j \cdot G_j(\hat{s}_{\tau_j})\right)\right)^+\right] \\
\leq & \mathbb{E}\left[\sum_{j=1}^{J} \theta\left(s_{\tau_j} - \left(\hat{s}_{\tau_j} - \eta_j \cdot G_j(\hat{s}_{\tau_j})\right)\right)^+\right] \\
\leq & \mathbb{E}\left[\sum_{j=1}^{J} \theta\left|\hat{s}_{\tau_j} - s_{\tau_j}\right| + \sum_{j=1}^{J} \theta\left|\eta_j \cdot G_j(\hat{s}_{\tau_j})\right|\right] \leq K_5\sqrt{T},
\end{aligned}$$

where K_5 is some positive constant. The result trivially holds if $s_{\tau_j} \leq s_{\tau_{j+1}}$. Now, consider the case where $s_{\tau_j} > s_{\tau_{j+1}}$, i.e., the firm salvages. The first equality holds because if the firm salvages, we must have $s_{\tau_{j+1}} = \hat{s}_{\tau_{j+1}}$ by the construction of DRC. The first inequality holds because $\bar{s} \geq s_{\tau_j}$. The second inequality holds by the triangular inequality. The last inequality follows the same idea as in the first part of this section.

Combing the above two parts completes the proof of Proposition 3.11.

Finally, Theorem 3.6 is a direct consequence of Propositions 3.9, 3.10, and 3.11, which gives us the desired regret upper bound.

3.6 Numerical Experiments

We conduct numerical experiments to demonstrate the efficacy of our proposed DRC algorithm. To the best of our knowledge, we are not aware of any existing learning algorithms that are applicable to random capacitated inventory systems. Thus, we have designed two simple heuristic learning algorithms (that are intuitively sound and practical), and use them as benchmarks to validate the performance of the DRC algorithm. Our results show that the performance of the DRC algorithms is superior to these two benchmarking heuristics both in terms of consistency and convergence rate. All the simulations were implemented on an Intel Xeon 3.50GHz PC.

3.6.1 Design of Experiments

We conduct our numerical experiments using a normal distribution for the random demand and a mixture of two normal distributions for the random capacity. More specifically, we set the demand to be $\mathbf{N}(10, 3^2)$. We test four different capacity distributions, namely, a mixture of 20% $\mathbf{N}(5, 1^2)$ and 80% $\mathbf{N}(14, 4^2)$, a mixture of 20% $\mathbf{N}(5, 1^2)$ and 80% $\mathbf{N}(17, 5^2)$, a mixture of 20% $\mathbf{N}(5, 1^2)$ and 80% $\mathbf{N}(20, 6^2)$,

and also a mixture of 20% $\mathbf{N}(5, 3^2)$ and 80% $\mathbf{N}(17, 5^2)$. The distributions correspond to environments where the product capacity is subject to downtime. Clearly, in a production environment, capacity may be random even if no significant downtime occurs (e.g., due to variations in operator speed). However, machine downtime can significantly impact capacity. These examples correspond to situations where the production system experiences downtime that affects capacity with 20% probability. (We have experimented with other examples of downtime and obtained similar results.)

The production cost $c = 10$, and the salvaging value is set to be half of the production cost, i.e., $\theta = 5$. The backlogging cost is linear in backorder quantity, with per-unit cost $b = 10$, and the holding cost is 2% per period of the production cost, i.e., $h = 0.2$. We set the time horizon $T = 1000$, and compare the average cost of our DRC algorithm with that of the two benchmarking heuristic algorithms (described below) as well as the clairvoyant optimal cost over 1000 periods.

Clairvoyant Optimal Policy: The clairvoyant optimal policy is a stationary policy, given that the firm knows both the demand and capacity distributions at the beginning of the planning horizon. The average cost is calculated by averaging 1000 runs over 1000 periods.

Benchmarking Heuristic 1: We start with an arbitrary inventory level s_1 and start the first production cycle. For $t \geq 1$, we keep the target level $s_t = s_j$ the same during one production cycle $j \geq 1$. If the inventory level y_t reaches s_j, we claim that the j^{th} production cycle ends and then we collect all the past observed demand data to form an empirical demand distribution and all the past observed capacity data (except the capacity data obtained at the end of each production cycle) to form an empirical capacity distribution. We omit the capacity data obtained at the end of each production cycle because we might not produce at full capacity (when the previous target level is achieved). Then we treat the updated empirical demand and capacity distributions as true distributions, and derive the *long-run* optimal target

level s_{j+1} for the subsequent cycle $j+1$. Note that the long-run optimal target level (with well-defined input demand and capacity distributions) can be computed using the detailed computational procedure described in *Ciarallo et al.* (1994). The average cost is calculated by averaging 1000 runs over 1000 periods.

Benchmarking Heuristic 2: We start with an arbitrary inventory level s_1, and keep the target level $s_t = s_j$ the same during one production cycle $j \geq 1$. We still update the empirical demand distribution at the end of each production cycle using all past observed demand data. However, in the first $N = 10$ periods, we always try to produce up to the maximum capacity $\bar{u}$, and we form the empirical capacity distribution using only these N full capacity sample points, and treat the empirical capacity distribution as the true capacity distribution for the rest of decision horizon. At the end of each production cycle, we still collect all the past observed demand data to form an empirical demand distribution, and similar to heuristic 1, derive the long-run optimal target level for the subsequent cycle together with the empirical capacity distribution. In other words, in the first N periods, we always produce up to the full capacity instead of the target level to get true information of the capacity, and after N periods, we carry out a regular modified base-stock policy. The average cost is calculated by averaging 1000 runs over 1000 periods. We have experimented with N values different than 10 and our results are similar to those we report below.

3.6.2 Numerical Results and Findings

The numerical results are presented in Figure 3.5. We observe that Heuristic 1 is inconsistent, i.e., it fails to converge to the clairvoyant optimal cost. This is because even if we collect all the capacity data only when we produce at full capacity, the empirical distribution formed by these data is still biased (as the capacity data we observe is smaller than the true capacity). Heuristic 2 performs better than Heuristic 1, but still suffers from inconsistency.

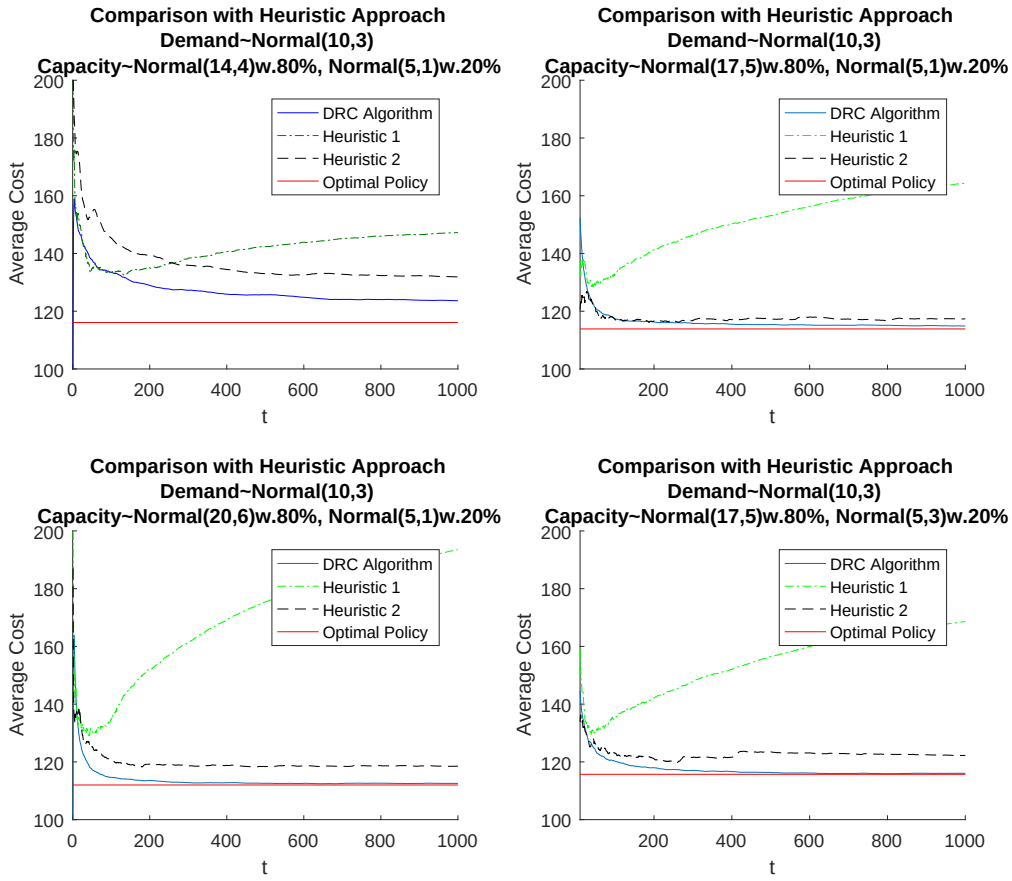

Figure 3.5: Computational performance of the DRC algorithm

Comparing to the benchmarking heuristic algorithms, the DRC algorithm converges to the clairvoyant optimal cost consistently and also at a much faster rate. We can also observe that when the capacity utilization (defined as the mean demand over the mean capacity) increases, the convergence rate slows down. This is because when the capacity utilization is high, it generally takes more periods for the system to reach the previous target level, resulting in longer production cycle length and slower updating frequency. Finally, we find that increasing the variability of distributions does not affect the performance of the DRC algorithm.

3.7 Concluding Remark

In this chapter, we have proposed a stochastic gradient descent type of algorithm for the stochastic inventory systems with random production capacity constraints, where the capacity is censored. Our algorithm utilizes the fact that the clairvoyant optimal policy is the extended myopic policy and updates the target inventory level in a cyclic manner. We have shown that the average T-period cost of our algorithm converges to the optimal cost at the rate of $O(1/\sqrt{T})$, which is the best achievable convergence rate. To the best of our knowledge, this chapter is the first to study learning algorithms for stochastic inventory systems under uncertain capacity constraints. We have also compared our algorithm with two straw heuristic algorithms that are easy to use, and we have shown that our proposed algorithm performs significantly better than the heuristics in both consistency and efficiency. Indeed, our numerical experiments have shown that with censored capacity information, the heuristics may not converge to the optimal policy.

To close this chapter, we leave an important *open question* on how to design an efficient and effective learning algorithm for the capacitated inventory systems with lost-sales and censored demand. In the present chapter, with backlogging demand, the length of the production cycle is independent of the target level, and therefore the production cycles in our proposed algorithm and the optimal system are perfectly aligned. With lost-sales and censored demand, the length of the production cycle becomes dependent on the target level, and comparing any two feasible policies becomes much more challenging, which would require significantly new ideas and techniques.

CHAPTER IV

Optimal Learning Algorithms for Make-To-Stock Queueing Systems

4.1 Introduction

We consider a classical infinite-horizon $M/G/1$ make-to-stock queueing system that arises in many practical production settings. There is a single facility which is dedicated to producing a single product type. The demand arrival process (of customers) is a Poisson process with rate λ, that is, the inter-arrival time between successive arrivals is denoted by an exponential random variable R with rate λ. The production time of each product in this single facility is random, which is denoted by an independent and identically distributed (i.i.d.) random variable U with finite mean $\mathbb{E}[U]$. The probability density function (p.d.f.) and cumulative distribution function (c.d.f.) of U are denoted by $f_U(\cdot)$, $F_U(\cdot)$, respectively.

The facility is either producing or idling, and the setup time for the facility to switch between these two modes is assumed to be negligible. While the facility is producing, the output is continuous and goes directly into the onhand inventory. Demand is satisfied from the onhand inventory on a first-come-first-served basis. If a customer arrives at the system with zero onhand inventory, the demand will be backlogged. The system incurs an inventory holding cost h per unit product per

unit time whenever the inventory is positive, and incurs a backlogging cost b per unit product per unit time whenever the inventory is negative (or backlogged). The objective is to minimize the long-run average expected sum of holding and backlogging costs.

4.1.1 Main result and our contribution

Different than the existing literature, at the beginning of the planning horizon, the decision maker has no prior information about the underlying distribution of the production time U as well as the customer arrival rate λ. The performance measure considered in our setting is the notion of *regret*, which is defined as the difference in cost between a feasible adaptive control policy (that does not have the prior distributional information but only relies on past observations) and a clairvoyant optimal policy (had the distributional information about U and λ been known). The main result of this paper is to devise an efficient adaptive control policy and prove that the cumulative regret $\mathcal{R}_T \leq O(\sqrt{T})$ for a T-period problem. In other words, the average T-period running cost converges to the clairvoyant optimal cost at $O(1/\sqrt{T})$, which is also shown to be tight (formally stated in Theorem 4.2).

4.1.2 Relevant literature

With complete distributional information on both the arrival and production times, this problem has been studied extensively in the literature (see, e.g., *Evans* (1967), *Sobel* (1982), *Gavish and Graves* (1980), *Federgruen and Zipkin* (1986a)). We refer interested readers to *Kapuscinski and Tayur* (1999) for a comprehensive survey. The optimal policy is typically of the base-stock type (i.e., the facility produces when inventory falls below a certain threshold and idles otherwise), which was first proved by *Gavish and Graves* (1980) and *Sobel* (1982) for the single product and single facility case. *Zheng and Zipkin* (1990) studied the policy for two symmetric products,

and the results were generalized to various multiproduct settings (see e.g., *H. Zipkin* (1995), *Wein* (1992), *Veatch and Wein* (1996), *Bertsimas and Paschalidis* (2001)). There are also several extensions to the single product case but with multiple demand classes (see e.g., *Ha* (1997a), *Ha* (1997b), *Ha* (2000)). The make-to-stock queues have also been studied in the context of pricing and admission control. *Li* (1992) considered a single product but with congestion and exogenous price. *Scott Carr* (2000) considered a make-to-stock production system where both sequencing and admission control decisions are made. *Caldentey and Wein* (2006) discussed a single-product make-to-stock system with two pricing options. There is a large body of literature incorporating more detailed modeling of the production facility, including tandem queues (see e.g., *Kapuscinski and Tayur* (1999), *Ahn et al.* (2002), *Ahn et al.* (1999), *Iravani et al.* (1997), *Duenyas et al.* (1998)), and unreliable production facilities (see e.g., *Feng and Yan* (2000), *Feng and Xiao* (2002)). There is also extensive body of queueing theory on admission control (see e.g., *Lippman* (1975), *Lippman and Stidham* (1977), *Stidham* (1978)). A comprehensive survey can be referred to *Crabill et al.* (1977) and *Stidham* (1985).

When there is no prior knowledge about the arrival rate and the distribution of production times, there is very little literature considering the joint learning and optimal control problem. The present chapter aims to fill in this important gap, by devising an adaptive algorithm with provably tight convergence rate to the clairvoyant optimal solution. Our algorithm is stochastic gradient descent type, motivated by the literature on robust stochastic approximation (see e.g., *Nemirovski et al.* (2009) and references therein) and online convex optimization (see e.g., *Hazan* (2016) and references therein). There has been some recent progress for the discrete-time production-inventory systems under incomplete information, giving rise to efficient learning algorithms for various models (see e.g., *Burnetas and Smith* (2000), *Huh and Rusmevichientong* (2009), *Huh et al.* (2009), *Shi et al.* (2016), *Zhang et al.*

(2018),*Zhang et al.* (2019), *Chen et al.* (2018a)). However, we note that the aforementioned learning problems are designed primarily for discrete time review systems with only the demand (arrival) distribution information unknown *a prior*. For the continuous review system considered in this chapter, in addition to the unknown arrival rate, we also need to learn the distribution of the production time while minimizing the total costs on the fly. As a result, the design and analysis of the proposed algorithm become more challenging. We hope that this work could open many future research avenues for joint learning and control for queueing systems.

4.1.3 Organization

The rest of the chapter is organized as follows. We formally present our model in §4.2. We give our learning algorithm in §4.3 and its performance analysis in §4.4. We conduct a numerical study to demonstrate the efficacy of the proposed algorithm in §4.5. We conclude this paper and point out several future research directions in §4.6.

4.2 Model, System Dynamics, and Costs

Consider an infinite-horizon make-to-stock inventory system wherein a single production facility is dedicated to producing one product. The facility can be set up to produce or turn down and idle. While the facility is producing, the output goes directly to the inventory. The demand arrival process (of customers) is a Poisson process with rate λ, and is supplied from the inventory. When the inventory is not available, the demand is backlogged (customer will wait). The inventory holding cost is h and the backlogging cost is b per inventory per unit time. The production time have distribution $F_R(\cdot)$ and density $f_R(\cdot)$.

At any time t, the decision maker can observe its inventory level $x(t)$. If a product finishes at time τ, then we consider the inventory level $x(\tau) = x(\tau^-) + 1$, where $x(\tau^-)$ is the time instance just before τ. Similarly, if a customer arrives at time τ, then we

consider the inventory level $x(\tau) = x(\tau^-) - 1$. Let the initial inventory level be $x(0)$, the inventory level at time t can be written as $x(t) = x(0) + P(0,t) - D(0,t)$ where $P(0,t)$ and $D(0,t)$ are the total number of products produced and the total number of customers arrived at and before time t, respectively.

The decision is when to turn on or off the facility. More precisely, if the facility is idle at time t, then when the next customer arrives and consumes one product from the inventory, we need to decide whether to start production or not. On the other hand, if the facility is producing at time t, then when the current product is finished, we need to decide whether to continue production or to set the facility idle.

At the beginning of the planning horizon, the decision maker has no prior knowledge about the customer arriving rate as well as the underlying production time distribution. At any time t, the decision maker has access to all past customer arrival times and all past production times up to time t. Define the decision epoch to be the time whenever a product is finished or a customer arrives. At each decision epoch, the decision maker will decide its target inventory level $s(t)$ and make the decision accordingly.

Let $\mathcal{H}(t)$ denote the information collected up to time t. Our objective is to find an adaptive policy π, or a series of inventory target levels $s(t) := \pi(\mathcal{H}(t), x(t))$ which minimizes the long run average expected cost

$$\limsup_{T\to\infty} \mathbb{E}\left[\frac{\int_0^T \left(hx(t)^+ + bx(t)^-\right) dt}{T}\right],$$

where $x(t)^+ = \max(x(t), 0)$ is the positive inventory in the system at time t, and $x(t)^- = -\min(x(t), 0)$ is the backlogging demand in the system at time t.

Symbol	Type	Description
h	Param	Per-unit per-unit-of-time holding cost.
b	Param	Per-unit per-unit-of-time backlogging cost.
$D(t_1, t_2), d(t_1, t_2)$	Param	Random demand and its realization within time interval $[t_1, t_2)$.
$P(t_1, t_2), p(t_1, t_2)$	Param	Random production and its realization within time interval $[t_1, t_2)$.
R_i, r_i	Param	Random inter-arrival time between ith and $(i+1)$th customer.
F_R, f_R	Param	Inter-arrival time c.d.f. and p.d.f.
U_j, u_j	Param	Random production time for the jth product.
F_U, f_U	Param	Production time c.d.f. and p.d.f.
$x(t)$	State	Initial inventory level at time t.
$s(t)$	State	Target inventory level at time t.
s_k	State	Target inventory level of the kth production cycle.
$\hat{s}_k$	State	(Integer) virtual target inventory level of the kth production cycle.
$\tilde{s}_k$	State	(float) virtual target inventory level of the kth production cycle.
α_i	State	The time when the ith product finishes.
β_j	State	The time when the jth customer shows up.
L_k	State	The cycle length of the kth production cycle.

Table 4.1: Summary of Major Notation

4.3 An Adaptive Learning Algorithm

Our algorithm utilizes a stochastic gradient updating rule to ensure that it converges to the optimal policy. We first introduce the concept of a *production cycle*, which gives rise to a renewal process in a queueing system. A production cycle is defined as the time elapsed between two successive hits of a certain inventory target level, i.e., the duration which begins when the inventory level hits the target level and ends when the inventory level is brought back to the same target level again.

In the clairvoyant problem, it is well known that a base-stock type policy is optimal. That is, there exists an optimal target level s^* such that the facility keeps producing whenever the inventory level $x(t) < s^*$ and shuts down and stays idle when the inventory level is brought back to s^*, i.e., $x(t) = s^*$. Let

$$\begin{aligned} \alpha_i &= \text{the time epoch when the } i\text{th product is finished} \qquad \text{for} \quad i = 1, 2, \ldots \\ \beta_j &= \text{the time epoch when the } j\text{th customer arrives} \qquad \text{for} \quad j = 1, 2, \ldots \end{aligned}$$

Figure 4.1(a) illustrates a sample path example of the clairvoyant system. The decision maker will always try to produce up to the optimal inventory level s^*. At α_5, the inventory level reaches s^*, then a production cycle starts at α_5 and ends at α_8 when the inventory level reaches s^* again. The next production cycle starts at α_8 and ends at α_{14}. A production cycle consists of an "off" cycle $[\alpha_5, \beta_6]$ when the facility is idle, and an "on" cycle $[\beta_6, \alpha_8]$ when the facility is producing. We call such a production cycle as a standard production cycle. Note that for the clairvoyant problem, the off cycle corresponds to the idle period and the on cycle corresponds to the busy period in an $M/G/1$ queue.

However, in the incomplete information model, the decision maker does not know s^*, and therefore needs to update the target level based on past realizations. Figures 4.1(b) and 4.1(c) show two possible cases. Suppose the initial target level is $\hat{s}_1$. Then

a standard production cycle starts at α_5 and ends at α_8, according to the above definition. In the first case shown by Figure 4.1(b), a new target level $\hat{s}_2 > \hat{s}_1$ is suggested by an algorithm π. At α_8, the facility will keep producing until the inventory level reaches $\hat{s}_2$ at α_{15}. Then α_{15} marks the start of the second standard production cycle. In this case, the transition period from $\hat{s}_1$ to $\hat{s}_2$ is called "busy transition period" (since the facility is trying to produce up to achieve the new target level).

On the other hand, in the second case shown by Figure 4.1(c), a new target level $\hat{s}_2 \leq \hat{s}_1$ is suggested by an algorithm π. At α_8, the facility shuts down and stays idle until enough customers arrive to bring the inventory level down to $\hat{s}_2$ at β_{11}. Then β_{11} marks the start of the second standard production cycle. In this case, the transition period from $\hat{s}_1$ to $\hat{s}_2$ is called "idle transition period" (since the facility is trying to stay idle to lower the inventory to achieve the new target level).

In order to correctly carry out the updates, we shall only utilize the information collected from a standard production cycle, thanks to the convexity property (shown in Lemma 4.3). We remark that $[\alpha_{10}, \alpha_{14}]$ in Figure 4.1(b) also forms a production cycle if we treat the inventory level at α_{10} as a target level. The difference between this production cycle and the ones previously discussed is that the facility is not idle at the beginning of this production cycle. Thus, $[\alpha_{10}, \alpha_{14}]$ does not form a standard production cycle, and a bias will be introduced if we use the information collected from $[\alpha_{10}, \alpha_{14}]$ to update the target level (which will be shown to vanish in the proposed algorithm at an appropriate rate).

4.3.1 Algorithm Description

Assumption 4.1. *We make the following assumptions.*

(a) The utilization factor $\rho = \lambda \mathbb{E}[U] < 1$.

(b) The optimal target level lies in a bounded interval $[0, \bar{s}]$.

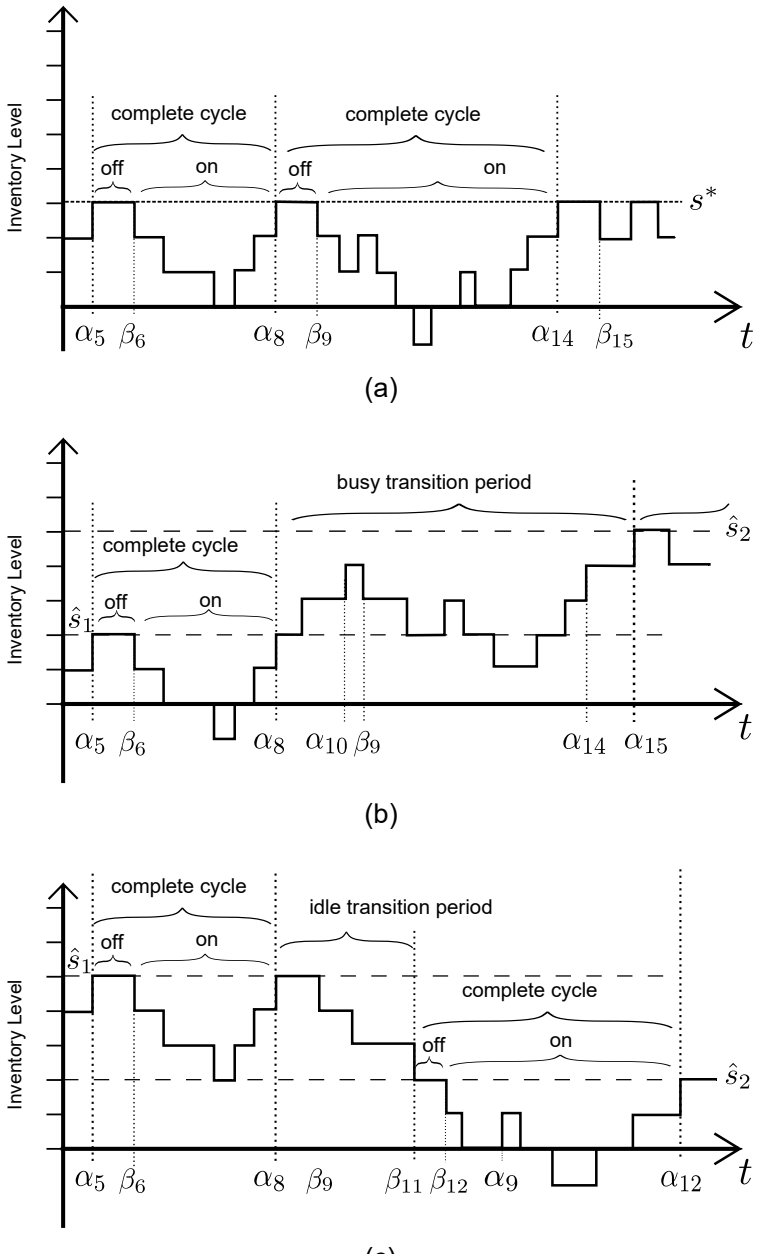

Figure 4.1: Illustration of the production cycles and dynamics of different policies

Now we shall introduce our data-driven algorithm for the make-to-stock queue (DMTS for short). The DMTS algorithm has two main design principles. One is to utilize the information collected from a standard production cycle to carry out unbiased updates, and the other one is to leverage the information in the transition period (between updating and actually attaining the target level) to improve efficiency.

We maintain two systems throughout the algorithm. The first system is the actual implemented system, where the algorithm keeps track of $x(t)$ as the actual inventory level at time t, α_i as the ith product completion time, and β_j as the jth customer arrival time. Based on $x(t), \alpha_i, \beta_j$ collected from the actual implemented system, we construct the second (infeasible) system termed the virtual system, which is a series of standard production cycles (that minimize the gaps between any two consecutive cycles). We refer to the standard production cycles in the virtual system as the virtual production cycles. The virtual system records τ_k^s as the starting time for the kth virtual production cycle, and τ_k^e as the ending time for the kth virtual production cycle. The algorithm maintains the virtual target level $\hat{s}_k$ for the kth virtual production cycle. At the beginning of the kth virtual production cycle, the virtual system *artificially* sets the virtual inventory level $\hat{x}(\tau_k^s) = \hat{s}_k$. At the end of the kth virtual production cycle, the algorithm computes the (fractional) virtual target level $\tilde{s}_{k+1}$ for the next production cycle, which is then rounded to an integer value $\hat{s}_{k+1}$.

The algorithm always wants to bring the actual inventory level $x(t)$ up to virtual target level $\hat{s}_{k+1}$ for $t \in [\tau_k^e, \tau_{k+1}^e]$, i.e., the facility will keep producing if $x(t) < \hat{s}_{k+1}$ and stays idle if $x(t) \geq \hat{s}_{k+1}$ for $t \in [\tau_k^e, \tau_{k+1}^e]$. However, a caveat is that $\hat{x}(\tau_{k+1}^s)$ may fail to reach the virtual target level $\hat{s}_{k+1}$. So the algorithm is forced to take the actual inventory at τ_{k+1}^s as the target level for the $(k+1)$th cycle, i.e., setting $s(t) = s_{k+1} = x(\tau_{k+1}^s)$ for $t \in [\tau_{k+1}^s, \tau_{k+1}^e]$. Note that during the transition period $t \in [\tau_k^e, \tau_{k+1}^s]$, we use the target level $s(t) = \hat{s}_{k+1}$.

On a high level, the algorithm repeats the following three processes. First, beginning at τ_k^s, we construct the corresponding virtual production cycle, and decide τ_k^e. Second, at τ_k^e, we update the current target level from s_k to $\hat{s}_{k+1}$. Third, we decide τ_{k+1}^s based on different cases.

Algorithm 1 Data-Driven Algorithm for the Make-To-Stock Queue (DMTS)

Step 0. (Initialization.)

Set initial inventory $x(0) \in [0, \bar{s}]$. Set $s_0 = \hat{s}_0 = \tilde{s}_0 = x(0)$. Initialize the cycle counter $k = 1$, and $\tau_1^s = 0$.

Step 1. (Keeping track of virtual production cycles.)

At time $t = \tau_k^s$, the actual implemented system sets target inventory level $s_k = x(\tau_k^s)$. We construct the corresponding virtual production cycle starting with inventory level $\hat{s}_k$ based on different cases.

(i). If the actual and virtual target levels are the same, i.e., $s_k = \hat{s}_k$, then we define

$$\tau_k^e = \min\{t > \tau_k^s | \hat{x}(t) = \hat{s}_k\} = \min\{t > \tau_k^s | x(t) = s_k\}$$

where in this case $\hat{x}(t) = x(t)$. We keep producing until τ_k^e and calculate the gradient for update by

$$G_k(\hat{s}_k) = \begin{cases} \int_{\tau_k^s}^{\tau_k^e} \left[h\mathbb{1}\left[\hat{x}(t) \geq 0\right] + b\mathbb{1}\left[\hat{x}(t) < 0\right]\right] dt, & \text{if } \hat{s}_k = \lfloor \tilde{s}_k \rfloor, \\ \int_{\tau_k^s}^{\tau_k^e} \left[h\mathbb{1}\left[\hat{x}(t) > 0\right] + b\mathbb{1}\left[\hat{x}(t) \leq 0\right]\right] dt, & \text{if } \hat{s}_k = \lceil \tilde{s}_k \rceil. \end{cases}$$

Figure 4.2(a) gives an example for this case.

(ii). If, on the other hand, the actual target level is lower than the virtual target level, i.e., $s_k < \hat{s}_k$, then we still need to determine when the virtual production

cycle ends and thus we define

$$\tau_k^e = \min\{t > \tau_k^s | \hat{x}(t) = \hat{s}_k\} \tag{4.1}$$

where

$$\hat{x}(t) = \hat{s}_k + p(\tau_k^s, t - \beta_k^s) - d(\tau_k^s, t)$$

and $\beta_k^s = \min\{\beta_i > \tau_k^s\}$ is the time in which the first customer arrives after τ_k^s. Then we keep producing if $x(t) < \hat{s}_k$ and stay idle if $x(t) = \hat{s}_k$. There are two possibilities as follows.

(a) If $x(t)$ does not reach the target level $\hat{s}_k$ *twice* before τ_k^e, we calculate the gradient for update by

$$G_k(\hat{s}_k) = \begin{cases} \int_{\tau_k^s}^{\tau_k^e} \left[h\mathbb{1}\left[\hat{x}(t) \geq 0\right] + b\mathbb{1}\left[\hat{x}(t) < 0\right]\right] dt, & \text{if } \hat{s}_k = \lfloor \tilde{s}_k \rfloor, \\ \int_{\tau_k^s}^{\tau_k^e} \left[h\mathbb{1}\left[\hat{x}(t) > 0\right] + b\mathbb{1}\left[\hat{x}(t) \leq 0\right]\right] dt, & \text{if } \hat{s}_k = \lceil \tilde{s}_k \rceil. \end{cases}$$

And we keep τ_k^e unchanged as in (4.1). Figures 4.2(d) and 4.2(e) give two examples for this case, where the system in 4.2(d) does not hit $\hat{s}_{k+1}$ and the system in 4.2(e) hits $\hat{s}_{k+1}$ exactly once.

(b) If $x(t)$ reaches $\hat{s}_k$ *twice* before τ_k^e, then we reset τ_k^e to be the second time $x(t)$ reaches $\hat{s}_k$, i.e.,

$$\tau_k^e = \min\{t > \tau_k'^s | x(t) = \hat{s}_k\}, \quad \text{and} \quad \tau_k'^s = \min\{t > \tau_k^s | x(t) = \hat{s}_k\}.$$

Then we calculate the gradient for update by

$$G_k(\hat{s}_k) = \begin{cases} \int_{\tau_k'^s}^{\tau_k^e} \left[h\mathbb{1}\left[x(t) \geq 0\right] + b\mathbb{1}\left[x(t) < 0\right]\right] dt, & \text{if } \hat{s}_k = \lfloor \tilde{s}_k \rfloor, \\ \int_{\tau_k'^s}^{\tau_k^e} \left[h\mathbb{1}\left[x(t) > 0\right] + b\mathbb{1}\left[x(t) \leq 0\right]\right] dt, & \text{if } \hat{s}_k = \lceil \tilde{s}_k \rceil. \end{cases}$$

Figure 4.2(f) gives an example for this case.

Step 2. (Updating the virtual target inventory level.)

At time $t = \tau_k^e$, we update the virtual target level via a stochastic gradient descent step as follows.

$$\tilde{s}_{k+1} = \mathcal{P}_{[0,\bar{s}]}\left(\hat{s}_k - \eta_k \cdot G_k(\hat{s}_k)\right), \quad \text{where the step size} \quad \eta_k = \frac{1}{\sqrt{\sum_{i=1}^{k} \tau_i^e - \tau_i^s}}.$$

Note that the projection operator $\mathcal{P}_{[0,\bar{s}]} = \max\{0, \min\{x, \bar{s}\}\}$.

Since $\tilde{s}_{k+1}$ could be fractional, we use the following randomized rounding rule to get $\hat{s}_{k+1}$.

$$\hat{s}_{k+1} = \begin{cases} \lceil \tilde{s}_{k+1} \rceil, & \text{with probability } \tilde{s}_{k+1} - \lfloor \tilde{s}_{k+1} \rfloor, \\ \lfloor \tilde{s}_{k+1} \rfloor, & \text{with probability } 1 - (\tilde{s}_{k+1} - \lfloor \tilde{s}_{k+1} \rfloor). \end{cases}$$

Step 3. (Updating the actual implemented target inventory level.)

At time $t = \tau_k^e$, we choose different updating strategy depending on the virtual target level $\hat{s}_{k+1}$ and the mode of the facility (either producing or idling).

(i). If $x(t) = \hat{s}_k$, then the facility is idle, and we have three cases.

(a) If $x(t) = \hat{s}_{k+1}$, we stay idle and set the new cycle target level $s_{k+1} = \hat{s}_{k+1}$ and set $\tau_{k+1}^s = \tau_k^e$. Figure 4.1(a) gives an example for this case.

(b) If $x(t) > \hat{s}_{k+1}$, we stay idle and set the new cycle target level $s_{k+1} = \hat{s}_{k+1}$ and set

$$\tau_{k+1}^s = \min\{t > \tau_k^e \mid x(t) = \hat{s}_{k+1}\}.$$

Figure 4.2(a) gives an example for this case.

(c) If $x(t) < \hat{s}_{k+1}$, we keep producing, and there are two sub-cases as follows.

i. If $x(t)$ reaches $\hat{s}_{k+1}$ before any customer arrives, then we set the new

cycle target level $s_{k+1} = \hat{s}_{k+1}$ and

$$\tau^s_{k+1} = \min\{t \geq \tau^e_k \mid x(t) = \hat{s}_{k+1}\}.$$

Figure 4.2(b) gives an example for this case.

ii. If $x(t)$ does not reach $\hat{s}_{k+1}$ before any customer arrives, then we set

$$\tau^s_{k+1} = \max\{\alpha_j < \beta'_k\}, \quad \text{and} \quad \beta'_k = \min\{\beta_i > \tau^e_k\}.$$

We set the new cycle target level $s_{k+1} = x(\tau^s_{k+1})$. Figure 4.2(b) gives an example for this case.

(ii). If $x(t) < \hat{s}_k$, then the facility is producing, and we have three cases.

(a) If $x(t) = \hat{s}_{k+1}$, we keep producing. Set $\tau^s_{k+1} = \tau^e_k$ and the new target level $s_{k+1} = \hat{s}_{k+1}$.

(b) If $x(t) < \hat{s}_{k+1}$, we keep producing, and we have two sub-cases as follows.

i. If $x(t)$ reaches $\hat{s}_{k+1}$ before any customer arrives, then we set the new cycle target level $s_{k+1} = \hat{s}_{k+1}$ and set

$$\tau^s_{k+1} = \min\{t > \tau^e_k \mid x(t) = \hat{s}_{k+1}\}.$$

ii. If $x(t)$ does not reach $\hat{s}_{k+1}$ before any customer arrives, then we set

$$\tau^s_{k+1} = \max\{\alpha_j < \beta'_k\}, \quad \text{and} \quad \beta'_k = \min\{\beta_i > \tau^e_k\}.$$

We set the new cycle target level $s_{k+1} = x(\tau^s_{k+1})$.

(c) If $x(t) > \hat{s}_{k+1}$, we finish the current product first, and we have two sub-cases as follows.

i. If at the time t when the current product is finished and $x(t) \geq \hat{s}_{k+1}$, then we idle and set new cycle target level $s_{k+1} = \hat{s}_{k+1}$, and set

$$\tau^s_{k+1} = \min\{t > \tau^e_k \mid x(t) = \hat{s}_{k+1}\}.$$

ii. If at the time t when the current product is finished and $x(t) < \hat{s}_{k+1}$, then we keep producing and apply the previous case (b).

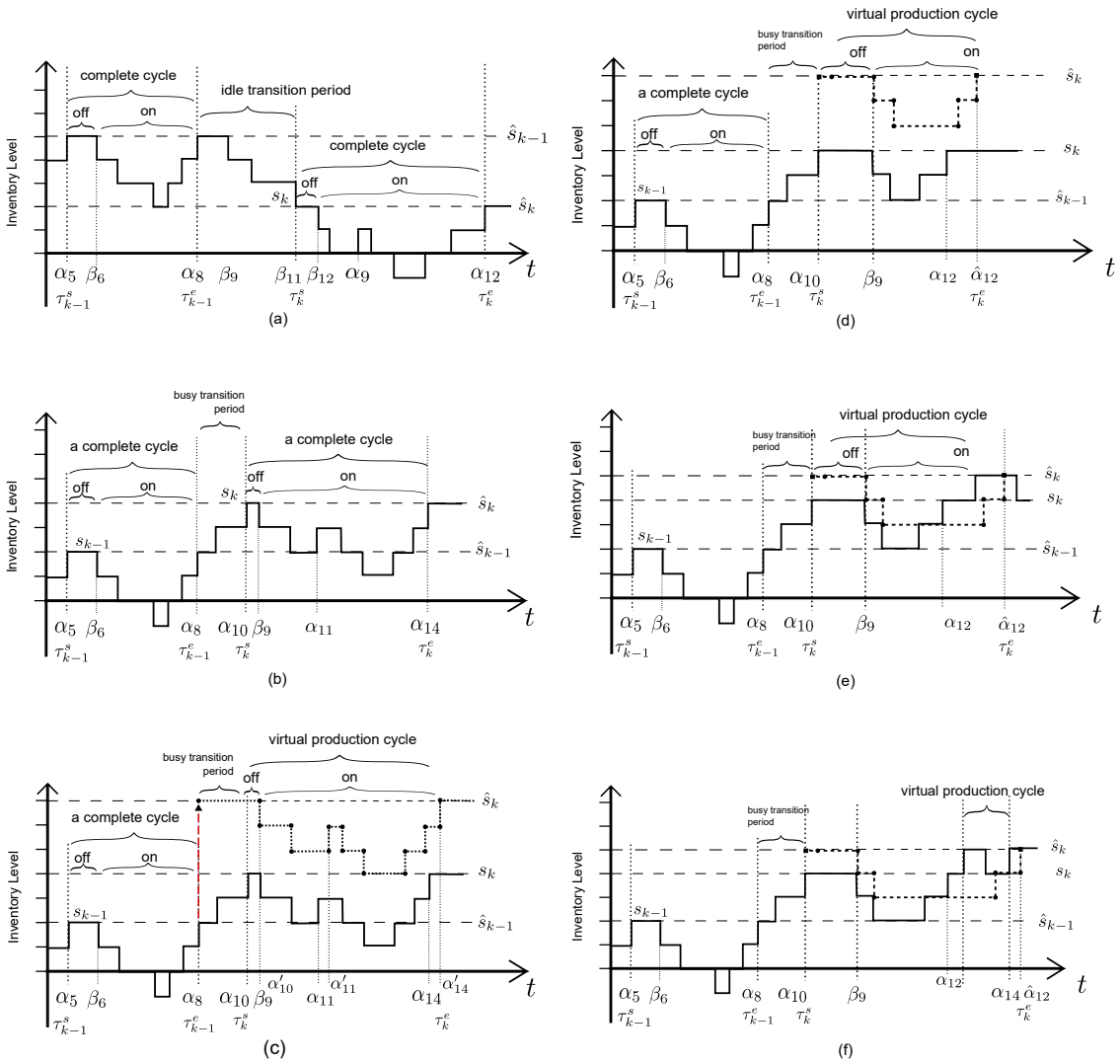

Figure 4.2: Illustration of the dynamics of our policy

4.4 Performance Analysis of the DMTS Algorithm

We measure the performance of the DMTS algorithm by *cumulative regret* or simply *regret*, which is defined as the difference between the cost incurred by our algorithm and the clairvoyant optimal cost (had the arrival rate and the production time distribution are both known *a priori*). That is, for any $T \geq 1$,

$$\mathcal{R}_T = \mathbb{E}\left[\int_0^T (hx(t)^+ + bx(t)^-)dt\right] - \mathbb{E}\left[\sum_{k=1}^{K} \mathcal{G}(s^*)\right],$$

where $x(t)$ is the inventory level at time t under the DMTS algorithm, and s^* is the clairvoyant optimal target level. Theorem 4.2 below states the main result of this paper.

Theorem 4.2. *For a make-to-stock queue with unknown arrival rate and production time distribution a priori, the cumulative regret $\mathcal{R}_T$ of the DMTS algorithm is bounded by $O(\sqrt{T})$. In other words, the average T-period running cost converges to the clairvoyant optimal cost at $O(1/\sqrt{T})$, which is provably tight.*

Proof. Let K be the total number of (τ^s, τ^e) pairs constructed by the algorithm during time $[0, T]$, including possibly the last incomplete production cycle. If $\tau_K^e > T$, then we truncate that cycle and let $\tau_K^e = T$. We also truncate the last transition period $[\tau_K^e, \tau_{K+1}^s]$ by setting $\tau_{K+1}^s = T$ and set $s_{K+1} = s_K$. We can then decompose the

cumulative regret as follows.

$$\begin{aligned}
\mathcal{R}_T &= \mathbb{E}\left[\int_0^T (hx(t)^+ + bx(t)^-)dt\right] - \mathbb{E}\left[\sum_{k=1}^K \mathcal{G}(s^*)\right] \\
&= \mathbb{E}\left[\sum_{k=1}^K \int_{\tau_k^s}^{\tau_k^e} (hx(t)^+ + bx(t)^-)dt + \sum_{k=1}^K \int_{\tau_k^e}^{\tau_{k+1}^s} (hx(t)^+ + bx(t)^-)dt\right] \\
&\quad -\mathbb{E}\left[\sum_{k=1}^K \mathcal{G}(s^*)\right] \\
&= \mathbb{E}\left[\sum_{k=1}^K \mathcal{G}(\hat{s}_k) - \sum_{k=1}^K \mathcal{G}(s^*)\right] + \mathbb{E}\left[\sum_{k=1}^K \int_{\tau_k^s}^{\tau_k^e} (hx(t)^+ + bx(t)^-)dt - \sum_{k=1}^K \mathcal{G}(\hat{s}_k)\right] \\
&\quad +\mathbb{E}\left[\sum_{k=1}^K \int_{\tau_k^e}^{\tau_{k+1}^s} (hx(t)^+ + bx(t)^-)dt\right], \qquad (4.2)
\end{aligned}$$

where $\mathcal{G}(\hat{s}_k)$ and $\mathcal{G}(s^*)$ are the costs for a standard production cycle using s_k and s^* as target levels, respectively. Note that $\mathcal{G}(\hat{s}_k)$ represents the cost for the virtual production cycle we constructed. In the third equality, the first part is the cost difference between the virtual production cycles and the optimal system, the second part is the cost difference between the actual implemented system and the corresponding virtual production cycles, and the third part is the cost for the transition period in the actual implemented system.

The proof of Theorem is a direct consequence of Propositions 4.5, 4.9, and 4.10 (shown below) which give bounds for the three parts in (4.2). Proposition 4.5 utilizes preliminary results Lemma 4.3 and 4.4. Proposition 4.9 utilizes preliminary results Lemma 4.6, 4.7 and 4.8. Moreover, the convergence rate is tight (or optimal) due to Proposition 4.11 (also shown below). □

Lemma 4.3. *The cost of the standard production cycle is convex in the target level with probability one.*

Proof. Let $\mathcal{G}(s)$ denote the cost of a standard production cycle under an arbitrary target level s. At the beginning of the cycle, the inventory level is s and the facility is idle. The facility starts producing when the first customer arrives. We can write the cost as follows.

$$\begin{aligned}
\mathcal{G}(s) &= \int_{\tau^s}^{\tau^e} \left[h\left(s \wedge \left(x(\tau^s) + P(\tau^s, t) - D(\tau^s, t)\right)\right)^+ \right. \\
&\qquad \left. -b\left(s \wedge \left(D(\tau^s, t) - x(\tau^s) - P(\tau^s, t)\right)\right)^+\right] dt \\
&= hs(\beta^s - \tau^s) + \int_{\beta^s}^{\tau^e} \left[h\left(s \wedge \left(x(\beta^s) + P(\beta^s, t) - D(\beta^s, t)\right)\right)^+ \right. \\
&\qquad \left. -b\left(s \wedge \left(D(\beta^s, t) - x(\beta^s) - P(\beta^s, t)\right)\right)^+\right] dt \\
&= hs(\beta^s - \tau^s) + \int_{\beta^s}^{\tau^e} \left[h\left(x(\beta^s) + P(\beta^s, t) - D(\beta^s, t)\right)^+ \right. \\
&\qquad \left. -b\left(D(\beta^s, t) - x(\beta^s) - P(\beta^s, t)\right)^+\right] dt, \\
&= hs(\beta^s - \tau^s) + \int_{\beta^s}^{\tau^e} \left[h\left(s - 1 + P(\beta^s, t) - D(\beta^s, t)\right)^+ \right. \\
&\qquad \left. -b\left(D(\beta^s, t) - s + 1 - P(\beta^s, t)\right)^+\right] dt,
\end{aligned}$$

where τ^s and τ^e are the production cycle starting and ending times, respectively. Note that $P(t_1, t_2)$ is the number of units produced during $[t_1, t_2]$, and $D(t_1, t_2)$ is the number of customers arrived during $[t_1, t_2]$, and β^s is the arrival time of the first customer after τ^s. The second equality holds because the facility is always idle before the first customer arrives, and the target level is always non-negative, i.e., $s \geq 0$. The second equality holds because $s \geq x(\beta^s) + P(\beta^s, t) - D(\beta^s, t)$ for all $t \in [\beta^s, \tau^e]$ due to the construction of the policy. The last equality holds because $x(\beta^s) = x(\tau^s) - 1 = s - 1$.

Taking the first derivative of $\mathcal{G}(s)$ with respect to s, we have

$$\frac{\partial}{\partial s}\mathcal{G}(s) = h(\beta^s - \tau^s) + \int_{\beta^s}^{\tau^e} [h\mathbb{1}[x(t) \geq 0] - b\mathbb{1}[x(t) < 0]]\, dt,$$

and

$$x(t) = s - 1 + P(\beta^s, t) - D(\beta^s, t).$$

It is clear that $\frac{\partial}{\partial s}\mathcal{G}(s+\delta) \geq \frac{\partial}{\partial s}\mathcal{G}(s)$ for any $\delta > 0$ and any $s > 0$. Thus, $\mathcal{G}(s)$ is convex. In addition, we use the Fundamental Theorem of Calculus to take derivative and have

$$\nabla \mathbb{E}\left[\mathcal{G}(\hat{s})\right] = h\mathbb{E}[\beta] + \mathbb{E}\left[\int_{\beta^s}^{\tau^e} [h\mathbb{1}[\hat{x}(t) \geq 0] + b\mathbb{1}[\hat{x}(t) < 0]]\, dt\right]$$

where $\mathbb{E}[\beta] = \mathbb{E}[\beta^s - \tau^s]$ is the expected customer inter-arrival time.

□

Lemma 4.4. *For any $k \geq 1$, $G_k(\hat{s}_k)$ is an unbiased estimator of the expected cost gradient of the kth production cycle, i.e., $\mathbb{E}[G_k(\hat{s}_k)] = \nabla \mathbb{E}\left[\mathcal{G}(\hat{s}_k)\right]$. Also, $G_k(\hat{s}_k)$ has a bounded second moment, i.e., $\mathbb{E}[(G_k(\hat{s}_k))^2] < \infty$.*

Proof. Since the system can observe both the inter-arrival times and production times, the algorithm can construct $\hat{x}(t)$ based on $\hat{s}_k$ and $x(t)$. However, since inventory is discrete, we incorporate a probabilistic rounding rule while calculating $\hat{s}_k$. Then we have the following two cases when calculating $G_k(\hat{s}_k)$.

If $\hat{s}_k = \lfloor \tilde{s}_k \rfloor$, then when the virtual inventory level $\hat{x}(t) = 0$, the system would have strictly positive inventory if $\tilde{s}_k$ were implemented as the target level. Therefore, to have any backlog in the virtual system, the virtual inventory level needs to be strictly negative, and therefore we use $b\mathbb{1}\left[\hat{x}(t) < 0\right]$ to indicate the backlogging cost. Similarly, if $\hat{s}_k = \lceil \tilde{s}_k \rceil$, then when the virtual inventory level $\hat{x}(t) = 0$, the system

would have strictly negative inventory if $\tilde{s}_k$ were implemented as the target level. Therefore, to have any leftover inventory in the virtual system, the virtual inventory level needs to be strictly positive, and therefore we use $h\mathbb{1}\left[\hat{x}(t) > 0\right]$ to indicate the holding cost.

Based on the algorithm, if $s_k < \hat{s}_k$ and $x(t)$ does not reach $\hat{s}_k$ twice before τ_k^e, then if $\hat{s}_k = \lfloor \tilde{s}_k \rfloor$, we have

$$\begin{aligned}
\mathbb{E}[G_k(\hat{s}_k)] &= \mathbb{E}\left[\int_{\tau_k^s}^{\tau_k^e} \left[h\mathbb{1}\left[x(t) \geq 0\right] + b\mathbb{1}\left[x < 0\right]\right] dt\right] \\
&= \mathbb{E}\left[h(\beta_k^s - \tau_k^s) + \int_{\beta_k^s}^{\tau_k^e} \left[h\mathbb{1}\left[\hat{x}(t) \geq 0\right] + b\mathbb{1}\left[\hat{x}(t) < 0\right]\right] dt\right] \\
&= h\mathbb{E}[\beta] + \mathbb{E}\left[\int_{\beta_k^s}^{\tau_k^e} \left[h\mathbb{1}[\hat{x}(t) \geq 0] + b\mathbb{1}[\hat{x}(t) < 0]\right] dt\right] = \nabla\mathbb{E}\left[\mathcal{G}(\hat{s}_k)\right].
\end{aligned}$$

Note that $\hat{x}(t) = \hat{s}_k + p(\tau_k^s, t - \beta_k^s) - d(\tau_k^s, t)$ has the same dynamics as $x(t) = s_k + p(\tau_k^s, t - \beta_k^s) - d(\tau_k^s, t)$. The same argument applies to the case where $\hat{s}_k = \lceil \tilde{s}_k \rceil$, and also the case where $s_k = \hat{s}_k$.

If $s_k < \hat{s}_k$ and $x(t)$ reaches $\hat{s}_k$ twice before τ_k^e, then $G_k(\hat{s}_k)$ is calculated using the actual inventory level $x(t)$ within $[\tau_k'^s, \tau_k^e]$. Because the cycle length is independent of the target level and only dependent on the inter-arrival and the production times, it is clear that $\mathbb{E}[G_k(\hat{s}_k)] = \nabla\mathbb{E}\left[\mathcal{G}(\hat{s}_k)\right]$.

Let B be the busy period of a standard production cycle. Then its second moment is given by

$$\mathbb{E}[B^2] = \frac{\mathbb{E}[U^2]}{(1-\rho)^2} = \frac{\mathbb{E}[U^2]}{(1-\lambda\mathbb{E}[U])^2} < \infty,$$

and the second moment of the idle period is

$$\mathbb{E}[\beta^2] = \frac{2}{\lambda^2} < \infty,$$

where β is the inter-arrival time of a single customer. Note that $\rho = \lambda\mathbb{E}[U] < 1$. Therefore, we have

$$\mathbb{E}[(G_k(\hat{s}_k))^2] \leq \mathbb{E}\left[(h \vee b)^2(\beta + B)^2\right] < \infty.$$

This completes the proof. □

Proposition 4.5. *For any $K \geq 1$, there exists a constant A_1 such that*

$$\mathbb{E}\left[\sum_{k=1}^{K} \mathcal{G}(\hat{s}_k) - \mathcal{G}(s^*)\right] \leq A_1\sqrt{T}.$$

Proof. Since $\mathcal{G}(s)$ is convex almost surely by Lemma 4.3, and also $\mathbb{E}\left[G_k(\hat{s}_k)\right]$ is an unbiased estimator of $\nabla\mathbb{E}\left[\mathcal{G}(\hat{s}_k)\right]$ by Lemma 4.4, we can bound the difference by

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{K}\left(\mathcal{G}(\hat{s}_k) - \mathcal{G}(s^*)\right)\right] &\leq \mathbb{E}\left[\sum_{k=1}^{K} \nabla\mathcal{G}(\hat{s}_k)(\hat{s}_k - s^*)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\sum_{k=1}^{K} \nabla\mathcal{G}(\hat{s}_k)(\hat{s}_k - s^*)\right] \mid \hat{s}_k\right] \\
&= \mathbb{E}\left[\sum_{k=1}^{K} \mathbb{E}\left[\nabla\mathcal{G}(\hat{s}_k)(\hat{s}_k - s^*)\right] \mid \hat{s}_k\right] \\
&= \mathbb{E}\left[\sum_{k=1}^{K} \nabla\mathbb{E}\left[\mathcal{G}(\hat{s}_k)\right](\hat{s}_k - s^*) \mid \hat{s}_k\right] \\
&= \mathbb{E}\left[\sum_{k=1}^{K} \mathbb{E}\left[G_k(\hat{s}_k)\right](\hat{s}_k - s^*)\right].
\end{aligned} \tag{4.3}$$

By the definition of the projection operator $\mathcal{P}_{[0,\bar{s}]}$, we have that $(\mathcal{P}_{[0,\bar{s}]}(\hat{s}_k - s^*))^2 \leq (\hat{s}_k - s^*)^2$, and so

$$\begin{aligned}
(\tilde{s}_{k+1} - s^*)^2 &= (\mathcal{P}_{[0,\bar{s}]}(\hat{s}_k - \eta_k G_k(\hat{s}_k) - s^*))^2 \\
&\leq (\hat{s}_k - \eta_k G_k(\hat{s}_k) - s^*)^2 \\
&= (\hat{s}_k - s^*)^2 + \eta_k^2 G_k(\hat{s}_k)^2 - 2\eta_k G_k(\hat{s}_k)(\hat{s}_k - s^*).
\end{aligned}$$

After re-arranging the terms, we have

$$G_k(\hat{s}_k)(\hat{s}_k - s^*) \leq \frac{1}{2\eta_k}\left((\hat{s}_k - s^*)^2 - (\tilde{s}_{k+1} - s^*)^2\right) + \frac{1}{2}\eta_k \left(G_k(\hat{s}_k)\right)^2. \tag{4.4}$$

Now, combining (4.3) and (4.4), we have

$$\begin{aligned}
& \mathbb{E}\left[\sum_{k=1}^{K}\left(\mathcal{G}(\hat{s}_k) - \mathcal{G}(s^*)\right)\right] \leq \sum_{k=1}^{K}\mathbb{E}\left[G_k(\hat{s}_k)(\hat{s}_k - s^*)\right] \\
\leq\ & \sum_{k=1}^{K}\left(\frac{1}{2\mathbb{E}[\eta_k]}\left(\mathbb{E}\left(\hat{s}_k - s^*\right)^2 - \mathbb{E}\left(\tilde{s}_{k+1} - s^*\right)^2\right) + \frac{1}{2}\mathbb{E}\left[\eta_k \left(G_k(\hat{s}_k)\right)^2\right]\right) \\
=\ & \frac{1}{2\mathbb{E}[\eta_1]}\mathbb{E}\left(\hat{s}_1 - s^*\right)^2 - \frac{1}{2\mathbb{E}[\eta_k]}\mathbb{E}\left(\tilde{s}_{k+1} - s^*\right)^2 + \\
& \frac{1}{2}\sum_{k=2}^{K}\left(\frac{1}{\mathbb{E}[\eta_k]} - \frac{1}{\mathbb{E}[\eta_{k-1}]}\right)\mathbb{E}\left(\hat{s}_k - s^*\right)^2 + \sum_{k=1}^{K}\frac{\mathbb{E}\left[\eta_k \left(G_k(\hat{s}_k)\right)^2\right]}{2} \\
\leq\ & 2\bar{s}^2\left(\frac{1}{2\mathbb{E}[\eta_1]} + \frac{1}{2}\sum_{k=2}^{K}\left(\frac{1}{\mathbb{E}[\eta_k]} - \frac{1}{\mathbb{E}[\eta_{k-1}]}\right)\right) + \frac{\mathbb{E}[\left(G_k(\hat{s}_k)\right)^2]}{2}\sum_{k=1}^{K}\mathbb{E}[\eta_k] \\
=\ & \frac{\bar{s}^2}{\mathbb{E}[\eta_K]} + \frac{\mathbb{E}[\left(G_k(\hat{s}_k)\right)^2]}{2}\sum_{k=1}^{K}\mathbb{E}[\eta_k] \leq A_1\sqrt{T},
\end{aligned} \tag{4.5}$$

where the last inequality holds due to the fact that $\mathbb{E}[\left(G_k(\hat{s}_k)\right)^2]$ is finite by Lemma 4.4 and

$$\sum_{k=1}^{K}\mathbb{E}[\eta_k] = \sum_{k=1}^{K}\mathbb{E}\left[1/\sqrt{\sum_{i=1}^{k}(\tau_k^e - \tau_k^s)}\right] \leq \int_{t=1}^{T} 1/\sqrt{t} \leq 2\sqrt{T}.$$

This completes the proof. □

Lemma 4.6. *For any $k \geq 1$,*

$$\mathbb{E}\left[\int_{\tau_k^s}^{\tau_k^e}(hx(t)^+ + bx(t)^-)dt - \mathcal{G}(\hat{s}_k)\right] \leq \mathbb{E}[L_k](h \vee b)(\hat{s}_k - s_k), \qquad \textit{where} \quad L_k = \tau_k^e - \tau_k^s.$$

Proof. There are two cases. If at $t = \tau_k^s$, the facility is idle, then based on the DMTS algorithm, we know that the actual inventory level must reach the virtual target level,

i.e., $x(\tau_k^s) = s_k = \hat{s}_k$. Then we have

$$\int_{\tau_k^s}^{\tau_k^e} (hx(t)^+ + bx(t)^-)dt = \mathcal{G}(\hat{s}_k).$$

Otherwise, if at $t = \tau_k^s$, the facility is not idle, then we need to the construct corresponding virtual production cycle. Therefore, applying the system dynamics of both actual implemented and virtual systems, we have

$$\begin{aligned}
& \int_{\tau_k^s}^{\tau_k^e} (hx(t)^+ + bx(t)^-)dt - \mathcal{G}(\hat{s}_k) \\
= & \int_{\tau_k^s}^{\tau_k^e} \left[h(x(t) \geq 0) + b(x(t) < 0)\right] dt - \int_{\tau_k^s}^{\tau_k^e} \left[h(\hat{x}(t) \geq 0) + b(\hat{x}(t) < 0)\right] dt \\
= & \int_{\tau_k^s}^{\tau_k^e} \left[h\left(s_k + p(\tau_k^s, t) - d(\tau_k^s, t)\right)^+ + b\left(d(\tau_k^s, t) - s_k - p(\tau_k^s, t)\right)^+\right] dt \\
& - \int_{\tau_k^s}^{\tau_k^e} \left[h\left(\hat{s}_k + p(\tau_k^s, t - \beta_k^s) - d(\tau_k^s, t)\right)^+ + b\left(d(\tau_k^s, t) - \hat{s}_k - p(\tau_k^s, t - \beta_k^s)\right)^+\right] dt \\
= & \int_{\tau_k^s}^{\tau_k^e} -h\left[\hat{s}_k + p(\tau_k^s, t - \beta_k^s) - \max\{s_k + p(\tau_k^s, t), d(\tau_k^s, t)\}\right]^+ dt \\
& + \int_{\tau_k^s}^{\tau_k^e} b\left[\min\{\hat{s}_k - p(\tau_k^s, t + \beta_k^s), d(\tau_k^s, t)\} - s_k + p(\tau_k^s, t)\right]^+ dt \\
\leq & \int_{\tau_k^s}^{\tau_k^e} (h \vee b)(\hat{s}_k + p(\tau_k^s, t - \beta_k^s) - s_k - p(\tau_k^s, t))^+ dt \\
\leq & \int_{\tau_k^s}^{\tau_k^e} (h \vee b)(\hat{s}_k - s_k) dt,
\end{aligned}$$

where the second equality is derived by applying the system dynamics of $x(t)$ and

$\hat{x}(t)$, and the third equality holds due to the fact that $s_k \leq \hat{s}_k$ for all k and when $x \leq \hat{x}$, we have

$$(x-d)^+-(\hat{x}-d)^+ = -(\hat{x}-\max\{x,d\})^+ \quad \text{and} \quad (d-x)^+-(d-\hat{x})^+ = (\min\{\hat{x},d\}-x)^+.$$

The last inequality is due to the fact that $p(\tau_k^s, t-\beta_k^s) \leq p(\tau_k^s, t)$ for any k and t. Since the length of the production cycle does not depend on $\hat{s}_k$ and s_k, we have

$$\begin{aligned} \mathbb{E}\left[\int_{\tau_k^s}^{\tau_k^e} (hx(t)^+ + bx(t)^-)dt - \mathcal{G}(\hat{s}_k)\right] &\leq \mathbb{E}\left[\int_{\tau_k^s}^{\tau_k^e} (h \vee b)(\hat{s}_k - s_k)dt\right] \\ &= \mathbb{E}[L_k](h \vee b)(\hat{s}_k - s_k). \end{aligned}$$

This completes the proof. □

Lemma 4.7. *Define Z_k as a stochastic process. $Z_0 = 0$, and for $k \geq 0$,*

$$Z_{k+1} = \left[Z_k + \frac{\upsilon_k}{\sqrt{\sum_{i=1}^k L_i}} - \omega_k\right]^+ \tag{4.6}$$

where random variable $\upsilon_k = (h \vee b)L_k$, and $\omega_k = P(\tau_k^e, \tau_{k+1}^s) - |\tilde{s}_{k+1} - \hat{s}_{k+1}|$.

Then we have for any $K \geq 1$,

$$\mathbb{E}\left[\sum_{k=1}^K (\hat{s}_k - s_k)\right] \leq \mathbb{E}\left[\sum_{k=1}^K Z_k\right]$$

Proof. When $\hat{s}_{k+1} \leq s_k$, according to the algorithm, we know that the facility must be idle at $t = \tau_{k+1}^s$. Therefore, we have $\hat{s}_{k+1} - s_{k+1} = 0 \leq Z_{k+1}$. When $\hat{s}_{k+1} > s_k$, then the algorithm keeps track of a transition period, and we have $s_{k+1} = s_k + P(\tau_k^e, \tau_{k+1}^s)$ where $P(\tau_k^e, \tau_{k+1}^s)$ is the number of products finished during $[\tau_k^e, \tau_{k+1}^s]$. Therefore, we

can write

$$\begin{aligned}
\hat{s}_{k+1} - s_{k+1} &\leq \lceil \mathcal{P}_{[0,\bar{s}]} \left(\hat{s}_k - \eta_k \cdot G_k(\hat{s}_k) \right) \rceil - s_{k+1} \\
&\leq |\mathcal{P}_{[0,\bar{s}]} \left(\hat{s}_k - \eta_k \cdot G_k(\hat{s}_k) \right)| + |\tilde{s}_{k+1} - \hat{s}_{k+1}| - s_{k+1} \\
&\leq |\hat{s}_k - \eta_k \cdot G_k(\hat{s}_k)| + |\tilde{s}_{k+1} - \hat{s}_{k+1}| - s_k - P(\tau_k^e, \tau_{k+1}^s) \\
&\leq |\hat{s}_k - s_k - \eta_k \cdot G_k(\hat{s}_k)| + |\tilde{s}_{k+1} - \hat{s}_{k+1}| - P(\tau_k^e, \tau_{k+1}^s) \\
&\leq |\hat{s}_k - s_k| + |\eta_k \cdot G_k(\hat{s}_k)| + |\tilde{s}_{k+1} - \hat{s}_{k+1}| - P(\tau_k^e, \tau_{k+1}^s) \\
&\leq (\hat{s}_k - s_k) + \eta_k (h \vee b) L_k - (P(\tau_k^e, \tau_{k+1}^s) - |\tilde{s}_{k+1} - \hat{s}_{k+1}|) \\
&= (\hat{s}_k - s_k) + \frac{1}{\sqrt{\sum_i^k L_i}} (h \vee b) L_k - (P(\tau_k^e, \tau_{k+1}^s) - |\tilde{s}_{k+1} - \hat{s}_{k+1}|).
\end{aligned}$$

The first inequality holds because $\hat{s}_{k+1}$ is derived from probabilistic rounding on $\tilde{s}_{k+1}$. The third inequality is due to the convexity property of $\mathcal{P}_{[0,\bar{s}]}$. The fifth inequality holds because of the triangular inequality. The last inequality holds because $|G_k(\hat{s}_k)| \leq (h \vee b) L_k$ where $L_k = \tau_k^e - \tau_k^s$. In addition, we know that $\hat{s}_0 - s_0 = 0$. Then, by the definition of Z_{k+1}, it is clear that $\hat{s}_{k+1} - s_{k+1} \leq Z_{k+1}$. Summing up both sides of the inequality completes the proof.

□

Define a $GI/G/1$ queue having the waiting time of the kth customer $(W_k \mid k \geq 0)$ by the Lindley's equation:

$$W_{k+1} = [W_k + \upsilon_k - \omega_k]^+, \quad \text{where} \quad W_0 = 0 \tag{4.7}$$

where υ_k denote the inter-arrival time between the kth and kth customers and υ_k denote the service time of the kth customer. Let $\varphi_0 = 0$, $\varphi_m = \inf\{k \geq 1 : W_k = 0\}$ for any $m \geq 1$. Then $\mathcal{B}_m = \varphi_m - \varphi_{m-1}$ denote the number of customer served during the mth busy cycle, where the busy cycle is defined as the time period between an

arrival which finds the system empty until another arrival which finds the system empty again.

Lemma 4.8. *For any period $K \geq 1$, we have*

$$\mathbb{E}\left[\sum_{k=1}^{K} Z_k\right] \leq 2(h \vee b)\sqrt{T}\mathbb{E}[\mathcal{B}_1].$$

Proof. Based on the definition of Z_k, υ_k and ω_k are independent and identically distributed random variables which only depends on the distribution of R and U. The stochastic process of W_k scales the service of Z_k by $1/\sqrt{\sum_{i=1}^{k} L_i}$ in each period k. Since the system utilization factor $\rho < 1$, we have $\mathbb{E}[P(\tau_k^e, \tau_{k+1}^s)] \geq 1$ if $\hat{s}_{k+1} > s_k$. Then we have $\mathbb{E}[\omega_k] = \mathbb{E}[P(\tau_k^e, \tau_{k+1}^s)] - |\tilde{s}_{k+1} - \hat{s}_{k+1}| > 0$. Therefore, the stochastic process W_k can be forced to be stable by having $\mathbb{E}[\upsilon_k]/\mathbb{E}[\omega_k] \leq 1$.

For each $k \geq 1$, let the random variable $n(k)$ denote the index such that the $n(k)$th busy cycle contains customer k. It is well-know that in a $GI/G/1$ queue, if the system is stable, then $\mathcal{B}_m$ is i.i.d, i.e.,

$$\mathbb{E}[\mathcal{B}_{n(k)}] = \mathbb{E}[\mathcal{B}_m] = \mathbb{E}[\mathcal{B}_1].$$

Then we rewrite Z_k in respect of U_m as follows,

$$\begin{aligned} Z_k &\leq \sum_{k'=1}^{k} \left(\frac{\sqrt{\upsilon_{k'}}}{\sqrt{\sum_{i=1}^{k'} L_i}} - \omega_{k'} \right) \mathbb{1}\left[n(k') = n(k)\right]] \\ &\leq \sum_{k'=1}^{k} \left(\frac{\sqrt{\upsilon_{k'}}}{\sqrt{\sum_{i=1}^{k'} L_i}} \right) \mathbb{1}\left[n(k') = n(k)\right]. \end{aligned} \tag{4.8}$$

The first inequality holds because the stochastic process W_k dominates Z_k and when $W_k = 0$, $Z_k = 0$. $\mathbb{1}\left[n(k') = n(k)\right]$ states that customer k' and customer k are in the same busy cycle.

Then we can bound the summation of Z_k by

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{K} Z_k\right] &\leq \mathbb{E}\left[\sum_{k=1}^{K}\sum_{k'=1}^{k} \frac{\sqrt{\upsilon_{k'}}}{\sqrt{\sum_{i=1}^{k'} L_i}} \mathbb{1}\left[n(k') = n(k)\right]\right] \\
&\leq \mathbb{E}\left[\sum_{k=1}^{K}\sum_{k'=1}^{K} \frac{(h \vee b) L_{k'}}{\sqrt{\sum_{i=1}^{k'} L_i}} \mathbb{1}\left[n(k') = n(k)\right]\right] \\
&\leq \mathbb{E}\left[\sum_{k'=1}^{K} \frac{(h \vee b) L_{k'}}{\sqrt{\sum_{i=1}^{k'} L_i}} \sum_{k=1}^{K} \mathbb{1}\left[n(k') = n(k)\right]\right] \\
&= \mathbb{E}\left[\sum_{k'=1}^{K} \frac{(h \vee b) L_{k'}}{\sqrt{\sum_{i=1}^{k'} L_i}} \mathcal{B}_{n(k')}\right] \\
&\leq \mathbb{E}(h \vee b)\mathbb{E}\left[\int_{t=0}^{T} \frac{1}{\sqrt{t}} dt\right] \mathbb{E}[\mathcal{B}_{n(k')}] \\
&\leq 2(h \vee b)\sqrt{T}\mathbb{E}[\mathcal{B}_1].
\end{aligned}$$

The last two inequalities hold because

$$\sum_{k'=1}^{K} \frac{L_{k'}}{\sqrt{\sum_{i=1}^{k'} L_i}} \leq \int_{t=0}^{T} \frac{1}{\sqrt{t}} dt \leq 2\sqrt{T}, \quad \text{where} \quad T = \sum_{k'=1}^{K} L'_k.$$

And since $L_{k'}$ relates with $\upsilon_{k'}$ which denotes the inter-arrival time between the k'th customer and the $k'+1$th customer, $L_{k'}$ is independent of the busy cycle containing the k'th customer, and thus independent of $\mathcal{B}_{n(k')}$.

□

Proposition 4.9. *For any $K \geq 1$, there exists a constant A_2 such that*

$$\mathbb{E}\left[\sum_{k=1}^{K} \int_{\tau_k^s}^{\tau_k^e} (hx(t)^+ + bx(t)^-)dt - \mathcal{G}(\hat{s}_k)\right] \leq A_2\sqrt{T}.$$

Proof. Combining Lemmas 4.6, 4.7 and 4.8, we have

$$\begin{aligned}\mathbb{E}\left[\sum_{k=1}^{K}\mathcal{G}(s_k)-\sum_{k=1}^{K}\mathcal{G}(\hat{s}_k)\right] &\leq \mathbb{E}\left[\sum_{k=1}^{K}\mathbb{E}[L_k](h\vee b)(\hat{s}_k-s_k)\right]\\ &\leq (h\vee b)\mathbb{E}[L_1]\mathbb{E}\left[\sum_{k=1}^{K}Z_k\right]\\ &\leq 2(h\vee b)^2\mathbb{E}[L_1]\mathbb{E}[\mathcal{B}_1]\sqrt{T}.\end{aligned}$$

It has been shown by *Loulou* (1978) that the expected number of customer served by the first busy cycle $\mathbb{E}[\mathcal{B}_1]$ for a $GI/G/1$ queue is bounded by a constant involving up to the third moment of $\upsilon-\omega$ (the difference between inter-arrival and service times). Moreover, $\mathbb{E}[L_1]=\mathbb{E}[\beta+B]$ is shown to be finite in Lemma 4.4. This completes the proof for Proposition 4.9. □

Proposition 4.10. *For any $K\geq 1$, there exists a constant A_3 such that*

$$\mathbb{E}\left[\sum_{k=1}^{K}\int_{\tau_k^e}^{\tau_{k+1}^s}(hx(t)^+ + bx(t)^-)dt\right]\leq A_3\sqrt{T}$$

Proof. We can write

$$\begin{aligned}&\mathbb{E}\left[\sum_{k=1}^{K}\int_{\tau_k^e}^{\tau_{k+1}^s}(hx(t)^+ + bx(t)^-)dt\right]\\ \leq\ & h\bar{s}\mathbb{E}\left[\sum_{k=1}^{K}(s_k-s_{k+1})^+\cdot U+\sum_{k=1}^{K}(s_{k+1}-s_k)^+\cdot R\right]\\ \leq\ & h\bar{s}\mathbb{E}\left[\sum_{k=1}^{K}|s_k-s_{k+1}|\right]\mathbb{E}\left[R+U\right]\leq A_3\sqrt{T}.\end{aligned}$$

The first inequality follows from the fact that if $s_k<s_{k+1}$, then it would take $(s_k-s_{k+1})U$ time for the system to bring the inventory level from s_k up to s_{k+1} where U is the production time. Similarly, if $s_k>s_{k+1}$, then it would take $(s_{k+1}-s_k)R$ time for

the system to bring the inventory level from s_k down to s_{k+1} where R is the customer inter-arrival time. Note that $\bar{s}$ is the upper bound on the target level s_k. Since $s_k \geq 0$ for all k, the transition period will not incur backlogging cost, and therefore we upper bound the system by the maximum holding cost. The second inequality holds because s_k is independent with R and U. The last inequality is derived from the fact that

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{K}|s_{k+1}-s_k|\right] &\leq \mathbb{E}\left[\sum_{k=1}^{K}|s_{k+1}-\hat{s}_k|+|\hat{s}_k-s_k|\right] \\
&\leq \mathbb{E}\left[\sum_{k=1}^{K}|\hat{s}_{k+1}-\hat{s}_k|\right]+\mathbb{E}\left[\sum_{k=1}^{K}(\hat{s}_k-s_k)\right] \\
&\leq A_6\sqrt{T},
\end{aligned}$$

where the first and second inequalities hold because $s_k \leq \hat{s}_k$ for all k, and the last inequality holds because

$$\mathbb{E}\left[\sum_{k=1}^{K}|\hat{s}_{k+1}-\hat{s}_k|\right] \leq \mathbb{E}\left[\sum_{k=1}^{K}|\eta_k G_k(\hat{s}_k)|\right] \leq A_4\sqrt{T}.$$

due to our updating rule and the fact that $G_k(\hat{s}_k)$ is bounded by Lemma 4.4. Finally, invoking Lemmas 4.7 and 4.8,

$$\mathbb{E}\left[\sum_{k=1}^{K}(\hat{s}_k-s_k)\right] \leq A_5\sqrt{T}.$$

This completes the proof. □

Proposition 4.11. *The lower bound of any learning algorithm is* $\Omega(\sqrt{T})$ *for* $T > 4$.

Proof. Consider a make-to-stock system where the customers arrive as a Poisson process with rate λ. The system production time of one product follows an exponential distribution with rate μ. The customer is backlogged when there is no inventory. The system incurs a holding cost of h per product per unit time and waiting cost of b per

customer per unit time. At the beginning, the inventory level is zero, and a policy chooses a target stock level y such that when the system has inventory level $x < y$, the facility keeps producing, and when the system inventory level reaches y, the facility stops. The expected cost over time T can be written as $\int_0^T hx(t)^+ + bx(t)^+ dt$, where $x(t)^+$ is the number of positive inventory at time t and $x(t)^-$ is the number of backorders (negative inventory) at time t.

Consider a pair of production rates, μ_1 and μ_2, where

$$\mu_1 = \frac{6\sqrt{T}}{3\sqrt{T}+2}, \qquad \mu_2 = \frac{6\sqrt{T}}{3\sqrt{T}-2}.$$

Consider $h = b = 1$ and $\lambda = 1$, the queue length follows the following pair of distributions:

$$F_Q^a(k) = \begin{cases} \frac{1}{2} + \frac{1}{3\sqrt{T}} & \text{for } k = 0 \\ \frac{3}{4} - \frac{1}{3\sqrt{T}} - \frac{1}{9T} & \text{for } k = 1 \\ 1 & \text{for } k = \infty \end{cases}, \qquad F_Q^b(k) = \begin{cases} \frac{1}{2} - \frac{1}{3\sqrt{T}} & \text{for } k = 0 \\ \frac{3}{4} + \frac{1}{3\sqrt{T}} - \frac{1}{9T} & \text{for } k = 1 \\ 1 & \text{for } k = \infty. \end{cases}$$

Since the optimal inventory level $y^* = \min\{y \geq 0 : \mathbb{P}(Q \leq y) \geq \frac{b}{h+b}\}$, it is clear that the optimal inventory level for F_Q^a is 0 and that for F_Q^b is 1 because $F_Q^b(1) > 1/2$. We prove that, no policy can achieve a worst-case expected regret better than $\Omega(\sqrt{T})$. We will use the fact that for discrete demand, we have

$$C(y) - C(y^*) = \int_0^T (h+b) \sum_{i=\min\{y^*,y\}}^{\max\{y^*,y\}-1} \left| \frac{b}{b+h} - F_Q(i) \right| dt.$$

Let π be an arbitrary policy. The worst-case expected regret under policy π is bounded

below as follows:

$$\begin{aligned}
& \sup_{F\in\mathcal{F}}\left\{\int_0^T \left(hx(t)^+ + bx(t)^+ - hx^*(t)^+ - bx^*(t)^+\right) dt\right\} \\
= & \sup_{F\in\{F_Q^a, F_Q^b\}} \{C(y) - C(y^*)\} \\
\geq & (b+h)\frac{1}{6\sqrt{T}} \max\left\{\int_0^T \mathbb{P}_a^\pi\left(y^\pi > \frac{1}{2}\right), \int_0^T \mathbb{P}_b^\pi\left(y^\pi \leq \frac{1}{2}\right)\right\} \\
\geq & (b+h)\frac{1}{12\sqrt{T}} \int_0^T \max\left\{\mathbb{P}_a^\pi\left(y^\pi > \frac{1}{2}\right), \mathbb{P}_b^\pi\left(y^\pi \leq \frac{1}{2}\right)\right\},
\end{aligned}$$

By Theorem 2.2 in *Tsybakov* (2009), we have

$$\max\left\{\mathbb{P}_a^\pi\left(y^\pi > \frac{1}{2}\right), \mathbb{P}_b^\pi\left(y^\pi \leq \frac{1}{2}\right)\right\} \geq \frac{1}{12}\exp\{-\mathcal{K}_{t-1}(\mathbb{P}_a, \mathbb{P}_b)\},$$

where

$$\mathcal{K}_t(\mathbb{P}_a, \mathbb{P}_b) = \mathbb{E}_a\left[\log\frac{\mathbb{P}_a(Q_1, \cdots, Q_t)}{\mathbb{P}_b(Q_1, \cdots, Q_t)}\right]$$

is the Kullback-Leibler divergence *Kullback and Leibler* (1951) between the distribution of $Q_1, \cdots, Q_t$ under F_Q^a and F_Q^b, which is equal to

$$\mathcal{K}_t(\mathbb{P}_a, \mathbb{P}_b) = t\left[\left(\frac{1}{2} + \frac{1}{3\sqrt{T}}\right)\log\left(\frac{1+\frac{2}{3\sqrt{T}}}{1-\frac{2}{3\sqrt{T}}}\right) + \left(\frac{1}{2} - \frac{1}{3\sqrt{T}}\right)\log\left(\frac{1-\frac{2}{3\sqrt{T}}}{1+\frac{2}{3\sqrt{T}}}\right)\right]. \tag{4.9}$$

By Taylor's theorem, one can establish that for all $x \in (0, 1/2)$,

$$2x \leq \log\frac{1+x}{1-x} \leq 2x + 2x^2. \tag{4.10}$$

Therefore, by substituting (4.10) into (4.9), we obtain $\mathcal{K}_t(\mathbb{P}_a, \mathbb{P}_b) \leq \frac{33t}{8T}$. Then we have

$$\max\left\{\mathbb{P}_a^\pi\left(y^\pi > \frac{1}{2}\right), \mathbb{P}_b^\pi\left(y^\pi \leq \frac{1}{2}\right)\right\} \geq \frac{1}{12}e^{-33/8},$$

which leads to

$$\begin{aligned}\sup\left\{\int_0^T hx(t)^+ + bx(t)^+ - hx^*(t)^+ - bx^*(t)^+\right\} &\geq (b+h)\frac{1}{12\sqrt{T}}\int_0^T \frac{1}{12}e^{-33/8} \\ &\geq \frac{1}{72}e^{-33/8}\sqrt{T}.\end{aligned}$$

Therefore, we have shown that even for this simple case, the lower bound of any learning algorithms is $\Omega(\sqrt{T})$. □

4.5 Numerical Experiments

We conduct numerical experiments to demonstrate the efficacy of our proposed algorithm. To the best of our knowledge, there are no existing learning algorithms in the literature. Thus, we designed an intuitive heuristic to compare against. Our numerical result shows that the proposed algorithm outperforms the heuristic. Moreover, the more loaded the system becomes, the greater improvement our algorithm achieves.

4.5.1 Design of Experiments

The customer arrival process is a Poisson process with rate $1/\lambda = 20$. The production time is tested through different normal distributions, e.g., $\mathbf{N}(8, 5^2)$, $\mathbf{N}(10, 5^2)$, $\mathbf{N}(12, 5^2)$, and gamma distributions with mean 14.

The holding cost for holding one product in the inventory is $h = 0.2$ per unit time. The penalty cost for backlogging one customer (due to insufficient inventory) is $b = 10$ per unit time. We set the time horizon to be $T = 100000$, and compare the average cost of our algorithm against the average cost of the heuristic and the optimal average cost. The initial inventory for both our algorithm and the heuristic is set to be $s_0 = 0$.

Clairvoyant Optimal Policy: The clairvoyant optimal policy is a stationary policy. Given that the decision maker knows the distribution of the production time and the customer arriving rate, the optimal make-to-stock level can be calculated using simulation.

A Simple Benchmark Heuristic: The heuristic works as follows. It starts with some arbitrary target inventory s_0 (we set $s_0 = 0$ here for convenience). The heuristic will record the production and inter-arrival times along the process. Every time that there is a new product finished or a new customer arrived, the heuristic generates a new empirical distribution of the production time or the arriving rate, and then calculate the next target level based on these empirical distributions. During the process, the facility keeps producing whenever the target level is higher than the current inventory level, and stays idle otherwise.

Multi-Start DMTS: The algorithm starts with some arbitrary target inventory s_0 (we also set $s_0 = 0$). Different than the original DMTS, we assume multiple virtual systems with different starting points $s_0' = \{1, 2, \cdots, \bar{s}\}$. Next, we will follow the steps in DMTS to calculate G_k using realized inter-arrival and production times for every virtual starting points and obtain multiple $\hat{s}_{k+1}$. We will pick the actual $\hat{s}_{k+1}$ to be the one with the minimum average cost. The rest is the same as DMTS. It is evident that the algorithm preserves the convergence result (regardless of the starting point).

4.5.2 Numerical Results and Findings

We compare our algorithm with the heuristic through three performance metrics. The first metric is the time to achieve within 5% error within the clairvoyant optimal cost. Note that this time is in terms of the time of the queueing system, not the computational time. The second metric is the improvement of empirical convergence rate of our algorithm over the heuristic, which is calculated as the percentage difference of the 5% optimality convergence time between our algorithm and the heuristic. The

Inter-arrival Time	Production Time	Time to Achieve 5% Optimality Gap	Improvement in Covergence Rate	Reduction in Policy Fluctuation
20	$\mathbf{N}(8,5)$	7641	1.65%	-33.64%
20	$\mathbf{N}(10,5)$	7164	2.00%	-13.99%
20	$\mathbf{N}(12,5)$	5891	31.00%	-5.25%
20	$\mathbf{N}(14,5)$	16114	43.96%	34.55%
20	$\mathbf{N}(16,5)$	14600	50.05%	57.48%
20	$\mathbf{N}(18,5)$	17985	56.87%	68.91%
20	$\mathbf{Gamma}\left(\frac{14^2}{3^2}, \frac{3^2}{14}\right)$	7340	0%	30.20%
20	$\mathbf{Gamma}\left(\frac{14^2}{5^2}, \frac{5^2}{14}\right)$	14519	37.73%	35.26%
20	$\mathbf{Gamma}\left(\frac{14^2}{7^2}, \frac{7^2}{14}\right)$	7546	50.02%	32.91%
20	$\mathbf{Gamma}\left(\frac{14^2}{9^2}, \frac{9^2}{14}\right)$	19417	31.48%	43.30%

Table 4.2: Summary of Computational Results

third metric is the reduction in policy fluctuation of our algorithm compared with the heuristic. The policy fluctuation here is defined to be the average change of two consecutive target levels. Note that the higher the fluctuation is, the more difficult it would be to implement the policy in practice. For each test case, we run both algorithms 1000 times and take the average performance. The numerical results are shown in Table 4.2.

The numerical results show that our algorithm achieves a better empirical convergence rate. We find that the higher the system utilization factor is, the greater the improvement our algorithm achieves. The reason can be explained by the policy fluctuation. When the utilization factor is higher, the optimal target level calculated based on empirical distributions can be fluctuating drastically, thus making the heuristic converge slowly to the clairvoyant optimal cost. In contrast, our algorithm exhibits a much smoother trajectory. We also find that our policy, in general, performs better when the variance of the production time is higher.

4.6 Concluding Remark

In this chapter, we have proposed an adaptive learning algorithm for a make-to-stock queueing system, where both the customer arriving rate and the production time distribution are unknown to the decision maker *a priori*. The algorithm is a stochastic gradient descent type, ensuring that the policy converges to the clairvoyant optimal policy. One key idea is that following the rules of our algorithm, one can effectively couple the production cycles of the actual implemented system and the virtual system. We have shown that the average T-period running cost converges to the clairvoyant optimal cost at the rate of $O(1/\sqrt{T})$, which is theoretically the best possible for this class of problems.

To close this chapter, we would like to point out several promising future research avenues. First, one could consider a make-to-stock queue with general inter-arrival distributions. Second, it would be interesting to see if one can incorporate setup cost or setup time into the model and devise a provably-good learning algorithm. Third, there are many other core queueing systems or networks, and we hope this work serves as a gateway to this topic.

CHAPTER V

Conclusion

This dissertation focuses on the data-driven management of inventory and queueing systems. Different than the conventional approach of first finding the best probabilistic representations of uncertainties and then carrying out the stochastic optimization, we develop a non-parametric approach focusing on the (continuous) interplay between learning and optimization.

The three essays presented in the previous chapters study three canonical stochastic systems through structured ways of trading off exploration and exploitation. They also give insights on how to establish the theoretical convergence rates when applying a stochastic gradient descent based algorithm with added constraints on inventory and timing. There are several promising future research directions. First, there might be other important factors that need to be considered and incorporated, e.g., fixed cost, seasonal and nonstationary demand, pricing decisions. Second, the methods developed in this thesis could be applied to tackle stochastic systems with censored data, physical constraints, and complex state transitions in other domains.

BIBLIOGRAPHY

BIBLIOGRAPHY

Ahn, H.-S., I. Duenyas, and R. Q. Zhang (1999), Optimal stochastic scheduling of a two-stage tandem queue with parallel servers, *Advances in Applied Probability*, *31*(4), 1095–1117.

Ahn, H.-S., I. Duenyas, and M. E. Lewis (2002), Optimal control of a two-stage tandem queuing system with flexible servers, *Probability in the Engineering and Informational Sciences*, *16*(4), 453–469.

Angelus, A., and W. Zhu (2017), Looking upstream: Optimal policies for a class of capacitated multi-stage inventory systems, *Production and Operations Management*, *26*(11), 2071–2088.

Aviv, Y., and A. Federgruen (1997), Stochastic inventory models with limited production capacity and periodically varying parameters, *Probab. Engrg. Informational Sci.*, *11*, 107–135.

Bertsekas, D. P. (2000), *Dynamic Programming and Optimal Control*, 2nd ed., Athena Scientific.

Bertsekas, D. P., and S. E. Shreve (2007), *Stochastic Optimal Control: The Discrete-Time Case*, Athena Scientific.

Bertsimas, D., and I. C. Paschalidis (2001), Probabilistic service level guarantees in make-to-stock manufacturing systems, *Operations Research*, *49*(1), 119–133.

Besbes, O., and A. Muharremoglu (2013), On implications of demand censoring in the newsvendor problem, *Management Science*, *59*(6), 1407–1424.

Beyer, D., S. P. Sethi, and R. Sridhar (2001), Stochastic multi-product inventory models with limited storage, *Journal of Optimization Theory and Applications*, *111*, 553–588.

Beyer, D., S. P. Sethi, and R. Sridhar (2002), Average-cost optimality of a base-stock policy for a multi-product inventory model with limited storage, in *Decision & Control in Management Science*, *Advances in Computational Management Science*, vol. 4, edited by G. Zaccour, pp. 241–260, Springer, New York, NY.

Bookbinder, J. H., and A. E. Lordahl (1989), Estimation of inventory re-order levels using the bootstrap statistical procedure, *IIE Transactions*, *21*(4), 302–312.

Boyd, S., and L. Vandenberghe (2004), *Convex Optimization*, Cambridge University Press, New York, NY, USA.

Brownlee, J. (2014), Manufacturing problems could make the iPhone 6 hard to find at launch., online; accessed 29 October 2018.

Burnetas, A. N., and C. E. Smith (2000), Adaptive ordering and pricing for perishable products, *Operations Research*, *48*(3), 436–443.

Caldentey, R., and L. M. Wein (2006), Revenue management of a make-to-stock queue, *Operations Research*, *54*(5), 859–875.

Chen, B., X. Chao, and C. Shi (2015), Nonparametric algorithms for joint pricing and inventory control with lost-sales and censored demand, Working paper, University of Michigan, Ann Arbor, MI.

Chen, B., X. Chao, and H.-S. Ahn (2019a), Coordinating pricing and inventory replenishment with nonparametric demand learning, forthcoming in Operations Research.

Chen, L., and E. L. Plambeck (2008), Dynamic inventory management with learning about the demand distribution and substitution probability, *Manufacturing & Service Operations Management*, *10*(2), 236–256.

Chen, W., C. Shi, and I. Duenyas (2018a), Nonparametric algorithms for stochastic inventory systems with random capacity, Working paper, University of Michigan, Ann Arbor, MI.

Chen, W., C. Shi, and I. Duenyas (2019b), Optimal learning algorithm for make-to-stock queueing systems, Working paper, University of Michigan, Ann Arbor, MI.

Chen, X., X. Gao, and Z. Pang (2018b), Preservation of structural properties in optimization with decisions truncated by random variables and its applications, *Operations Research*, *66*(2), 340–357.

Choi, J., J. J. Cao, H. E. Romeijn, J. Geunes, and S. X. Bai (2005), A stochastic multi-item inventory model with unequal replenishment intervals and limited warehouse capacity, *IIE Transactions*, *37*(12), 1129–1141.

Chu, L. Y., J. G. Shanthikumar, and Z.-J. M. Shen (2008), Solving operational statistics via a bayesian analysis, *Operations Research Letters*, *36*(1), 110 – 116.

Ciarallo, F. W., R. Akella, and T. E. Morton (1994), A periodic review, production planning model with uncertain capacity and uncertain demand — optimality of extended myopic policies, *Management Science*, *40*(3), 320–332.

Crabill, T. B., D. Gross, and M. J. Magazine (1977), A classified bibliography of research on optimal design and control of queues, *Operations Research*, *25*(2), 219–232.

Duenyas, I., W. J. Hopp, and Y. Bassok (1997), Production quotas as bounds on interplant JIT contracts, *Management Science*, *43*(10), 1372–1386.

Duenyas, I., D. Gupta, and T. L. Olsen (1998), Control of a single-server tandem queueing system with setups, *Operations Research*, *46*(2), 218–230.

Evans, R. V. (1967), Inventory control of a multiproduct system with a limited production resource, *Naval Research Logistics*, *14*(2), 173–184.

Federgruen, A., and N. Yang (2011), Procurement strategies with unreliable suppliers, *Operations research*, *59*(4), 1033–1039.

Federgruen, A., and P. Zipkin (1986a), An inventory model with limited production capacity and uncertain demands I: The average-cost criterion, *Mathematics of Operations Research*, *11*(2), 193–207.

Federgruen, A., and P. Zipkin (1986b), An inventory model with limited production capacity and uncertain demands II: The discounted-cost criterion, *Mathematics of Operations Research*, *11*(2), 208–215.

Feng, Q. (2010), Integrating dynamic pricing and replenishment decisions under supply capacity uncertainty, *Management Science*, *56*(12), 2154–2172.

Feng, Y., and B. Xiao (2002), Optimal threshold control in discrete failure-prone manufacturing systems, *IEEE Transactions on Automatic Control*, *47*(7), 1167–1174.

Feng, Y., and H. Yan (2000), Optimal production control in a discrete manufacturing system with unreliable machines and random demands, *IEEE Transactions on Automatic Control*, *45*(12), 2280–2296.

Flaxman, A. D., A. T. Kalai, and H. B. McMahan (2005), Online convex optimization in the bandit setting: Gradient descent without a gradient, in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pp. 385–394.

Gavish, B., and S. C. Graves (1980), A one-product production/inventory problem under continuous review policy, *Operations Research*, *28*(5), 1228–1236.

Glasserman, P. (1991), *Gradient Estimation Via Perturbation Analysis*, Kluwer international series in engineering and computer science: Discrete event dynamic systems, Springer, New York, NY.

Godfrey, G. A., and W. B. Powell (2001), An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution, *Management Science*, *47*(8), 1101–1112.

Güllü, R. (1998), Base stock policies for production/inventory problems with uncertain capacity levels, *European Journal of Operational Research*, *105*(1), 43–51.

H. Zipkin, P. (1995), Performance analysis of a multi-item production-inventory system under alternative policies, *Management Science*, *41*, 690–703.

Ha, A. Y. (1997a), Inventory rationing in a make-to-stock production system with several demand classes and lost sales, *Management Science*, *43*(8), 1093–1103.

Ha, A. Y. (1997b), Optimal dynamic scheduling policy for a make-to-stock production system, *Operations Research*, *45*(1), 42–53.

Ha, A. Y. (2000), Stock rationing in an M/Ek/1 make-to-stock queue, *Management Science*, *46*(1), 77–87.

Hazan, E. (2016), Introduction to online convex optimization, *Found. Trends Optim.*, *2*(3-4), 157–325.

Hazan, E., A. Kalai, S. Kale, and A. Agarwal (2006), Logarithmic regret algorithms for online convex optimization, in *In 19th COLT*, pp. 499–513.

Henig, M., and Y. Gerchak (1990), The structure of periodic review policies in the presence of random yield, *Operations Research*, *38*(4), 634–643.

Huh, W. H., and P. Rusmevichientong (2009), A non-parametric asymptotic analysis of inventory planning with censored demand, *Mathematics of Operations Research*, *34*(1), 103–123.

Huh, W. H., P. Rusmevichientong, R. Levi, and J. Orlin (2011), Adaptive data-driven inventory control with censored demand based on kaplan-meier estimator, *Operations Research*, *59*(4), 929–941.

Huh, W. T., and M. Nagarajan (2010), Linear inflation rules for the random yield problem: Analysis and computations, *Operations research*, *58*(1), 244–251.

Huh, W. T., G. Janakiraman, J. A. Muckstadt, and P. Rusmevichientong (2009), An adaptive algorithm for finding the optimal base-stock policy in lost sales inventory systems with censored demand, *Mathematics of Operations Research*, *34*(2), 397–416.

Ignall, E., and A. F. Veinott (1969), Optimality of myopic inventory policies for several substitute products, *Management Science*, *15*(5), 284–304.

Iravani, S. M. R., M. J. M. Posner, and J. A. Buzacott (1997), A two-stage tandem queue attended by a moving server with holding and switching costs, *Queueing Systems*, *26*(3-4), 203–228.

Kapuscinski, R., and S. Tayur (1998), A capacitated production-inventory model with periodic demand, *Operations Research*, *46*(6), 899–911.

Kapuscinski, R., and S. Tayur (1999), Optimal policies and simulation-based optimization for capacitated production inventory systems, in *Quantitative Models for Supply Chain Management*, pp. 7–40, Springer, New York, NY.

Karlin, S. (1958), *Optimal Inventory Policy for the Arrow-Harris-Marschak Dynamic Model*, Stanford University Press, Stanford, California., in K. Arrow, S. Karlin, and H. Scarf (Eds.), Studies in the Mathematical Theory of Inventory and Production.

Kleywegt, A. J., A. Shapiro, and T. Homem-de Mello (2002), The sample average approximation method for stochastic discrete optimization, *SIAM J. on Optimization*, *12*(2), 479–502.

Kullback, S., and R. A. Leibler (1951), On information and sufficiency, *Ann. Math. Statist.*, *22*(1), 79–86.

Kunnumkal, S., and H. Topaloglu (2008), Using stochastic approximation methods to compute optimal base-stock levels in inventory control problems, *Operations Research*, *56*(3), 646–664.

Lariviere, M. A., and E. L. Porteus (1999), Stalking information: Bayesian inventory management with unobserved lost sales, *Management Science*, *45*(3), 346–363.

Levi, R., R. O. Roundy, and D. B. Shmoys (2007), Provably near-optimal sampling-based policies for stochastic inventory control models, *Mathematics of Operations Research*, *32*(4), 821–839.

Levi, R., R. O. Roundy, D. B. Shmoys, and V. A. Truong (2008), Approximation algorithms for capacitated stochastic inventory models, *Operations Research*, *56*, 1184–1199.

Levi, R., G. Perakis, and J. Uichanco (2015), The data-driven newsvendor problem: New bounds and insights, *Operations Research*, *63*(6), 1294–1306.

Li, L. (1992), The role of inventory in delivery-time competition, *Management Science*, *38*(2), 182–197.

Lippman, S. A. (1975), Applying a new device in the optimization of exponential queuing systems, *Operations Research*, *23*(4), 687–710.

Lippman, S. A., and S. Stidham (1977), Individual versus social optimization in exponential congestion systems, *Operations Research*, *25*(2), 233–247.

Liyanage, L. H., and J. G. Shanthikumar (2005), A practical inventory control policy using operational statistics, *Operations Research Letters*, *33*(4), 341 – 348.

Loulou, R. (1978), An explicit upper bound for the mean busy period in a GI/G/1 queue, *Journal of Applied Probability*, *15*(2), 452–455.

Maglaras, C., and S. Eren (2015), A maximum entropy joint demand estimation and capacity control policy, *Production and Operations Management*, *24*(3), 438–450.

Nemirovski, A., A. Juditsky, G. Lan, and A. Shapiro (2009), Robust stochastic approximation approach to stochastic programming, *SIAM J. on Optimization*, *19*(4).

Özer, O., and W. Wei (2004), Inventory control with limited capacity and advance demand information, *Operations Research*, *52*(6), 988–1000.

Powell, W., A. Ruszczyński, and H. Topaloglu (2004), Learning algorithms for separable approximations of discrete stochastic optimization problems, *Mathematics of Operations Research*, *29*(4), 814–836.

Randall, T., and D. Halford (2018), Tesla model 3 tracker, online; accessed 29 October 2018.

Roundy, R. O., and J. A. Muckstadt (2000), Heuristic computation of periodic-review base stock inventory policies, *Management Science*, *46*(1), 104–109.

Schäl, M. (1993), Average optimality in dynamic programming with general state space, *Math. Oper. Res.*, *18*(1), 163–172.

Scott Carr, I. D. (2000), Optimal admission control and sequencing in a make-to-stock/make-to-order production system, *Operations Research*, *48*(5), 709–720.

Shaked, M., and J. G. Shanthikumar (2007), *Stochastic Orders*, Springer Series in Statistics, Physica-Verlag.

Shalev-Shwartz, S. (2012), Online learning and online convex optimization, *Found. Trends Mach. Learn.*, *4*(2), 107–194.

Shi, C., W. Chen, and I. Duenyas (2016), Nonparametric data-driven algorithms for multiproduct inventory systems with censored demand, *Operations Research*, *64*(2), 362–370.

Simchi-Levi, D., X. Chen, and J. Bramel (2014), *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics and Supply Chain Management*, Springer Series in Operations Research and Financial Engineering, Springer, New York, NY.

Snyder, L. V., and Z.-J. M. Shen (2011), *Fundamentals of supply chain theory*, John Wiley & Sons, Hoboken, New Jersey.

Sobel, M. J. (1982), The optimality of full service policies, *Operations Research*, *30*(4), 636–649.

Sohail, O. (2018), Production problems might delay LCD iPhone 9 model to launch in november - notch said to be the culprit., online; accessed 29 October 2018.

Sparks, D. (2018), Tesla model 3 production rate: 3,000 units per week, online; accessed 29 October 2018.

Stidham, S. (1978), Socially and individually optimal control of arrivals to a GI/M/1 queue, *Management Science*, *24*(15), 1598–1610.

Stidham, S. (1985), Optimal control of admission to a queueing system, *IEEE Transactions on Automatic Control*, *30*(8), 705–713.

Tayur, S. (1992), Computing the optimal policy for capacitated inventory models, *Stochastic Models*, *9*, 585–598.

Tsybakov, A. (2009), *Introduction to Nonparametric Estimation*, Springer-Verlag, New York, NY.

Veatch, M. H., and L. M. Wein (1996), Scheduling a make-to-stock queue: Index policies and hedging points, *Operations Research*, *44*(4), 634–647.

Veinott, A. F. (1965), Optimal policy for a multi-product, dynamic, nonstationary inventory problem, *Management Science*, *12*(3), pp. 206–222.

Veinott, A. F. (1966), On the optimality of (s,S) inventory policies: new conditions and a new proof, *SIAM J. Appl. Math*, *14*, 1067–1083.

Wang, Y., and Y. Gerchak (1996), Periodic review production models with variable capacity, random yield, and uncertain demand, *Management science*, *42*(1), 130–137.

Wein, L. M. (1992), Dynamic scheduling of a multiclass make-to-stock queue, *Operations Research*, *40*(4), 724–735.

Zhang, H., X. Chao, and C. Shi (2018), Perishable inventory systems: Convexity results for base-stock policies and learning algorithms under censored demand, *Operations Research*, *66*(5), 1276–1286.

Zhang, H., X. Chao, and C. Shi (2019), Closing the gap: A learning algorithm for the lost-sales inventory system with lead times, forthcoming in Management Science.

Zheng, Y.-S., and P. Zipkin (1990), A queueing model to analyze the value of centralized inventory information, *Operations Research*, *38*(2), 296–307.

Zinkevich, M. (2003), Online convex programming and generalized infinitesimal gradient ascent, in *Proceedings of the 20th International Conference on Machine Learning (ICML)*, edited by T. Fawcett and N. Mishra, pp. 928–936, AAAI Press, Cambridge, MA, USA.

Zipkin, P. (2000), *Foundations of Inventory Management*, McGraw-Hill, New York.