Chapter 12
Joint Pricing and Inventory Control with Demand Learning

Boxiao Chen

12.1 Problem Formulation in General

Since the seminal paper of Whitin (1955), joint pricing and inventory control problems have attracted tremendous attention and been studied by hundreds of research papers in the literature. For a comprehensive review, see survey papers Petruzzi and Dada (1999), Elmaghraby and Keskinocak (2003), Yano and Gilbert (2005), and Chen and Simchi-Levi (2012). Traditional literature assumes the demand distribution is known and takes this information as model input, which is hardly satisfied in practice. In this chapter, we relax this assumption and discuss online algorithms to learn the demand only from historical data. As time goes by, the learning algorithms will learn the demand better and better, so that the solutions prescribed by the algorithms converge to the true optimal solution had the demand distribution been known.

In this section, we discuss the general setup for the problem of joint inventory and pricing. Consider a periodic review system in which a firm (e.g., a retailer) sells a non-perishable product over a planning horizon of T periods. At the beginning of each period t, the firm observes on-hand inventory x_t and determines an inventory order-up-to level y_t and a price p_t, where $y_t \geq x_t,\ y_t \in \mathcal{Y} = [y^l, y^h]$ and $p_t \in \mathcal{P} = [p^l, p^h]$ with $y^l < y^h$ and $p^l < p^h$. For simplicity we assume that the system is initially empty, i.e., $x_1 = 0$. Demand for period t, denoted by $D_t(p_t)$, is stochastic and price dependent. Demand is satisfied as much as possible by on-hand inventory, and profits are collected by the firm. There might be a mismatch between supply and demand. If $y_t > D_t(p_t)$, any leftover inventories will be carried over to the next period, for each of which the firm pays a holding cost h. If $y_t < D_t(p_t)$,

B. Chen (✉)
College of Business Administration, University of Illinois Chicago, Chicago, IL, USA
e-mail: bbchen@uic.edu

X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_12

some demands are not fulfilled, and the firm pays a penalty cost b for any unit of stockout. Per-unit ordering cost is normalized to 0 without loss of generality. The firm's objective is to maximize the T-period total profit.

If the distribution of $D_t(p_t)$ is known a priori to the firm (complete information scenario), then the optimization problem the firm wishes to solve is

$$\max_{\substack{(p_t, y_t) \in \mathcal{P} \times \mathcal{Y} \\ y_t \geq x_t}} \sum_{t=1}^{T} v(p_t, y_t), \tag{12.1}$$

where $v(p_t, y_t)$ is the instantaneous reward during period t. Let V^* represent the maximum T-period expected profit generated from the optimal policy under complete information.

In practice, the demand distribution is unknown; therefore, the firm needs to develop an admissible policy which prescribes pricing and ordering decisions for each period. An admissible policy is represented by a sequence of prices and order-up-to levels, $\{(p_t, y_t), t \geq 1\}$, where (p_t, y_t) depends only on realized data and decisions made prior to period t, and $y_t \geq x_t$, i.e., (p_t, y_t) is adapted to the filtration generated by $\{(p_s, y_s), o_s : s = 1, \ldots, t-1\}$. Here o_s represents the observable data of demand. Ideally, $o_s = D_s(p_s)$, meaning that demand is fully observable, but in some cases demand data is censored, which yields $o_s < D_s(p_s)$. Given any admissible policy π, the sequence of events for each period t is described as follows:

1. At the beginning of period t, the retailer observes the initial inventory level x_t.
2. The retailer decides the selling price p_t and the inventory level-up-to level $y_t \geq x_t$. New orderings, if there is any, arrive instantaneously.
3. Demand realizes and is satisfied to the maximum extent using on-hand inventory. Unsatisfied demand is backlogged or lost, and any leftover inventory is carried to the next period. The retailer observes data o_t.
4. At the end of period t, the retailer collects profit of the current period.

The firm's objective is to find an admissible policy to maximize the T-period total profit while learning the unknown demand distribution on the fly. The regret of policy π, denoted by $R^{\pi}(T)$, is defined as the total profit loss over T periods, which is

$$R^{\pi}(T) = V^* - \mathbb{E}\left[\sum_{t=1}^{T} v(p_t, y_t)\right].$$

The smaller the regret, the better the policy.

In this chapter, we will discuss a number of models under the framework of joint inventory and pricing. These models differ in the following three dimensions.

1. Backlog versus lost-sales
   - In a backlog system, if $y_t < D_t(p_t)$, any unsatisfied demands will be backlogged and served in future periods, and $x_{t+1} = y_t - D_t(p_t)$.
   - In a lost-sales system, unmet demands will leave the market without any purchases, and $x_{t+1} = (y_t - D_t(p_t))^+$.
2. Unlimited price changes versus limited price changes
   - Most models we will discuss allow unlimited number of price changes, i.e., the retailer is allowed to change price every period.
   - We will discuss one model where the firm is not allowed to make price changes more than a certain number of times.
3. With versus without setup cost
   - If a setup cost is present, a fixed amount of fee will be charged whenever a positive amount of inventory is ordered.

In Sects. 12.2 and 12.3, we discuss the classic joint inventory and pricing problem with backlogged demand and lost sales, respectively. In Sect. 12.4, we consider scenarios with a limited number of price changes. In Sect. 12.5, we discuss the joint pricing and inventory control problem with setup cost. In Sect. 12.6, we discuss other models of joint pricing and inventory control that have been studied in the literature.

12.2 Nonparametric Learning for Backlogged Demand

In this section, we discuss the joint pricing and inventory control problem with backlogged demand, one of the most classical models under the topic of joint pricing and inventory control. We will discuss the model, algorithm, regret convergence results and proof sketch based on Chen et al. (2019).

Per-period demand can be either $D_t(p_t) = \lambda(p_t) + \epsilon_t$ (additive) or $D_t(p_t) = \lambda(p_t)\,\epsilon_t$ (multiplicative), where $\lambda(\cdot)$ is a strictly decreasing deterministic function and $\epsilon_t, t = 1, 2, \ldots, T$, are independent and identically distributed random variables with probability density function $f(\cdot)$ and cumulative distribution function $F(\cdot)$. Here we focus on the multiplicative demand form. Unsatisfied demands are backlogged, and one has $x_{t+1} = y_t - D_t(p_t)$ for all $t = 1, \ldots, T$. The instantaneous reward for period t is $G(p_t, y_t) = p_t\mathbb{E}[D_t(p_t)] - h\mathbb{E}[y_t - D_t(p_t)]^+ - b\mathbb{E}[D_t(p_t) - y_t]^+$.

By Sobel (1981), myopic policy is optimal for this problem. Therefore, to optimize the T-period problem in (12.1), it suffices to solve the single-period problem

$$\max_{(p,y)\in\mathcal{P}\times\mathcal{Y}} G(p, y), \tag{12.2}$$

where

$$G(p, y) = p\mathbb{E}[D(p)] - h\mathbb{E}[y - D(p)]^{+} - b\mathbb{E}[D(p) - y]^{+}$$
$$= pe^{\lambda(p)}\mathbb{E}\big[e^{\epsilon}\big] - \Big\{h\mathbb{E}\big[y - e^{\lambda(p)}e^{\epsilon}\big]^{+} + b\mathbb{E}\big[e^{\lambda(p)}e^{\epsilon} - y\big]^{+}\Big\}.$$

Let $Q(p, e^{\lambda(p)}) := \max_{y\in\mathcal{Y}} G(p, y)$, then problem (12.2) can be re-written as

Problem CI:

$$\max_{p\in\mathcal{P}} Q(p, e^{\lambda(p)})$$
$$:= \max_{p\in\mathcal{P}} \Big\{pe^{\lambda(p)}\mathbb{E}\big[e^{\epsilon}\big] - \min_{y\in\mathcal{Y}} \Big\{h\mathbb{E}\big[y - e^{\lambda(p)}e^{\epsilon}\big]^{+} + b\mathbb{E}\big[e^{\lambda(p)}e^{\epsilon} - y\big]^{+}\Big\}\Big\}. \tag{12.3}$$

The inner optimization problem (minimization) determines the optimal order-up-to level that minimizes the expected holding and backlog cost for a given price p, and we denote it by $\overline{y}\left(e^{\lambda(p)}\right)$. The outer optimization solves for the optimal price p. Because (p^*, y^*) is the optimal solution for (12.3), they satisfy $y^* = \overline{y}(e^{\lambda(p^*)})$.

The firm knows neither the function $\lambda(\cdot)$ nor the distribution of random variable ϵ_t. In the backlog system, true demand realizations can be observed. Therefore, $o_t = D_t(p_t)$, and an admissible policy (p_t, y_t) is adapted to the filtration generated by $\{(p_s, y_s), D_s(p_s) : s = 1, \ldots, t-1\}$.

Learning Algorithm A learning algorithm named DDA (shorthand for Data-Driven Algorithm) is proposed in Chen et al. (2019). DDA approximates $\lambda(p)$ by an affine function, and it constructs empirical and dependent error samples from the collected data, called centered samples. DDA divides the planning horizon into stages whose lengths are exponentially increasing (in the stage index). At the start of each stage, the firm sets two pairs of prices and order-up-to levels based on its current linear estimation of the demand-price function and the constructed centered samples of random error, and the collected demand data from this stage are used to update the linear estimation of the demand-price function and the empirical distribution of random error. These are then utilized to find the pricing and inventory decision for the next stage. The detailed algorithm design is presented in Algorithm 1.

As shown in Algorithm 1, for $i = 1, 2, \ldots$ in the DDA algorithm, iteration i focuses on stage i that consists of $2I_i$ periods. The algorithm sets the ordering quantity and selling price for each period in stage i derived from the previous iteration. The first I_i periods (from $t_i + 1$ to $t_i + I_i$) try to implement order-up-to $\hat{y}_{i,1}$ policy while the second I_i periods try to implement order-up-to $\hat{y}_{i,2}$ policy. Because starting inventory level may be higher than the order-up-to level, $\hat{y}_{i,1}$ and $\hat{y}_{i,2}$ may not be achieved, and one challenge is to identify the impact of the carryover inventory constraint on the performance of a learning algorithm.

Algorithm 1 Data-Driven Algorithm (DDA)

```
Input $v > 1$, $\rho > 0$ and $I_0 > 0$, and $\hat{p}_1, \hat{y}_{1,1}, \hat{y}_{1,2}$. Compute $I_1 = \lfloor I_0 v \rfloor$, $\delta_1 = \rho(2I_0)^{-\frac{1}{4}}$, and $\hat{p}_1 + \delta_1$.
for $i = 1, \dots, n$ do
    for $t = t + i + 1, \dots, t_i + I_i$ do
        Set $p_t = \hat{p}_i$, $y_t = \max\{\hat{y}_{i,1}, x_t\}$.
        Let $D_t = \log \tilde{D}_t(p_t)$.
    end for
    for $t = t_i + I_i + 1, \dots, t_i + 2I_i$ do
        Set $p_t = \hat{p}_i + \delta_i$, $y_t = \max\{\hat{y}_{i,2}, x_t\}$.
        Let $D_t = \log \tilde{D}_t(p_t)$.
    end for
    Compute
        $(\hat{\alpha}_{i+1}, \hat{\beta}_{i+1}) = \operatorname{argmin}_{\alpha,\beta} \left\{ \sum_{t=t_i+1}^{t_i+2I_i} \Big(D_t - (\alpha - \beta p_t)\Big)^2 \right\},$
        $\eta_t = D_t - \frac{1}{I_i} \sum_{t=t_i+1}^{t_i+I_i} D_t, \qquad \text{for } t = t_i + 1, \dots, t_i + I_i,$
        $\eta_t = D_t - \frac{1}{I_i} \sum_{t=t_i+I_i+1}^{t_i+2I_i} D_t, \qquad \text{for } t = t_i + I_i + 1, \dots, t_i + 2I_i.$
    The data-driven optimization problem (Problem DD) is
        $\max_{(p,y)\in\mathcal{P}\times\mathcal{Y}} G_{i+1}^{DD}(p, y) = \max_{p\in\mathcal{P}} Q_{i+1}^{DD}\big(p, e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p}\big), \qquad (12.4)$
    where
        $G_{i+1}^{DD}\big(p, y\big) = p e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p} \frac{1}{2I_i} \sum_{t=t_i+1}^{t_i+2I_i} e^{\eta_t}$
        $\quad - \frac{1}{2I_i} \sum_{t=t_i+1}^{t_i+2I_i} \left( h \left( y - e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p + \eta_t} \right)^+ + b \left( e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p + \eta_t} - y \right)^+ \right),$
    and
        $Q_{i+1}^{DD}\big(p, e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p}\big) = \min_{y\in\mathcal{Y}} G_{i+1}^{DD}\big(p, y\big).$
    If $\hat{\beta}_{i+1} > 0$, then solve problem DD and set the first pair of price and inventory level to
        $(\hat{p}_{i+1}, \hat{y}_{i+1,1}) = \arg \max_{(p,y)\in\mathcal{P}\times\mathcal{Y}} G_{i+1}^{DD}(p, y);$
    otherwise, set
        $(\hat{p}_{i+1}, \hat{y}_{i+1,1}) = \left( \frac{p^l + p^h}{2}, \frac{y^l + y^h}{2} \right).$
    Set $\hat{p}_{i+1,2} = \hat{p}_{i+1} + \delta_{i+1}$ (in case $\hat{p}_{i+1} + \delta_{i+1} \notin \mathcal{P}$, set $\hat{p}_{i+1,2} = \hat{p}_{i+1} - \delta_{i+1}$), and
        $\hat{y}_{i+1,2} = \arg\max_{y\in\mathcal{Y}} G_{i+1}^{DD}(\hat{p}_{i+1,2}, y).$
end for
```

The algorithm applies the realized demand data and least-square method to update the linear approximation, $\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p$, of $\lambda(p)$ and computes a centered sample η_t of random error ϵ_t, for $t = t_i + 1, \ldots, t_i + 2I_i$. Note that η_t is not a sample of the random error ϵ_t. This is because $\epsilon_t = D_t(p_t) - \lambda(p_t)$ but $1/I_i \sum_{k=t_1+1}^{t_i+I_i} D_k \neq \lambda(p_t)$. For this reason, the constructed objective function for holding and shortage costs is not a sample average of the newsvendor problem. In the traditional SAA, mathematical expectations are replaced by true sample averages, see, e.g., Kleywegt et al. (2002); Levi et al. (2007, 2015). When only biased samples are available, techniques from statistics such as jackknife resampling can be applied to reduce bias for SAA (Wu et al., 1986). In this work, samples of ϵ_t cannot be observed, however,

$$\eta_t = D_t(p_t) - \frac{1}{I_i} \sum_{k=t_1+1}^{t_i+I_i} D_k = \epsilon_t - \frac{1}{I_i} \sum_{k=t_1+1}^{t_i+I_i} \epsilon_k$$

can be obtained. Since $\mathbb{E}[\epsilon_k] = 0$, $1/I_i \sum_{k=t_1+1}^{t_i+I_i} \epsilon_k$ converges to 0 in probability as I_i grows, and one would expect $\eta_t \to \epsilon_t$ in probability as t grows. Thus, DDA use η_t in place of ϵ_t in computing proxy objectives. Since these samples are obtained from the original i.i.d. samples after subtracting the sample average, we call η_t *centered samples*, and $\{\eta_t, t = t_i + 1, \ldots, t_i + 2I_i\}$ are dependent.

A data-driven optimization problem is then constructed. When $\hat{\beta}_{i+1} > 0$, the algorithm solves an optimization problem of a jointly concave function. Technical analyses in the paper show that the probability for $\hat{\beta}_{i+1} > 0$ converges to 1 as i grows.

The DDA algorithm integrates a process of earning (exploitation) and learning (exploration) in each stage. The earning phase consists of the first I_i periods starting at $t_i + 1$, during which the algorithm implements the optimal strategy for the proxy optimization problem $G_i^{DD}(p, y)$. In the next I_i periods of learning phase that starts from $t_i + I_i + 1$, the algorithm uses a different price $\hat{p}_i + \delta_i$ and its corresponding order-up-to level. The purpose of this phase is to extract demand sensitivity information around the selling price. Note that, even though the firm deviates from the optimal strategy of the proxy problem in the second phase, the policies, $(\hat{p}_i + \delta_i, \hat{y}_{i,2})$ and $(\hat{p}_i, \hat{y}_{i,1})$, will be very close to each other as i increases. Chen et al. (2019) show that they both converge to the clairvoyant optimal solution and the loss of profit from this deviation converges to zero.

Regret Convergence An upper bound for regret of the DDA policy is provided as $R^{DDA}(T) = V^* - \mathbb{E}\left[\sum_{t=1}^{T} G(p_t, y_t)\right] \leq C_1 T^{1/2}$, for some constant $C_1 > 0$. The lower bound for regret is $\Omega(T^{1/2})$, which is implied by Keskin and Zeevi (2014). This shows that the regret convergence rate for DDA is tight.

The intuitions for regret convergence are the following. Note that during cycle i, two distinct prices got implemented, based on which demand data is generated. The two prices are different by δ_i, which decreases to 0 as i increases. Therefore, the two prices are getting closer, and the linear function yielded by linear approximation approaches the tangent line of $\lambda(\cdot)$, providing gradient information for future decisions.

Proof Sketch To compare the DDA policy with the clairvoyant optimal policy, i.e., the optimal solutions of problem DD (12) and problem CI (12.3), note that these two objective functions have significant differences. In problem CI, both $\lambda(p)$ and the distribution of ϵ are known, but in problem DD, $\lambda(p)$ is approximated by a linear function and distribution of ϵ is estimated using centered samples instead of true samples. Therefore, to analyze DDA, the authors' approach is to introduce several "intermediate" bridging problems, and in each step we compare two "adjacent" problems that differ along only one dimension.

First, for parameters α and $\beta > 0$, we introduce bridging problem B1 defined by

$$\begin{aligned}&\textbf{Bridging Problem B1}:\\&\max_{p\in\mathcal{P}} \overline{Q}(p, e^{\alpha-\beta p})\\&:= \max_{p\in\mathcal{P}} \left\{ pe^{\alpha-\beta p}\mathbb{E}\left[e^{\epsilon}\right] - \min_{y\in\mathcal{Y}} \left\{ h\mathbb{E}\left[y - e^{\alpha-\beta p+\epsilon}\right]^{+} + b\mathbb{E}\left[e^{\alpha-\beta p+\epsilon} - y\right]^{+} \right\}\right\}.\end{aligned} \tag{12.5}$$

It is easy to see that, the only difference between problem B1 and problem CI in (12.3) is that, in problem B1 we replace the demand-price function in CI by an affine function $\alpha - \beta p$. Let $\overline{p}\big(\alpha, \beta\big)$ denote the optimal price for problem B1, and for given $p \in \mathcal{P}$, we let $\overline{y}(e^{\alpha-\beta p})$ denote its optimal order-up-to level, which is the optimal solution for the inner minimization problem in (12.5).

The second bridging problem, B2, is defined for each iteration i of the DDA algorithm, and for any α and $\beta > 0$, it is given by

$$\begin{aligned}&\textbf{Bridging Problem B2}:\\&\max_{p\in\mathcal{P}} \tilde{Q}_{i+1}(p, e^{\alpha-\beta p}) := \max_{p\in\mathcal{P}} \left\{ pe^{\alpha-\beta p} \left(\frac{1}{2I_i} \sum_{t=t_i+1}^{t_i+2I_i} e^{\epsilon_t} \right)\right.\\&\left. - \min_{y\in\mathcal{Y}} \left\{ \frac{1}{2I_i} \sum_{t=t_i+1}^{t_i+2I_i} \left(h\big(y - e^{\alpha-\beta p+\epsilon_t}\big)^{+} + b\big(e^{\alpha-\beta p+\epsilon_t} - y\big)^{+} \right) \right\}\right\}.\end{aligned} \tag{12.6}$$

Compared with problem B1, it is seen that B2 is obtained from B1 after replacing the expectations in B1 by sample averages, hence B2 is the sample average approximation (SAA) of problem B1. Here ϵ_t, $t = t_i + 1, \ldots, t_i + 2I_i$, represent the realizations of random errors during stage i. Let $\tilde{p}_{i+1}\,(\alpha, \beta)$ denote the optimal price and $\tilde{y}_{i+1}(e^{\alpha-\beta p})$ the optimal order-up-to level for problem B2, which is the optimal solution for the inner minimization problem in (12.6).

The third bridging problem B3 is a variation of problem B2, which replaces the true random error ϵ_t by a biased error sample ζ_t, $t = t_i + 1, \ldots, t_i + 2I_i$. That is, for

$$\zeta_{t=t_i+1}^{t_1+I_i} = \big(\zeta_{t_i+1}, \ldots, \zeta_{t_i+I_i}\big), \quad \zeta_{t=t_i+I_i+1}^{t_1+2I_i} = \big(\zeta_{t_i+I_i+1}, \ldots, \zeta_{t_i+2I_i}\big),$$

and parameters α and $\beta > 0$, we define the third bridging problem B3 as

Bridging Problem B3 :

$$\max_{p\in\mathcal{P}} \breve{Q}_{i+1}\Big(p, e^{\alpha-\beta p}, \zeta_{t=t_i+1}^{t_1+I_i}, \zeta_{t=t_i+I_i+1}^{t_1+2I_i}\Big) := \max_{p\in\mathcal{P}} \left\{ pe^{\alpha-\beta p}\left(\frac{1}{2I_i}\sum_{t=t_i+1}^{t_i+2I_i} e^{\zeta_t}\right)\right.$$
$$\left. - \min_{y\in\mathcal{Y}}\left\{\frac{1}{2I_i}\sum_{t=t_i+1}^{t_i+2I_i}\Big(h\big(y - e^{\alpha-\beta p+\zeta_t}\big)^+ + b\big(e^{\alpha-\beta p+\zeta_t} - y\big)^+\Big)\right\}\right\}.$$

Note that when $(\alpha, \beta) = (\hat{\alpha}_{i+1}, \hat{\beta}_{i+1})$, and $\zeta_t = \eta_t$ for $t = t_1 + 1, \ldots, t_i + 2I_i$, problem B3 reduces to problem DD (12) in the DDA algorithm. Thus, problem B3 serves as a bridge between problem B2 and problem DD. We denote the optimal price of problem B3 by $\breve{p}_{i+1}\big((\alpha, \beta), \zeta_{t=t_i+1}^{t_1+I_i}, \zeta_{t=t_i+I_i+1}^{t_1+2I_i}\big)$ and its optimal order-up-to level, for given price p, by $\breve{y}_{i+1}\big(e^{\alpha-\beta p}, \zeta_{t=t_i+1}^{t_1+I_i}, \zeta_{t=t_i+I_i+1}^{t_1+2I_i}\big)$.

Based on their definitions, problem CI, bridging problems B1–B3, and problem DD require less and less information about the demand process. Problem CI has complete information about both $\lambda(\cdot)$ and the distribution of ϵ; problem B1 does not know $\lambda(\cdot)$ but knows the distribution of ϵ; problem B2 does not know either $\lambda(\cdot)$ or the distribution of ϵ but has access to true samples of ϵ; problems B3 and DD do not have true samples and have to use biased samples. Chen et al. (2019) prove convergence for each pair of adjacent problems, and eventually establish convergence of problem DD to problem CI.

12.3 Nonparametric Learning for Lost-Sales System

Different from Sect. 12.2 that considers backlogged demand, in this section we consider lost sales and censored demand. This scenario happens when, in case of a stockout, rejected customers leave the store without purchasing. These customers cannot be observed by the retailer, and demand data is thus truncated by inventory levels. We will discuss the model, algorithms, and regret convergence results based on Chen et al. (2021a, 2020b).

Consider the additive demand model $D_t(p_t) = \lambda(p_t) + \epsilon_t$ with $\lambda(\cdot)$ being a non-increasing deterministic function and ϵ_t, $t = 1, 2, \ldots, T$, being i.i.d. random variables with $\mathbb{E}[\epsilon_t] = 0$. We denote the CDF of ϵ_t by $F(\cdot)$, which is assumed to be continuous and differentiable, the PDF by $f(\cdot)$ such that $f(\epsilon_t) < \infty$ for any ϵ_t, and the standard deviation of ϵ_t by σ. For notational convenience, we use ϵ_t and ϵ interchangeably because of the i.i.d. assumption. Demands are satisfied as much as possible by on-hand inventory, and unsatisfied demands are lost and unobservable. For system dynamics one has $x_{t+1} = (y_t - D_t(p_t))^+$. The instantaneous reward for period t is $p_t\mathbb{E}[\min\{y_t, D_t(p_t)\}] - b\mathbb{E}[D_t(p_t) - y_t]^+ - h\mathbb{E}[y_t - D_t(p_t)]^+ = p_t\mathbb{E}[D_t(p_t)] - (b + p_t)\mathbb{E}[D_t(p_t) - y_t]^+ - h\mathbb{E}[y_t - D_t(p_t)]^+$.

The firm knows neither the function $\lambda(p_t)$ nor the distribution of the random term ϵ_t a priori, which must be learned from censored demands collected over time while maximizing the cumulative profit. In this system, demand is censored, therefore, $o_t = \min\{D_t(p_t), y_t\}$. For an admissible policy, (p_t, y_t) is adapted to the filtration generated by $\{(p_s, y_s), \min\{D_s(p_s), y_s\} : s = 1, \ldots, t-1\}$ under censored demand.

If the underlying demand-price function $\lambda(p)$ and the distribution of the error term ϵ_t were known a priori, the clairvoyant optimal policy for this problem is a myopic policy (refer to Sobel 1981). Define the single-period problem by

$$Q(p, y) = p\mathbb{E}[D_1(p)] - (b + p)\mathbb{E}[D_1(p) - y]^+ - h\mathbb{E}[y - D_1(p)]^+.$$

To find the optimal pricing and inventory decisions, it suffices to maximize the single-period revenue $Q(p, y)$, which can be expressed as

$$\begin{aligned}&\max_{p,y}\left\{p\mathbb{E}[D_1(p)] - (b + p)\mathbb{E}[D_1(p) - y]^+ - h\mathbb{E}[y - D_1(p)]^+\right\}\\ &\quad = \max_p\left\{p\lambda(p) - \min_y\left\{(b + p)\mathbb{E}\left[\lambda(p) + \epsilon - y\right]^+ + h\mathbb{E}\left[y - \lambda(p) - \epsilon\right]^+\right\}\right\}.\end{aligned}$$

Hence, we rewrite the clairvoyant problem as

$$\begin{aligned}&\max_{p,y} Q(p, y) = \max_p G(p),\\ &\text{where}\quad G(p) = p\lambda(p) - \min_y\left\{(b + p)\mathbb{E}\left[\lambda(p) + \epsilon - y\right]^+ + h\mathbb{E}\left[y - \lambda(p) - \epsilon\right]^+\right\}.\end{aligned}$$

This problem was first studied in Chen et al. (2021a), whose learning method and result will be briefly reviewed. Then we shift our focus to Chen et al. (2020b), which studies the problem in a more general setting and improves the convergence rate in Chen et al. (2021a).

12.3.1 Algorithms and Results in Chen et al. (2021a)

Chen et al. (2021a) assume $G(\cdot)$ is concave and $\lambda(p)$ is differentiable to a high order. They provide a spline approximation based learning algorithm (SALA) under an exploration-exploitation framework.

Algorithm for Concave $G(\cdot)$ The learning algorithm follows an exploration-exploitation framework and is based on spline approximation.

We now formally describe how a spline approximation for the demand-price function $\lambda(\cdot)$ is constructed. Before doing that, we first present a high-level view of the approximation method.

Spline approximation needs two integer inputs, $m > 0$ and $l > 0$, and it requires the specification of *knots, basis functions*, and *coefficients*. Knots, denoted as w_i, $i = 1, \ldots, 2m + l$, are equally spaced price points on the whole price interval, and there are in total $2m + l$ of them. The more knots a model has, the more observations of $\lambda(\cdot)$ the model uses to do estimation, which in general leads to a more accurate spline approximation. Let $\mathcal{L}\lambda(p)$ denote the spline approximation operator of a deterministic function $\lambda(p)$, and it can be represented as

$$\mathcal{L}\lambda(p) = \sum_{i=1}^{m+l} \gamma_i^{\lambda} \cdot N_i^m(p), \tag{12.7}$$

where $N_i^m(p)$, $i = 1, \ldots, m + l$, are the basis functions with coefficients γ_i^{λ}. The base function $N_i^m(p)$ is polynomial in p with the highest order $m - 1$ and is constructed based on knots $w_i, \ldots, w_{i+m}$. The larger the m, the smoother the $N_i^m(p)$ and $\mathcal{L}\lambda(p)$. The coefficient γ_i^{λ} is computed based on some specific price points on $[w_i, w_{i+m}]$ and the corresponding values of $\lambda(p)$ at these price points. To be more specific, price points used here include $w_i, \ldots, w_{i+m}$ and $\tau_{i1}, \ldots, \tau_{im}$ that will be defined shortly in Algorithm 2. The detailed procedure of spline approximation is also presented in Algorithm 2.

It follows from Schumaker (2007) that for the basis function $N_i^m(p)$, its $(m - 2)$-th order derivative exists and is continuous. Together with Theorem 4.9 in Schumaker (2007), one can verify that the basis function $N_i^m(p) = 0$ for $p \notin (w_i, w_{i+m})$ and $N_i^m(p) > 0$ for $p \in (w_i, w_{i+m})$.

Given the detailed construction of the spline approximation, we are ready to present the main learning algorithm termed SALA in Algorithm 3.

The learning algorithm SALA separates the planning horizon into a disjoint exploration phase and exploitation phase.

The algorithm specifies the parameters m and l for determining the density for spline approximation, the parameter Δ for determining the grid size for (sparse) discrete optimization problem, and the parameter L for determining the length of the exploration phase. Note that these parameters are determined "optimally" via (12.10) to minimize the theoretical regret rate.

Algorithm 2 Constructing a Spline Approximation (SA)

```
Let integers $m \geq 2$ and $l \geq 1$ be the inputs of a spline approximation. The (optimal) values of $m$ and $l$ will be specified later.
Let the set of $2m + l$ points $\{w_1, \ldots, w_{2m+l}\}$ be a partition of the interval
    $$\left[p^l - \frac{p^h - p^l}{l+1}(m-1),\ p^h + \frac{p^h - p^l}{l+1}(m-1)\right],$$
  where each point $w_i$ is defined by
    $$w_i = p^l + \frac{p^h - p^l}{l+1}(i - m), \qquad \text{for } i = 1, \ldots, 2m + l.$$
  Note that $w_m = p_l$ and $w_{m+l+1} = p_h$ and there are $l$ equally spaced points strictly between $p_l$ and $p_h$. Also, there are $m - 1$ extension points to the left of $p_l$ and $m - 1$ extension points to the right of $p_h$. Thus, there are in total $2m + l$ equally spaced points for the above specified interval.
for $i = 1, 2, \ldots, m + l$ do
    $\varphi_{im}(x) = \Pi_{r=1}^{m-1}(x - w_{i+r})$.
    for $j = 1, 2, \ldots, m$ do
        $\tau_{ij} = w_i + (w_{i+m} - w_i)\frac{j-1}{m-1}$.
        $\psi_{ij}(x) = \Pi_{r=1}^{j-1}(x - \tau_{ir})$,   with $\psi_{i1}(x) \equiv 1$.
        Then define
            $$\alpha_{ij} = \sum_{r=1}^{j} \frac{(-1)^{r-1}\varphi_{im}^{(m-r)}(0)\psi_{ij}^{(r-1)}(0)}{(m-1)!}. \qquad (12.8)$$
    end for
end for
Given a single variate real function $\lambda(\cdot)$ and a sequence of numbers $x_1 < x_2 < \cdots < x_{r+1}$, let $\mathcal{D}_{[x_1,\ldots,x_{r+1}]}\lambda$ be the operator that gives the $r$-th order divided difference of $\lambda(\cdot)$, defined by
    $$\mathcal{D}_{[x_1,\ldots,x_{r+1}]}\lambda = \sum_{j=1}^{r+1} \frac{\lambda(x_j)}{\Pi_{i=1,i\neq j}^{r+1}(x_j - x_i)},$$
  and if $r = 0$, $\mathcal{D}_{[x_1]}\lambda \equiv \lambda(x_1)$.
for $i = 1, \ldots, m + l$ do
    The *spline approximation coefficients* are
        $$\gamma_i^\lambda = \sum_{j=1}^{m} \alpha_{ij} \cdot \mathcal{D}_{[\tau_{i1},\ldots,\tau_{ij}]}\lambda.$$
    Moreover, for $p \in [p^l, p^h]$, define the $m$-th order *spline approximation basis functions* associated with knots $w_i, \ldots, w_{i+m}$ by
        $$N_i^m(p) = (-1)^m (w_{i+m} - w_i)\mathcal{D}_{[w_i,\ldots,w_{i+m}]}(\max\{0, p - w\})^{m-1}. \qquad (12.9)$$
    In $\mathcal{D}_{[w_i,\ldots,w_{i+m}]}(\max\{0, p - w\})^{m-1}$, the argument $(\max\{0, p - w\})^{m-1}$ is considered as a function of $w$ for given $p$, and the resulting basis function $N_i^m(p)$ is a function of $p$.
end for
The spline approximation of function $\lambda(p)$, denoted by $\mathcal{L}\lambda(p)$, is given by (12.7)
```

Algorithm 3 Spline Approximation Based Learning Algorithm (SALA)

```
Set input parameters
    $m = \max\left\{3, \lceil (\log T)^{\frac{1}{2}} \rceil\right\}, \ L = \left\lceil T^{\frac{1}{2}+\frac{1}{\sqrt[3]{\log T}}} \right\rceil, \ l = \left\lceil (\log T)^{\frac{3}{2}} T^{\frac{1}{4\sqrt{\log T}}} \right\rceil, \ \Delta = T^{-\frac{1}{4}}.$   (12.10)
    Define a sparse discretized set of prices by
    $\mathcal{S} = \{p^l, p^l + \Delta, p^l + 2\Delta, \ldots, p^h\},$   (12.11)
    which is the discrete search space for pricing decisions. We refer to $\mathcal{S}$ as the (sparse) grid.
for $i = 1, \ldots, l + m$ do
    for $j = 1, \ldots, l$ do
        for $t = (i-1)mL + (j-1)L + 1, \ldots, (i-1)mL + jL$ do
            Implement the following pricing and order-up-to decisions: $p_t = \tau_{ij}, y_t = \lceil \log L \log\log L \rceil$, where $\tau_{ij}$ is defined in Algorithm 2 spline approximation.
        end for
    end for
end for
for $i = 1, \ldots, l + m$ do
    for $j = 1, \ldots, l$ do
        Let the average empirical sales be $s_{ij} = \frac{\sum_{t=(i-1)mL+(j-1)L+1}^{(i-1)mL+jL} d_t \wedge y_t}{L}$.
    end for
    Let the empirical spline approximation coefficients be
        $\beta_i = \alpha_{i1} s_{i1} + \sum_{j=2}^{m} \sum_{v=1}^{j} \frac{\alpha_{ij} s_{iv}}{\Pi_{r=1, r \neq v}^{j} (\tau_{iv} - \tau_{ir})},$
    where $\alpha_{ij}$ is defined in (8).
end for
The spline approximation of function $\lambda(p)$ using sales (or censored demand) is then given by $\hat{\lambda}(p) = \sum_{i=1}^{m+l} \beta_i N_i^m(p)$, where the basis function $N_i^m(p)$ is defined in (13).
for $i = 1, \ldots, l + m$ do
    for $j = 1, \ldots, l$ do
        for $t = (i-1)mL + (j-1)L + 1, \ldots, (i-1)mL + jL$ do
            Let
                $\eta_t = d_t \wedge y_t - s_{ij}$   (12.12)
            be the *residual error*, which is used to approximate the random error (with some biases).
        end for
    end for
end for
Solve the following surrogate optimization problem on a sparse grid $\mathcal{S}$ (based on sales and spline approximation):
    $\max_{p,y} \hat{Q}(p, y) \triangleq \max_{p \in \mathcal{S}} \hat{G}(p),$ where
    $\hat{G}(p) \triangleq p\hat{\lambda}(p) - \min_{y} \left\{ (b + p) \frac{\sum_{t=1}^{L(m+l)m} [\hat{\lambda}(p) + \eta_t - y]^+}{L(m+l)m} + h \frac{\sum_{t=1}^{L(m+l)m} [y - \hat{\lambda}(p) - \eta_t]^+}{L(m+l)m} \right\}.$
    Let $(\hat{p}, \hat{y}) = \arg\max \hat{Q}(p, y)$.
for $t = L(m+l)m + 1, \ldots, T$ do
    Set the price and target inventory level to $p_t = \hat{p}, y_t = x_t \vee \hat{y}$.
end for
```

SALA then enters the exploration phase of total length of $L(m + l)m$ periods, which is roughly on the order of $\sqrt{T}$. The price space is discretized into equally spaced prices $\{\tau_{ij}\}$'s (which will also be used for constructing a spline approximation). For each i and j, SALA offers the price τ_{ij}, together with the pre-specified target inventory level y_t, for an equal number of periods. We note here that the high-level reason for the target inventory level y_t to be on the order of $\log L \log\log L$ is to ensure that the bias caused by demand censoring is appropriately bounded.

SALA leverages the sales collected over prices $\{\tau_{ij}\}$'s to carry out an empirical spline approximation $\hat{\lambda}(p)$ of the true demand-price function $\lambda(p)$. Also, SALA computes the so-called residual error η_t, which is used to approximate the random error ϵ_t. It is important to note that $\hat{\lambda}(p)$ is constructed based on sales (or censored demand) and, therefore, it suffers a bias in estimating $\lambda(p)$, which must be quantified in the regret analysis. Similarly, due to demand censoring, η_t is also a biased representation of ϵ_t, in which the bias must also be quantified.

SALA essentially treats the empirical spline approximation $\hat{\lambda}(p)$ as the true demand-price function $\lambda(p)$ and the residual error η_t as the true random error ϵ_t, and constructs the corresponding *sample average approximation* (SAA) based surrogate optimization problem. Note that the surrogate optimization problem is solved sequentially: the inner problem is to find the optimal inventory target level for a given price, while the outer problem is to find the optimal price on the grid. The inner problem is convex in the inventory target level, which can be efficiently solved using first-order methods, whereas the outer problem is a one-dimensional discretized problem but solved on a sparse grid.

Finally, SALA completes the exploration phase and enter the exploitation phase. For the remaining planning horizon, SALA implements the optimal price and target inventory level suggested by the (sampled) surrogate optimization problem. Note that the length of the exploitation phase is $T - L(m + l)m$, which is roughly on the order of $T - \sqrt{T}$.

Regret Convergence In Chen et al. (2021a), it shows that the convergence of the spline approximation can be bounded as $\mathbb{P}\Big\{\|\lambda'(p) - \hat{\lambda}'(p)\|_\infty \le C_2 T^{-1/4}\Big\} > 1 - T^{-2}$ and $\mathbb{P}\Big\{\|\lambda'(p) - \hat{\lambda}'(p)\|_\infty \le C_2 T^{-1/4}\Big\} > 1 - T^{-2}$ for some constant $C_2 > 0$ and any $p \in \mathcal{P}$. Moreover, the convergence of error estimation is shown as $\mathbb{P}\Bigg\{\Big|\mathbb{E}[\epsilon - z^*(p)]^+ - \frac{1}{L(m+l)m}\sum_{t=1}^{L(m+l)m}(\eta_t - \hat{z}(p))^+\Big| \le C_3 T^{-1/4}\Bigg\} > 1 - 10T^{-2}$, where $z^*(p) = F^{-1}\Big(\frac{b+p}{b+p+h}\Big)$ and $\hat{z}(p) = \min\Bigg\{\eta_j : \sum_{t=1}^{L(m+l)m} \mathbb{1}(\eta_t \le \eta_j) \ge \frac{b+p}{b+p+h}\Bigg\}$, for some constant $C_3 > 0$ and any $p \in \mathcal{P}$. Based on these results, the regret convergence rate of SALA is upper bounded as $R^{SALA}(T) \le C_4 T^{1/2+\varepsilon}(\log T)^3 \log\log T$, where $\varepsilon = 1/\sqrt[3]{\log T} + 0.25/\sqrt{\log T}$ and constant $C_4 > 0$. Here note that for any constant $c > 0$, one has $\log\log T/\log T < \varepsilon < c$ (or equivalently, $\log T < T^\varepsilon < T^c$), for large enough T. Since the regret lower bound for this problem is $\Omega(T^{1/2})$, the SALA algorithm matches the lower bound up to T^{ε}.

12.3.2 Algorithms and Results in Chen et al. (2020b)

Chen et al. (2020b) consider both concave and non-concave $G(\cdot)$, provide learning algorithms for the two scenarios, and show that the convergence rates of both algorithms match the theoretical lower bounds, respectively.

12.3.2.1 Concave $G(\cdot)$

In this section, we discuss the scenario with concave $G(\cdot)$.

Algorithm for Concave $G(\cdot)$ A different algorithm is proposed in Chen et al. (2020b) for concave $G(\cdot)$, which approaches the optimal y using bisection and optimal p using trisection. The detailed algorithm is presented in Algorithms 4, 5, and 6.

With the SEARCHORDERUPTO routine in Algorithm 4, for every price $p \in [\underline{p}, \overline{p}]$ one can estimate, using relatively few selling periods, the near-optimal order-up-to level $\hat{y}_n$ so that $Q(p, \hat{y}_n) \approx Q(p, y^*(p)) = G(p)$, where $y^*(p)$ is the optimal inventory level under price p. It is tempting to use a similar strategy on $G(\cdot)$ which is

Algorithm 4 Bisection search for order-up-to level y

```
function SEARCHORDERUPTO($p, n, C_1$)
    Initialize: $L_\tau = 0, U_\tau = \bar{y}, m_\tau = \bar{y}/2, \tau = 0, g_\tau = 0$;
    Offer the lowest price $\underline{p}$ until current inventory level is below $m_\tau$;*
    while $n$ review periods have not been reached do
        Set order-up-to level at $y_t = m_\tau$ and price at $p_t = p$;
        Observe censored demand and update $n_\tau \leftarrow n_\tau + 1$; $g_\tau \leftarrow g_\tau + (b + p)$ if no inventory is left; $g_\tau \leftarrow g_\tau - h$ if positive inventory is left;
        Construct confidence intervals $[\underline{g}(m_\tau), \overline{g}(m_\tau)] = \hat{g}_\tau \pm C_1/\sqrt{n_\tau}$, where $\hat{g}_\tau = g_\tau / n_\tau$;
        if $\tau < \lceil \log_2(n\bar{y}) \rceil$ and $\underline{g}(m_\tau) > 0$ then
            Update $L_{\tau+1} = m_\tau, U_{\tau+1} = U_\tau, m_{\tau+1} = (L_{\tau+1} + U_{\tau+1})/2, n_{\tau+1} = 0, \tau \leftarrow \tau + 1$;
            Offer the lowest price $\underline{p}$ until current inventory level is below $m_\tau$;*
        else if $\tau < \lceil \log_2(n\bar{y}) \rceil$ and $\overline{g}(m_\tau) < 0$ then
            Update $L_{\tau+1} = L_\tau, U_{\tau+1} = m_\tau, m_{\tau+1} = (L_{\tau+1} + U_{\tau+1})/2, n_{\tau+1} = 0, \tau \leftarrow \tau + 1$;
            Offer the lowest price $\underline{p}$ until current inventory level is below $m_\tau$;*
        end if
    end while
    Return $\hat{y}_n = m_\tau$ which is explored for the most number of times (largest $n_\tau$).
end function
```

* Review periods in these steps do *not* count towards the total budget of n periods.

Algorithm 5 Estimation of reward ($G(\cdot)$) differences at $p<p'$

```
function EstimateGDifference($p, \hat{y}, p', \hat{y}', n$)
    Set prices and order-up-to levels at $(p, \hat{y})$ for $n$ periods, and let $\{o_t = \min\{\lambda(p) + \varepsilon_t, \hat{y}\}\}_{t\in\mathcal{T}_1}$ be the censored demands, where $\mathcal{T}_1$ is the $n$ periods in this step;
    Set prices and order-up-to levels at $(p', \hat{y}')$ for the next $n$ periods, and let $\{o_t' = \min\{\lambda(p') + \varepsilon_t, \hat{y}'\}\}_{t\in\mathcal{T}_2}$ be the censored demands, where $\mathcal{T}_2$ is the $n$ periods in this step;
    Define $\delta_t := \hat{y} - o_t$, $\delta_t' := \hat{y}' - o_t'$ and let $\hat{\nu}, \hat{\nu}'$ be the empirical distributions of $\{\delta_t\}_{t\in\mathcal{T}_1}, \{\delta_t'\}_{t\in\mathcal{T}_2}$, respectively. Let $F_{\hat{\nu}}, F_{\hat{\nu}'}$ be the CDFs of $\hat{\nu}, \hat{\nu}'$. Find $\hat{u}$ such that
        $$\hat{u} := \sup\left\{u : F_{\hat{\nu}'}(u) \le \tfrac{h}{b+p+h}\right\};$$
    Return the estimate reward difference $\hat{\Delta}_G(p, p')$ as
        $$\hat{\Delta}_G(p, p') = \left[\frac{1}{n}\sum_{t\in\mathcal{T}_2} p'o_t' - h\delta_t'\right] - \left[\frac{1}{n}\sum_{t\in\mathcal{T}_1} po_t - h\delta_t\right] + b\left[\hat{u} \times \frac{h}{b+p+h} - \frac{1}{n}\sum_{t\in\mathcal{T}_2}\delta_t'\mathbf{1}\{0 < \delta_t' \le \hat{u}\}\right].$$
end function
```

Algorithm 6 The main algorithm: trisection search on prices

```
Input: time horizon $T$, price range $[\underline{p}, \overline{p}]$, parameters $C_1, C_2 > 0$.
Initialization: $\zeta = 0, L_\zeta = \underline{p}, U_\zeta = \overline{p}$.
while $T$ review periods have not been reached do
    Set $\alpha_\zeta = \frac{2}{3}L_\zeta + \frac{1}{3}U_\zeta, \beta_\zeta = \frac{1}{3}L_\zeta + \frac{2}{3}U_\zeta, N_\zeta = \lceil g(C_2/(\beta_\zeta - \alpha_\zeta)^4)\rceil$; **
    $\hat{y}_\zeta \leftarrow$ SearchOrderUpTo($\alpha_\zeta, N_\zeta, C_1$), $\hat{y}_\zeta' \leftarrow$ SearchOrderUpTo($\beta_\zeta, N_\zeta, C_1$);
    $\hat{\Delta}_G(\alpha_\zeta, \beta_\zeta) \leftarrow$ EstimateGDifference($\alpha_\zeta, \hat{y}_\zeta, \beta_\zeta, \hat{y}_\zeta', N_\zeta$);
    if $\hat{\Delta}_G(\alpha_\zeta, \beta_\zeta) > 0$ then
        Update $L_{\zeta+1} \leftarrow \alpha_\zeta, U_{\zeta+1} \leftarrow U_\zeta, \zeta \leftarrow \zeta + 1$;
    else
        Update $L_{\zeta+1} \leftarrow L_\zeta, U_{\zeta+1} \leftarrow \beta_\zeta, \zeta \leftarrow \zeta + 1$;
    end if
end while
```

** We use $g(x) := (x + \lceil\log_2(x\bar{y})\rceil)\lceil\log_2(x + \lceil\log_2(x\bar{y})\rceil)\rceil$.

strongly concave to the price to find the optimal price p^*, which has been applied to pure pricing without inventory replenishment problems in the literature (Wang et al., 2014; Lei et al., 2014). Such an approach, however, encounters a major technical hurdle that neither the reward $G(\cdot)$ nor its derivative can be directly observed or even accurately estimated, due to the censoring of the demands and the lost-sales component in the objective function.

In this section we present the key idea of this paper that overcomes this significant technical hurdle. The important observation is that, in a bisection or trisection search method, it is *not* necessary to estimate $G(p)$ accurately. Instead, one only needs to accurately estimate the *difference* of rewards $G(p') - G(p)$ at two prices p, p' in order to decide on how to progress, which can be accurately estimated even in the presence of censored demands and lost sales. We sketch and summarize this idea below.

The Key Idea of Algorithm 5—"Difference Estimator" Let $p < p'$ be two different prices and recall the definition that $G(p) = p\mathbb{E}[\min\{\lambda(p) + \varepsilon, y^*(p)\}] - b\mathbb{E}[(\varepsilon + \lambda(p) - y^*(p))^+] - h\mathbb{E}[(y^*(p) - \lambda(p) - \varepsilon)^+]$. When $y^*(p)$ is relatively accurately estimated (from the previous section and Algorithm 4), the only term in $G(p)$ that cannot be directly observed without bias is the lost-sales penalty $-b\mathbb{E}[(\varepsilon + \lambda(p) - y^*(p))^+]$. Hence, to estimate $G(p') - G(p)$ accurately (Chen et al., 2020b) only need to estimate the difference

$$\mathbb{E}[(\varepsilon + \lambda(p) - y^*(p))^+] - \mathbb{E}[(\varepsilon + \lambda(p') - y^*(p'))^+]. \tag{12.13}$$

By the property of newsvendor solution, $y^*(p) = \lambda(p)+z_p$ where z_p is such that $F_\mu(z_p) = \int_{-\infty}^{z_p} f_\mu(u)\mathrm{d}u = \phi(p) = \frac{b+p}{b+p+h}$, and similarly $y^*(p') = \lambda(p') + z_{p'}$ such that $F_\mu(z_{p'}) = \phi(p') = \frac{b+p'}{b+p'+h}$. Since $p < p'$, we have $z_p < z_{p'}$. Equation (12.13) can be subsequently simplified to

$$\begin{aligned}\mathbb{E}[(\varepsilon - z_p)^+] - \mathbb{E}[(\varepsilon - z_{p'})^+] &= \mathbb{E}[(\varepsilon - z_p)^+ - (\varepsilon - z_{p'})^+] \\ &= \underbrace{(z_{p'} - z_p) \times \Pr[\varepsilon \geq z_p]}_{\text{Part A}} - \underbrace{\mathbb{E}[(z_{p'} - \varepsilon)\mathbf{1}\{z_p \leq \varepsilon \leq z_{p'}\}]}_{\text{Part B}}.\end{aligned} \tag{12.14}$$

For Part A of Eq. (12.14), the $\Pr[\varepsilon \geq z_p]$ term has the closed-form, known formula of $\Pr[\varepsilon \geq z_p] = 1 - F_\mu(z_p) = 1 - \phi(p) = \frac{h}{b+p+h}$. To estimate $z_{p'} - z_p$, which is nonnegative, (Chen et al., 2020b) use the following observation:

$$1 - \phi(p) = \frac{h}{b + p + h} = \Pr[\varepsilon \geq z_p] \overset{(*)}{=} \Pr[(z_{p'} - \varepsilon)^+ \leq z_{p'} - z_p]. \tag{12.15}$$

Here the crucial Eq. (*) holds because $z_{p'} > z_p$, and, therefore, the event $\varepsilon \geq z_p$ is equivalent to either $\varepsilon > z_{p'}$ (for which $(z_{p'} - \varepsilon)^+$ is zero), or $\varepsilon \leq z_{p'}$ and $z_{p'} - \varepsilon \leq z_{p'} - z_p$. Furthermore, the random variable $(z_{p'} - \varepsilon)^+ = (y^*(p') - \lambda(p') - \varepsilon)^+$ is (approximately) *observable* when $y^*(p')$ is estimated accurately, because this is the leftover inventory at ordering-up-to level $y^*(p')$ and posted price p'. Therefore, one can collect samples of $(z_{p'} - \varepsilon)^+$, construct an empirical cumulative distribution function (CDF) and infer the value of $z_{p'} - z_p$ by inverting the empirical CDF at $h/(b + p + h)$. A similar approach can be taken to estimate Part B of Eq. (12.14), by plugging in the empirical distribution of the random variable $(z_{p'} - \varepsilon)^+\mathbf{1}\{0 \leq (z_{p'} - \varepsilon)^+ \leq z_{p'} - z_p\}$.

A pseudo-code description of the reward difference estimation routine is given in Algorithm 5. The design of Algorithm 5 roughly follows the key ideas demonstrated in the previous paragraph. The o_t and δ_t random variables correspond to the censored demand and the leftover inventory at time period t, and the distribution of δ_t (or δ_t') would be close to the distribution of $(z_p - \varepsilon)^+$ (or $(z_{p'} - \varepsilon)^+$). Using the observation in Eq. (12.15), $\hat{u}$ in Algorithm 5 would be a good estimate of $z_{p'} - z_p$ by inverting the empirical CDFs.

As the last component and the main entry point of the algorithm framework, (Chen et al., 2020b) describe a trisection search method to localize the optimal price p^* that maximizes $G(\cdot)$, based on the strong concavity of $G(\cdot)$ in p that is assumed for this scenario. The trisection principle for concave functions itself is not a new idea and has been explored in the existing literature on pure pricing without inventory replenishment problems (Lei et al., 2014; Wang et al., 2014). A significant difference, nevertheless, is that in this application the expected reward function $G(\cdot)$ cannot be observed directly (even up to centered additive noise) due to the presence of censored demands, and one must rely on the procedure described in the previous section to estimate the reward difference function $\Delta_G(\cdot, \cdot)$ instead. Below we describe the key idea for this component.

The Key Idea of Algorithm 6 Recall that $G(p) = \max_{y \in [0, \bar{y}]} Q(p, y)$ and $\Delta_G(p, p') = G(p') - G(p)$. A trisection search algorithm is used to locate $p^* \in [\underline{p}, \overline{p}]$ that maximizes $G(\cdot)$, under the assumption that $G(\cdot)$ is twice continuously differentiable and strongly concave in p. The algorithm starts with $I_0 = [\underline{p}, \overline{p}]$ and attempts to shorten the interval by $2/3$ after each epoch ζ, without throwing away the optimal price p^* with high probability. Suppose at epoch ζ the interval $I_\zeta = [L_\zeta, U_\zeta]$ includes p^*, and let α_ζ, β_ζ be the trisection points of I_ζ. Depending on the location of p^* relative to α_ζ, β_ζ, the updated, shrunk interval $I_{\zeta+1} = [L_{\zeta+1}, U_{\zeta+1}]$ can be computed. The above discussion shows that trisection search updates can be carried out by simply determining the signs of $\Delta_G(\alpha_\zeta, \beta_\zeta)$. A complete pseudo-code description of the procedure is given in Algorithm 6.

Regret Convergence for Concave $G(\cdot)$ The regret rate of the algorithm for concave $G(\cdot)$ is upper bounded as $R(T) \leq O\left(T^{1/2} (\ln T)^2\right)$ with probability $1 - O(T^{-1})$. This upper bound almost matches the theoretical lower bound of $\Omega(T^{1/2})$.

12.3.2.2 Non-Concave $G(\cdot)$

In this section, we discuss the scenario with non-concave $G(\cdot)$.

Algorithm for Non-Concave $G(\cdot)$ For non-concave $G(\cdot)$, (Chen et al., 2020b) still rely on bisection to search for the optimal y, but for p, the previous trisection framework cannot be applied anymore due to loss of concavity. They design an active tournament algorithm based on the difference estimator to search for the optimal p.

Key idea 1: discretization. The price interval $[\underline{p}, \overline{p}]$ is first being partitioned into J evenly spaced points $\{p(j)\}_{j \in [J]}$, with $J = \lceil T^{1/5} \rceil$. Because $G(\cdot)$ is twice continuously differentiable (implied by the first condition in Chen et al. (2020b)) and $p^* \in (\underline{p}, \overline{p})$, there exists p_{j^*} for some $j^* \in [J]$ such that $G(p^*) - G(p_{j^*}) \leq O(|p^* - p_{j^*}|^2) \leq O(J^{-2}) = O(T^{-2/5})$, because $G'(p^*) = 0$. The problem then reduces to a multiarmed bandit problem over the J arms of $\{p_j\}_{j\in[J]}$, with the important difference of the actual reward of each arm *not* directly observable due to the censored demands.

Key idea 2: active elimination with tournaments. With the sub-routines developed in Algorithms 4 and 5 in the previous section, we can in principle estimate the reward difference $\Delta_G(p, p')$ at two prices $p < p'$ up to an error on the order of $\widetilde{O}(1/\sqrt{n})$, with $\approx 2n$ review periods for each price and without incurring large regret. In Algorithm 6, we successfully applied this "pairwise comparison" oracle in a trisection approach to utilize the concavity of $G(\cdot)$. Without concavity of $G(\cdot)$, we are going to use an active elimination with tournaments approach to find the price with the highest rewards in $\{p_j\}_{j\in[J]}$.

More specifically, consider epochs $\gamma = 1, 2, \cdots$ with geometrically increasing sample sizes n_γ implied by geometrically decreasing accuracy levels $\Delta_\gamma = 2^{-\gamma}$. At the beginning of each epoch γ, the algorithm maintains an "active set" $\mathcal{S}_\gamma \subseteq [J]$ of prices such that for all $p \in \mathcal{S}_\gamma$, $G(p_{j^*}) - G(p) \leq \Delta_\gamma$ where $\Delta_\gamma = \widetilde{O}(1/\sqrt{n_\gamma})$. Chen et al. (2020b) use a "tournament" approach to eliminate prices in $\mathcal{S}_\gamma$ that have large sub-optimality gaps. In particular, all prices in $\mathcal{S}_\gamma$ are formed into pairs and each pair is allocated n_γ samples to either eliminate the inferior price in the pair, or to combine both prices into one and advance to the next round of the tournament. The tournament ends once there is only one price left, $\hat{p}_\gamma$. Afterwards a separate elimination procedure is invoked to retain all other prices that are close to $\hat{p}_\gamma$ in terms of performance. A detailed algorithm for non-concave $G(\cdot)$ is presented in Algorithm 7.

Regret Convergence for Non-Concave $G(\cdot)$ The regret convergence rate for non-concave $G(\cdot)$ is upper bounded as $R(T) \leq O\left(T^{3/5}(\ln T)^2\right)$ with probability $1 - O(T^{-1})$. Chen et al. (2020b) then prove the lower bound for non-concave $G(\cdot)$ and show that the upper bound matches the lower bound. They prove that there exist a problem instance such that for any learning-while-doing policy π and the sequential decisions $\{p_t, y_t\}_{t=1}^T$ the policy π produces, it holds for sufficiently large T that $\sup_\lambda \mathbb{E}\left[V^* - \sum_{t=1}^T Q(p_t, y_t)\right] \geq C_5 \times T^{3/5}/\ln T$ for some constant $C_5 > 0$. The lower bound is established by a novel information-theoretical argument based on generalized squared Hellinger distance, which is significantly different from conventional arguments that are based on Kullback–Leibler divergence.

12.4 Parametric Learning with Limited Price Changes

Models discussed in Sects. 12.2 and 12.3 assume that price can be adjusted at the beginning of every period. In practice, however, retailers may hesitate changing prices too frequently. Cheung et al. (2017) discussed several practical reasons for not

Algorithm 7 A discretization + tournament approach with non-concave $G(\cdot)$

```
Input: time horizon $T$, discretization parameter $J$, parameters $C_1, C_3 > 0$;
Let $\{p_j\}_{j=1}^{J}$ be $J$ prices that evenly partition $[\underline{p}, \overline{p}]$; $\mathcal{S}_0 = [J]$;
for $\gamma = 0, 1, 2, \cdots$ until $T$ review periods are reached do
    $\Delta_\gamma \leftarrow 2^{-\gamma}$, $n_\gamma \leftarrow \lceil g(C_3/\Delta_\gamma^2) \rceil^{***}$, $\mathcal{V}_{\gamma,0} \leftarrow \mathcal{S}_\gamma$, $\ell \leftarrow 0$;        ▷ the tournament phase
    while $|\mathcal{V}_{\gamma,\ell}| > 1$ do
        Group prices in $\mathcal{V}_{\gamma,\ell}$ into pairs;
        If $|\mathcal{V}_{\gamma,\ell}|$ is odd then transfer one arbitrary price to form $\mathcal{V}_{\gamma,\ell+1}$; else set $\mathcal{V}_{\gamma,\ell+1} = \emptyset$;
        for each pair of prices $p$, $p'$ in $\mathcal{V}_{\gamma,\ell}$ do
            $\hat{y} \leftarrow$ SEARCHORDERUPTO$(p, n_\gamma, C_1)$, $\hat{y}' \leftarrow$ SEARCHORDERUPTO$(p', n_\gamma, C_1)$;
            $\hat{\Delta}_G(p, p') \leftarrow$ ESTIMATEGDIFFERENCE$(p, \hat{y}, p', \hat{y}', n_\gamma)$;
            Update $\mathcal{V}_{\gamma,\ell+1} \leftarrow \mathcal{V}_{\gamma,\ell+1} \cup \{p'\}$ if $\hat{\Delta}_G(p, p') > 0$ and $\mathcal{V}_{\gamma,\ell+1} \leftarrow \mathcal{V}_{\gamma,\ell+1} \cup \{p\}$ otherwise;
        end for
        $\ell \leftarrow \ell + 1$;
    end while
    Obtain $\hat{p}_\gamma$ as the only price in $\mathcal{V}_{\gamma,\ell}$ and initialize $\mathcal{S}_{\gamma+1} \leftarrow \emptyset$;
                                                                        ▷ the elimination phase
    for each $p \in \mathcal{S}_\gamma$ do
        $\hat{y}_1 \leftarrow$ SEARCHORDERUPTO$(\hat{p}_\gamma, n_\gamma, C_1)$, $\hat{y}_2 \leftarrow$ SEARCHORDERUPTO$(p, n_\gamma, C_1)$;
        $\hat{\Delta}_G(\hat{p}_\gamma, p) \leftarrow$ EstimateGDifference$(\hat{p}_\gamma, p)$;
        If $\hat{\Delta}_G(\hat{p}_\gamma, p) \geq -\Delta_\gamma$ then update $\mathcal{S}_{\gamma+1} \leftarrow \mathcal{S}_{\gamma+1} \cup \{p\}$;
    end for
end for
```

*** Recall that we use $g(x) := (x + \lceil \log_2(x\bar{y}) \rceil) \lceil \log_2(x + \lceil \log_2(x\bar{y}) \rceil) \rceil$.

allowing frequent price changes, including customers' negative responses (e.g., that may cause confusion and affect the seller's brand reputation) and the cost associated with such changes (e.g., due to changing price labels in brick-and-mortar stores, etc.). In this section, we introduce a constraint that only allows the retailer to change prices no more than a certain number of times. Clearly, such a constraint limits the firm's ability to learn demand.

Demand in period t, $t \in \{1, 2, \ldots, T\}$, is random and depends on the selling price p_t, and its distribution function belongs to some family parameterized by $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^k$, $k \geq 1$, where $\mathcal{Z}$ is a compact and convex set. Let $D_t(p_t, \mathbf{z})$ be the demand in period t with probability mass function $f(\cdot; p_t, \mathbf{z})$, cumulative distribution function $F(\cdot; p_t, \mathbf{z})$, and support $\{d^l, d^l + 1, \ldots, d^h\}$ with d^l being a nonnegative integer and $d^h \leq +\infty$, and let d_t denote the realization of $D_t(p_t, \mathbf{z})$. The firm knows $f(\cdot; p_t, \mathbf{z})$ up to the parameter vector $\mathbf{z}$, which has to be learned from sales data.

Chen and Chao (2019) consider the backlog system and (Chen et al., 2020a) consider the lost-sales system with censored demand. This section will be mainly devoted to discussing algorithms and results in Chen et al. (2020a), where the firm can only observe sales data but not the actual demand when stockout occurs. Therefore, $o_t = \min\{D_t(p_t, \mathbf{z}), y_t\}$, and (p_t, y_t) is adapted to the filtration generated by $\{(p_s, y_s), o_s : s = 1, \ldots, t-1\}$ under censored demand. Let $p_t \in \mathcal{P} = [p^l, p^h]$ and $y_t \in \mathcal{Y} = \{y^l, y^l + 1, \ldots, y^h\}$, where the bounds of support $0 \leq p^l \leq p^h <$

$+\infty$ and $0 \le y^l \le y^h < +\infty$ are known. Assume for any $p_t \in \mathcal{P}$ it holds that $\mathbb{E}[D_t(p_t, \mathbf{z})] > 0$. The state transition is $x_{t+1} = (y_t - D_t(p_t, \mathbf{z}))^+$.

The expected total profit over the planning horizon, given an admissible policy $\phi = ((p_1, y_1), (p_2, y_2), \ldots, (p_T, y_T))$, is

$$V^{\phi}(T, \mathbf{z}) = \sum_{t=1}^{T} \Big\{ p_t \mathbb{E}[\min\{D_t(p_t, \mathbf{z}), y_t\}] - \left\{ h\mathbb{E}\left[y_t - D_t(p_t, \mathbf{z})\right]^+ + b\mathbb{E}\left[D_t(p_t, \mathbf{z}) - y_t\right]^+ \right\} \Big\} \tag{12.16}$$

and the prices need to satisfy the *limited price change constraint* for some given integer $m \ge 1$:

$$\sum_{t=1}^{T-1} \mathbf{1}(p_t \ne p_{t+1}) \le m, \tag{12.17}$$

where $\mathbf{1}(A)$ is the indicator function taking value 1 if statement A is true and 0 otherwise.

The single-period objective function is

$$G(p, y, \mathbf{z}) = p\mathbb{E}[D(p, \mathbf{z})] - h\mathbb{E}\left[y - D(p, \mathbf{z})\right]^+ - (b + p)\mathbb{E}\left[D(p, \mathbf{z}) - y\right]^+, \tag{12.18}$$

where $D(p, \mathbf{z})$ is a generic random demand when the true parameter is $\mathbf{z}$ and the price is $p \in \mathcal{P}$. For the underlying system parameter vector $\mathbf{z}$, let (p^*, y^*) be a maximizer of $G(p, y, \mathbf{z})$. If $\mathbf{z}$ is known, then the firm could set (p^*, y^*) every period without changing the price, and this is the clairvoyant solution, for which the T-period total profit is denoted as V^*.

Demand models are categorized into two groups, (1) the well-separated case and (2) the general case. Two probability mass functions are said to be identifiable if they are not identically the same.

12.4.1 Well-Separated Demand

The family of distributions $\{f(\cdot; p, z) : z \in \mathcal{Z}\}$ is called well-separated if for any $p \in \mathcal{P}$, the class of probability mass functions $\{f(\cdot; p, z) : z \in \mathcal{Z}\}$ is identifiable, i.e., $f(\cdot; p, z_1) \ne f(\cdot; p, z_2)$ for $z_1 \ne z_2 \in \mathcal{Z}$.

If a family of distributions is well-separated, then no matter what selling price p is charged, the sales data will allow the firm to learn about the parameter z. This shows that, in the well-separated case, pricing exploration can be a side benefit from exploitation, thus no active pricing exploration is necessary.

Algorithm 8 *m* price changes for the well-separated case

```
Input $\hat{p}_1, \hat{y}_1$.
Let $I_i = \lceil T^{i/(m+1)} \rceil$, for $i = 1, \ldots, m$, and $I_{m+1} = T - \sum_{i=1}^{m} I_i$. Let $t_1 = 0$, and $t_i = \sum_{j=1}^{i-1} I_j$ for $i = 2, \ldots, m+2$.
for stage $i \le m+1$ do
    Set
        $\tilde{y}_i = \begin{cases} \hat{y}_i, & \text{if } \hat{y}_i > d^l, \\ \min\{\max\{\hat{y}_i + \Delta, y^l\},\ y^h\}, & \text{if } \hat{y}_i = d^l. \end{cases}$
    for $t = t_i + 1, \ldots, t_{i+1}$ do
        $p_t = \hat{p}_i$, $y_t = \max\{x_t, \tilde{y}_i\}$, $x_{t+1} = \max\{y_t - d_t, 0\}$.
    end for
    Compute the MLE estimator for $z$ by
        $\hat{z}_i = \arg\max_{z \in \mathcal{Z}} \Bigg\{ \sum_{\{t \in \{t_i+1, \ldots, t_{i+1}\}: d_t < y_t\}} \log f(d_t; \hat{p}_i, z) + \sum_{\{t \in \{t_i+1, \ldots, t_{i+1}\}: d_t \ge y_t\}} \log\Big(1 - F(y_t - 1; \hat{p}_i, z)\Big) \Bigg\}.   (12.19)$
    Solve the data-driven optimization problem
        $(\hat{p}_{i+1}, \hat{y}_{i+1}) = \arg\max_{(p,y) \in \mathcal{P} \times \mathcal{Y}} G(p, y, \hat{z}_i).   (12.20)$
end for
```

Chen et al. (2020a) consider two scenarios of limited-price constraint for well-separated demand. The first scenario is that the number of price changes is restricted to be no more than a given integer $m \ge 1$ that is independent of the length of planning horizon T, while for the second scenario, the number of allowed price changes is at most $\beta \log T$ for the T-period problem for some constant $\beta > 0$.

Algorithm for *m* Price Changes Under Well-Separated Demand The main idea of the algorithm is to estimate the known parameter z by maximum likelihood estimation based on censored demand. The detailed algorithm is presented in Algorithm 8.

As shown in Algorithm 8, exploration in the inventory space is needed. If $\hat{y}_i$ equals d^l, then implementing $\hat{y}_i$ will not yield any information about the demand. Hence the algorithm imposes $\tilde{y}_i = \hat{y}_i + \Delta$, which ensures to reveal some demand information with a positive probability. Then the algorithm constructs an MLE estimator using censored data, $\min\{d_t, y_t\}$, which are neither independent nor identically distributed. This is because, inventory level y_t depends on carryover inventory x_t that is a function of earlier inventory level and demand, and earlier demand depends on the pricing decisions. Assumption 1(i) in the paper guarantees that, with a high probability (its complement has a probability decaying exponentially fast in

I_i), the objective function in (12.19) is strictly concave, thus there exists a unique global maximizer.

Regret Convergence for m Price Changes Under Well-Separated Demand Chen et al. (2020a) provide both regret upper and lower bounds for well-separated demand with m price changes. The regret upper bound is $R(T) \leq C_6\ T^{\frac{1}{m+1}}$ for some constant $C_6 > 0$. The lower bound is provided as following. There exist problem instances such that the regret for any admissible learning algorithm that changes price at most m times is lower bounded by $R(T) \geq C_7\ T^{\frac{1}{m+1}}$ for some constant $C_7 > 0$ and large enough T.

One fundamental challenge to prove this lower bound is that the times of price changes are dynamically determined, i.e., they are increasing random stopping times. An adversarial parameter class is constructed, among which a policy needs to identify the true parameter. The parameter class is constructed in a hierarchical manner such that when going further down the hierarchy the parameters are harder to distinguish. A delicate information-theoretical argument is employed to prove the lower bound. Here we only illustrate the high-level idea using a special case $m = 2$.

Chen et al. (2020a) construct a problem instance in which the inventory order-up-to level for each period is fixed and high enough so that any realization of the demand can be satisfied under any price. Therefore, the effect of lost sales and censored data is eliminated and the original joint pricing and inventory control problem is reduced to a dynamic pricing problem with fixed inventory control strategies. Suppose the demand follows a Bernoulli distribution with a single unknown parameter $z \in [0, 1]$.

Let (p_0, p_1, p_2) be the $m + 1 = 3$ different prices of a policy π, (T_0, T_1, T_2) be the number of time periods each price is committed to, with $T_2 = T - T_0 - T_1$. The paper constructs an adversarial parameter class consisting of $2^{m+1} = 8$ parameters, among which policy π needs to identify the true parameter. These parameters are constructed in a hierarchical way. The 8 parameters are first partitioned into two 4-parameter groups, with the parameters in each group being close to each other, and the two groups are about $1/4$ apart. Each 4-parameter group can then be divided into two 2-parameter groups, with a distance of $T^{-1/6}$ between them. Within each 2-parameter group, the two parameters are $T^{-1/3}$ apart. A policy needs to work down the hierarchy levels to locate the true parameter, and the further it works down, the harder to differentiate between groups/parameters.

The proof first shows the tradeoff in deciding (p_0, T_0) at the first hierarchy level. Assume without loss of generality that z resides in the first branch of the tree. Because policy π does not have any observations when deciding p_0, there is a constant probability that p_0 is selected to favor the other branch. This high risk yields that T_0 cannot be longer than $O(T^{1/3})$, because otherwise the regret accumulated during T_0 would immediately imply an $\Omega(T^{1/3})$ regret.

If T_0 is upper bounded by $O(T^{1/3})$, the tradeoff in deciding (p_1, T_1) is as follows. With so few demand observations during T_0, policy π will *not* be able to distinguish groups on the second level. Therefore, assuming the true z resides in the first group, it can (and will) be shown that p_1 is selected to favor the wrong (second) group with a constant probability. Given this risk and that the parameters between the first and second groups are distanced at $T^{-1/6}$, T_1 cannot be longer than $O(T^{2/3})$ to yield an $\Omega(T^{1/3})$ regret. The same argument then carries over to the third level when deciding p_2. After summing up the regrets from all the three levels, it is shown that the total regret of policy π cannot be better than $\Omega(T^{1/3})$.

In making real decisions it may happen that T is not clearly specified at the beginning. The firm requires that the price change be not too often, but it usually allows more price changes for longer planning horizon. Chen et al. (2020a) propose a learning algorithm where the number of price changes is restricted to $\beta \log T$ for some constant $\beta > 0$.

Algorithm for $\beta \log T$ Price Changes Under Well-Separated Demand The algorithm runs very similarly to the one for m price changes, except that now the number of periods in i is given by $I_i = \lceil I_0 v^i \rceil$, $i = 1, 2 \ldots, N$, and there is a total of $N = O(\log T)$ iterations.

Regret Convergence for $\beta \log T$ Price Changes Under Well-Separated Demand The regret convergence rate for the algorithm with less than $\beta \log T$ price changes is upper bounded as $R(T) \leq C_8 \log T$, for a constant $C_8 > 0$ and large enough T. The lower bound is also provided. There exist problem instances such that the regret for any learning algorithm satisfies $R(T) \geq C_9 \log T$ for some constant $C_9 > 0$ and $T \geq 1$.

12.4.2 General Demand

Now we consider the more general case that the parameters in probability mass function $f(\cdot; p, \mathbf{z})$ is a k-dimensional vector, i.e., $\mathbf{z} = (z_1, \ldots, z_k) \in \mathcal{Z} \subset \mathbb{R}^k$ for some integer $k \geq 1$. For a set of given prices $\mathbf{p} = (p_1, \ldots, p_k) \in \mathcal{P}^k$, and correspondingly realized demands $\mathbf{d} = (d_1, \ldots, d_k) \in \{d^l, d^l + 1, \ldots, d^h\}^k$, define

$$Q^{\mathbf{p},\mathbf{z}}(\mathbf{d}) = \prod_{j=1}^{k} f(d_j; p_j, \mathbf{z}).$$

The family of distributions $\{Q^{\mathbf{p},\mathbf{z}}(\cdot) : \mathbf{z} \in \mathcal{Z}\}$ is said to belong to the general case if there exist k price points $\bar{\mathbf{p}} = (\bar{p}_1, \ldots, \bar{p}_k) \in \mathcal{P}^k$ such that the family of distributions $\{Q^{\bar{\mathbf{p}},\mathbf{z}}(\cdot) : \mathbf{z} \in \mathcal{Z}\}$ is identifiable, i.e., $Q^{\bar{\mathbf{p}},\mathbf{z}_1}(\cdot) \neq Q^{\bar{\mathbf{p}},\mathbf{z}_2}(\cdot)$ for any $\mathbf{z}_1 \neq \mathbf{z}_2$ in $\mathcal{Z}$.

Suppose we are allowed to make up to m price changes during the planning horizon. We consider the case of $m \geq k$ in this section, as in the case of $m < k$ no algorithm will be able to identify the k unknown parameters and, therefore, the regret would be linear in T.

Algorithm for General Demand The algorithm follows an exploration-exploitation framework, and the unknown parameter vector $\mathbf{z}$ is estimated by MLE. Detailed algorithm is presented in Algorithm 9.

Algorithm 9 $m \geq k$ price changes for the general case

```
Input $\bar{y} \in \mathcal{Y}$ for the initial inventory order-up-to level, and constant $s > 0$.
Let $I = \lceil T^{1/2}/k \rceil$.
for $i = 1, \cdots, k$ do
    for $t = (i-1)I + 1, \ldots, iI$ do
        Set $p_t = \bar{p}_i$.
    end for
    For $t = (i-1)I + 1$, set $y_t = \max\{x_t, \bar{y}\}$, thus $x_{t+1} = \max\{y_t - d_t, 0\}$;
    for $t = (i-1)I + 2, \ldots, iI$ do
        Set
            $y_t = \begin{cases} y_{t-1}, & \text{if } d_{t-1} < y_{t-1}; \\ \min\left\{(1+s)y_{t-1}, \lceil \log T \rceil\right\}, & \text{otherwise.} \end{cases}$
            $x_{t+1} = \max\{y_t - d_t, 0\}.$
    end for
end for
Estimate $\mathbf{z}$ by the MLE estimator
    $\hat{\mathbf{z}} = \arg\max_{\mathbf{z} \in \mathcal{Z}} \left\{ \sum_{\{t \in \{1,\ldots,kI\}: y_t > d_t\}} \log f(d_t; p_t, \mathbf{z}) + \sum_{\{t \in \{1,\ldots,kI\}: y_t \leq d_t\}} \log\Big(1 - F(y_t - 1; p_t, \mathbf{z})\Big) \right\}.$   (12.21)
Solve the data-driven optimization problem $(\hat{p}, \hat{y}) = \max_{(p,y) \in \mathcal{P} \times \mathcal{Y}} G(p, y, \hat{\mathbf{z}})$.
for $t = kI + 1, \ldots, T$ do
    $p_t = \hat{p}$,    $y_t = \max\{x_t, \hat{y}\}$, and $x_{t+1} = \max\{0, y_t - d_t\}$.
end for
```

As shown in Algorithm 9, during the exploration phase, Algorithm-II experiments with k prices (thus $k - 1$ price changes). Because of censored data, the true demand realizations exceeding inventory level cannot be observed. To make sure to receive sufficient demand data, every time a stockout occurs, the algorithm increases the order-up-to level by a certainty percentage. Because d^h may be infinity, this does not mean that the data censoring issue will be totally resolved, but with high probability. In the MLE step, the sales data $\min\{d_t, y_t\}$ are correlated and non-identically distributed, because inventory levels y_t are dependent through the "raising inventory" decisions as well as the carryover inventories. Propositions in Chen et al. (2020a) state that, despite the dependent data, the MLE possesses the desired property. The empirical optimal solution is implemented for the rest of the planning horizon, resulting in k price changes.

Regret Convergence for General Demand Chen et al. (2020a) provide the regret upper bounded for the general demand case as follows: if the demand is unbounded $d^h = +\infty$, then the regret for general demands is upper bounded by $R(T) \leq C_{10} T^{1/2} \log T$; if the demand is bounded $d^h < +\infty$, then the regret for general demands is upper bounded by $R(T) \leq C_{10} T^{1/2}$, for some constant $C_{10} > 0$. The theoretical lower bound for this problem is $\Omega(T^{1/2})$, which is established in Broder and Rusmevichientong (2012) for a dynamic pricing problem with infinite initial inventory.

12.5 Backlog System with Fixed Ordering Cost

In this section, we consider the presence of fixed ordering cost, which is a fixed cost that is incurred by the firm whenever a positive amount of inventory is ordered.

Demand is modeled as $D = D_0(p) + \beta$, where $D_0 : [0, 1] \to [\underline{d}_0, \overline{d}_0]$ is the (expected) demand function and β is the random noise with 0 mean. Unsatisfied demands are backlogged. Chen et al. (2021b) consider both linear models and generalized linear models for $D_0(p)$ with *unknown* parameters θ_0. The distribution for β is unknown in the nonparametric sense. Let $k > 0$ be the fixed ordering cost, $c > 0$ be the variable ordering cost of ordering one unit of inventory, and $h : \mathbb{R} \to \mathbb{R}^+$ be the holding cost (when the remaining inventory level is positive) or the backlogging cost (when the remaining inventory level is negative). The instantaneous reward for period t is

$$\mathfrak{r}_t = -k \times \mathbf{1}\{y_t > x_t\} - c(y_t - x_t) + p_t(D_0(p_t) + \beta_t) - h(y_t - D_0(p_t) - \beta_t),$$

and the firm would like to maximize the T-period total reward.

With known demand curve D_0 and noise distribution μ_0, the work of Chen and Simchi-Levi (2004a) proves that, under mild conditions, for both the average and discounted profit criterion there exists an $(s, S, \mathbf{p})$ policy that is optimal in the long run. Under an $(s, S, \mathbf{p})$-policy, the retailer will only order new inventories when $x_t < s$, and after the ordering of new inventories maintain $y_t = S$. The function $\mathbf{p}$ prescribes the pricing decision that depends on the initial inventory level of the same period.

The performance of a particular $(s, S, \mathbf{p})$ policy can be evaluated as follows. Define $H_0(x, p; \mu)$ as the *expected* immediate reward of pricing decision p at inventory level x, without ordering new inventories. It is easy to verify that

$$H_0(x, p; \mu) = -\mathbb{E}_\mu[h(x - D_0(p) - \beta)] + pD_0(p) - cD_0(p). \tag{12.22}$$

For a certain $(s, S, \mathbf{p})$ policy, define quantities $I(s, x, \mathbf{p}; \mu)$ and $M(s, x, \mathbf{p}; \mu)$ as follows:

$$I(s, x, \mathbf{p}; \mu) = \begin{cases} H_0(x, \mathbf{p}(x); \mu) + \mathbb{E}_\mu[I(s, x - D_0(\mathbf{p}(x)) - \beta, \mathbf{p}; \mu)], & x \geq s, \\ 0, & x < s; \end{cases} \tag{12.23}$$

$$M(s,x,\mathbf{p};\mu) = \begin{cases} 1 + \mathbb{E}_{\mu}[M(s, x - D_0(\mathbf{p}(x)) - \beta, \mathbf{p}; \mu)], & x \geq s, \\ 0, & x < s; \end{cases} \tag{12.24}$$

Define $r(s,S,\mathbf{p};\mu)$ as

$$r(s,S,\mathbf{p};\mu) = \frac{-k + I(s,S,\mathbf{p};\mu)}{M(s,S,\mathbf{p};\mu)}. \tag{12.25}$$

When $I(s,S,\mathbf{p};\mu_0)$ and $M(s,S,\mathbf{p};\mu_0)$ are bounded, Lemma 2 from Chen and Simchi-Levi (2004a) shows that $\lim_{T\to\infty} R_T(\pi) = r(s,S,\mathbf{p};\mu_0)$.

Learning Algorithm The learning algorithm proposed in Chen et al. (2021b) is based on an $(s,S,\mathbf{p})$-policy with evolving inventory levels (s,S) and pricing strategies $\mathbf{p}$. Because unsatisfied demands are backlogged, the decision maker can observe true demand realizations. A regularized least-squares estimation is used to estimate θ_0, and a sample average approximation approach is used to construct an empirical distribution for β.

Next we present the detailed learning algorithm. For linear models, $\mathfrak{D}(\eta(p)|\theta_0) = \eta(p)^\top\theta_0$, and the unknown parameter θ_0 is estimated by the (regularized) least-squares estimation, i.e., let

$$\hat{\theta}_{\text{Linear}} := \arg\min_{\theta\in\mathbb{R}^{\mathfrak{d}}} \left\{ \frac{1}{2}\sum_{t\in\mathcal{H}} \left| d_t - \langle \eta(p_t), \theta\rangle \right|^2 + \frac{1}{2}\|\theta\|_2^2 \right\}. \tag{12.26}$$

For generalized linear models, $\mathfrak{D}(\eta(p)|\theta_0) = \upsilon(\eta(p)^\top\theta_0)$ for $\upsilon(\cdot)$ as a given *link function*. Let the unknown parameter θ_0 be estimated by

$$\hat{\theta}_{\text{GLM}} := \arg\min_{\theta\in\Theta} \left\| \sum_{t\in\mathcal{H}} (\upsilon(\eta(p_t)^\top\theta) - d_t)\eta(p_t) \right\|_{\Lambda^{-1}}. \tag{12.27}$$

Let $b \in \{1,2,\cdots\}$ be a particular epoch and $\mathcal{H}_{b-1} = \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_{b-1}$ be the union of all epochs prior to b. For time period $t \in \mathcal{H}_{b-1}$, let p_t be the advertised price and $d_t = D_0(p_t) + \beta_t$ be the realized demand. Let the *estimate* $\hat{\theta}_b$ of the unknown regression parameter θ_0 be computed by (12.26) if demand is linear or (12.27) if demand is generalized linear given samples from $\mathcal{H}_{b-1}$. Define $\Lambda_b := I_{\mathfrak{d}\times\mathfrak{d}} + \sum_{t\in\mathcal{H}_{b-1}} \eta(p_t)\eta(p_t)^\top$. For every $p \in [0,1]$, define $\Delta_b(p)$ as

$$\Delta_b(p) := \gamma\sqrt{\eta(p)^\top\Lambda_b^{-1}\eta(p)},$$

where $\gamma > 0$ is the oracle-specific parameter. We then define an upper estimate of D_0, $\bar{D}_b$, as

$$\bar{D}_b(p) := \min\left\{\overline{d}_0, \underline{d}_0 + L^2(1-p), \mathfrak{D}(\eta(p)|\hat{\theta}_b) + \Delta_b(p)\right\}, \tag{12.28}$$

where $\overline{d}_0$, $\underline{d}_0$ are maximum and minimum demands and L is the Lipschitz constant. Note that the Lipschitz continuity of $\eta(p)$ and $\Lambda_b \succeq I$ imply the continuity of $\Delta_b(\cdot)$ in p, which further implies the continuity of $\bar{D}_b(\cdot)$ in p.

One key challenge in the learning-while-doing setting is the fact that all of the important quantities H_0, I, M and r involve expectational evaluated under the noise distribution μ_0, an object which we do not know a priori. In this section, we give details on how empirical distributions are used to approximate μ_0.

At the beginning of epoch b, let $\mathcal{E}_{<b} \subseteq \mathcal{B}_1 \cup \cdots \mathcal{B}_{b-1}$ be a *non-empty subset* of historical selling periods used to approximate the noise distribution μ_0. We define the empirical noise distribution $\hat{\mu}_b$ as

$$\hat{\mu}_b := \frac{1}{|\mathcal{E}_{<b}|} \sum_{t \in \mathcal{E}_{<b}} \mathbb{I}[d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})], \tag{12.29}$$

where $\mathbb{I}[\beta']$ is the point mass at β' and $b(t)$ denotes the epoch to which selling period t belongs. Note that samples in $\{d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})\}_t$ are *dependent* because both p_t and $\hat{\theta}_{b(t)}$ are dependent across periods. Due to technical reasons, $\mathcal{E}_{<b}$ is *not* chosen to include all selling periods prior to epoch b. Instead, we construct $\mathcal{E}_{<b}$ such that all $t \in \mathcal{E}_{<b}$ have small estimation errors of D_0 on the advertised prices.

To further upper bound the deviation of $H_0(x, p; \hat{\mu}_b)$ from $H_0(x, p; \mu_0)$, we need to demonstrate that the empirical distribution $\hat{\mu}_b$ is close to the true noise distribution μ_0. Because such deviations must include the estimation errors of D_0 by $\bar{D}_{b(t)}$ themselves, it is crucial to select time periods $t \in \mathcal{B}_1 \cup \cdots \mathcal{B}_{b-1}$ during which the error $\Delta_{b(t)}(p_t)$ is small. To this end, we define $\mathcal{E}_{<b}$ as

$$\mathcal{E}_{<b} := \left\{ t \in \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_{b-1} : \Delta_{b(t)}(p_t) \leq \kappa/\sqrt{b} \right\}, \tag{12.30}$$

where $\kappa > 0$ is a scaling algorithm parameter, set as $\kappa = 2\underline{d}^{-3/2}\overline{dS}^{3/2}\gamma\sqrt{\mathfrak{d}\ln(TL^2)}$. Note that κ will only depend logarithmically on T. As is shown in the proof of the paper, the selection of κ leads to $|\mathcal{E}_{<b}| \geq b/2$, meaning that the set is non-empty, and, therefore, the definition in Eq. (12.30) is proper. The idea of the construction of $\mathcal{E}_{<b}$ in Eq. (12.30) is as follows. Note that $d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)}) = \beta_t + (\mathfrak{D}(\eta(p_t)|\theta_0) - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)}))$. While β_t is the desired sample from the noise distribution, $\mathfrak{D}(\eta(p_t)|\theta_0) - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})$ is incurred due to the estimation error of $\hat{\theta}_{b(t)}$, which may be very large. Also note that the absolute value of this estimation error is upper bounded by $\Delta_{b(t)}(p_t)$. Constructing $\mathcal{E}_{<b}$ as in Eq. (12.30) allows us to only exploit selling periods during which the estimation errors are sufficiently small. This ensures that the obtained (approximate) noise samples $\{d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})\}_{t \in \mathcal{E}_{<b}}$ are of high quality.

With the upper-confidence bounds $\bar{D}_b$ and the approximate noise distribution $\hat{\mu}_b$ constructed at the beginning of epoch b, (Chen et al., 2021b) use the dynamic programming approach detailed in the work of Chen and Simchi-Levi (2004a) to obtain an approximately optimal strategy $(s_b, S_b, \mathbf{p}_b)$ to be carried out during epoch b.

First define an upper bound estimate $\bar{H}_b(x, p; \hat{\mu}_b)$ on $H_0(x, p; \hat{\mu}_b)$ as

$$\bar{H}_b(x, p; \hat{\mu}_b) := -\mathbb{E}_{\hat{\mu}_b}[h(x - \bar{D}_b(p) - \beta)] + p\bar{D}_b(p) - c\bar{D}_b(p) + (c + L')\Delta_b(p), \tag{12.31}$$

where L' is a constant defined in Assumption (A3) of the paper.

For any $s \in [\underline{s}, \overline{s}]$, $S \in [\underline{S}, \overline{S}]$, $r \in \mathbb{R}$, demand function $D : [0, 1] \to [\underline{d}, \infty)$, noise distribution μ and their associated $H : \mathbb{R} \times [0, 1] \to \mathbb{R}$, define

$$\begin{aligned} &\phi^{(s,S)}(x; D, r, \mu) \\ &:= \begin{cases} \sup_{p\in[0,1]} H(x, p; \mu) - r + \mathbb{E}_\mu[\phi^{(s,S)}(x - D(p) - \beta; D, r, \mu)], & x \geq s; \\ 0, & x < s. \end{cases} \end{aligned} \tag{12.32}$$

With $D = \bar{D}_b$ and $H = \bar{H}_b(\cdot, \cdot; \hat{\mu}_b)$, the functions $\phi^{(s,S)}(x; \bar{D}_b, r, \hat{\mu}_b)$ can be computed for every $s \in [\underline{s}, \overline{s}]$, $S \in [\underline{S}, \overline{S}]$ and $r \in \mathbb{R}$, since both $H(\cdot, \cdot; \hat{\mu}_b)$ and the expectation with respect to $\hat{\mu}_b$ can be evaluated. For every (s, S), define

$$\bar{r}_b(s, S) := \inf\{r \in \mathbb{R} : \phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b) = k\} \tag{12.33}$$

and let the pricing strategy $\mathbf{p}$ (associated with inventory levels s, S) be the optimal solution to the $\phi^{(s,S)}(\cdot; \bar{D}_b, \bar{r}_b(s, S), \hat{\mu}_b)$ dynamic programming; that is, $\mathbf{p}(x)$ is defined such that $\phi^{(s,S)}(x; \bar{D}_b, \bar{r}_b(s, S), \hat{\mu}_b) = \bar{H}_b(x, \mathbf{p}(x); \hat{\mu}_b) - \bar{r}_b(s, S) + \mathbb{E}_{\hat{\mu}_b}[\phi^{(s,S)}(x - \bar{D}_b(\mathbf{p}(x)) - \beta; \bar{D}_b, \bar{r}_b(s, S), \hat{\mu}_b)]$ for all x.

Comparing equations in (12.32)–(12.33) with those in (12.22)–(12.25), it is easy to observe connections between them. $r(s, S, \mathbf{p}; \mu)$ in (12.25) represents the expected per-period profit, which includes both the immediate reward H and the fixed ordering cost k. On the other hand, $\phi^{(s,S)}(S; D, r, \mu)$ in (12.32) accumulates the immediate reward H over time and subtracts a constant r every period. If the constant r in (12.32) equals the expected per-period profit involving both H and k, intuitively one would expect $\phi^{(s,S)}(S; D, r, \mu)$ to be equal to k. Lemma 3 of Chen and Simchi-Levi (2004b) confirms this connection, which shows that $\phi^{(s,S)}(S; D, r^*(s, S), \mu) = k$, where $r^*(s, S) = \sup_{\mathbf{p}} r(s, S, \mathbf{p}; \mu)$. Therefore, $\bar{r}_b(s, S)$ can be considered as an empirical approximation of $r^*(s, S)$.

We finally remark that in practice, one may discretize the choices of s, S, x, and p in the dynamic programming scheme described above with granularity T^{-1}. This leads to a computationally efficient algorithm. On the other hand, by the Lipschitz property of $\overline{H}_b(\cdot, \cdot; \hat{\mu}_b)$, it can be shown that the error caused by discretization is at most $O(T^{-1})$, which does not affect the order of the overall regret.

The proposed algorithm is based on an $(s, S, \mathbf{p})$-policy with evolving inventory levels (s, S) and pricing strategies $\mathbf{p}$. As mentioned earlier, in the learning algorithm the T time periods are partitioned into *epochs*, labeled as $\mathcal{B}_1, \mathcal{B}_2, \cdots$. Re-stocking only occurs at the first time period of each epoch $\mathcal{B}_b$, $b \in \{1, 2, \cdots\}$. Each epoch $\mathcal{B}_b$ is also associated with inventory levels (s_b, S_b) and pricing strategy $\mathbf{p}_b$, such that for the first time period $t_b \in \mathcal{B}_b$, the re-stocked inventory level is $y_{t_b} = S_b$; the epoch

Algorithm 10 The main algorithm: dynamic inventory control and pricing with unknown demand

```
1: Input: problem parameters $k, c, h$, time horizon $T$, the regression-oracle-specific parameter $\gamma$.
2: Output: inventory and pricing decisions $y_t$, $p_t$ for each $t \in [T]$.
3: for epoch $b = 1, 2, 3, \cdots$ do
4:     Compute the model estimate $\hat{\theta}_b$ using the regression oracle $\mathcal{O}$ and samples from $\mathcal{H}_{b-1}$;
5:     Construct upper-confidence bounds $\bar{D}_b$ as in Eqs. (12.28, 12.31);
6:     Construct $\hat{\mu}_b = \frac{1}{|\mathcal{E}_{<b}|} \sum_{t \in \mathcal{E}_{<b}} \mathbb{I}[d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})]$, where $\mathcal{E}_{<b}$ is constructed in Eq. (12.30);
7:     For every $s \in [\underline{s}, \overline{s}]$, $S \in [\underline{S}, \overline{S}]$ compute $\phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b)$ as in Eq. (12.32) and find $\bar{r}_b(s, S) = \inf\{r \in \mathbb{R} : \phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b) = k\}$;
8:     Select $(s_b, S_b) = \arg\max_{s,S} \bar{r}_b(s, S)$ and let $\mathbf{p}_b$ be the optimal pricing decisions associated with dynamic programming $\phi^{(s_b,S_b)}(\cdot; \bar{D}_b, \bar{r}_b(s_b, S_b), \hat{\mu}_b)$;
9:     For the first time period $t_b$ in epoch $\mathcal{B}_b$ set $y_{t_b} = S_b$ and $p_{t_b} = \mathbf{p}_b(S_b)$; for the rest of epoch $\mathcal{B}_b$ set $y_t = x_t$ and $p_t = \mathbf{p}_b(x_t)$; epoch $\mathcal{B}_b$ terminates once $x_t < s_b$;
10: end for
```

$\mathcal{B}_b$ terminates whenever $x_t < s_b$, and for all $t \in \mathcal{B}_b \backslash \{t_b\}$, $y_t = x_t$ and $p_t = \mathbf{p}_b(x_t)$. Algorithm 10 gives a pseudo-code description of the proposed algorithm.

Updates of the $(s, S, \mathbf{p})$ policies being implemented occur at the beginning of each epoch, as detailed from Step 4 to Step 8 in Algorithm 10. More specifically, at the beginning of epoch b when policy update is due, the algorithm first collects all realized demand information from previous epochs to construct model estimate $\hat{\theta}_b$ (of the demand-rate curve) and noise distribution $\hat{\mu}_b$. With estimates $\hat{\theta}_b$ and $\hat{\mu}_b$, dynamic programming (reflected in $\phi^{(s_b,S_b)}(\cdot; \bar{D}_b, \bar{r}_b, \hat{\mu}_b)$) is computed to obtain an approximately optimal pricing function $\mathbf{p}_b$, as well as the inventory levels s_b, S_b.

Regret Convergence Regret of the algorithm described above is upper bounded by $\widetilde{O}(T^{1/2})$ with probability $1 - O(T^{-1})$, where π^* is the optimal policy that maximizes $r(s, S, \mathbf{p}; \mu_0)$. In the $\widetilde{O}(\cdot)$ notation we omit polynomial dependency on $\log T$ and other problem parameters. With $k = c = 0$ and $h(\cdot) \equiv 0$, the problem becomes a pure pricing problem with unknown linear demand functions. As long as $\tau > 1$, the work of Broder and Rusmevichientong (2012) proves an $\Omega(T^{1/2})$ lower bound for any admissible pricing policies. Therefore, the $\widetilde{O}(T^{1/2})$ regret established here is optimal.

In Algorithm 10, a dynamic programming needs to be carried out after each epoch b to obtain a new policy $(s_b, S_b, \mathbf{p}_b)$. Because each epoch lasts at most $\overline{S}/\underline{d} = O(1)$ selling periods, the algorithm requires $\Omega(T)$ DP calculations which can be computationally expensive. Chen et al. (2021b) then propose an improved algorithm that only needs $O(\tau \log T)$ DP calculations to achieve virtually the same regret, which is much more computationally efficient.

Algorithm with Infrequent DP Updates The detailed description is presented in Algorithm 11.

Note that in Algorithm 11, a new $(s, S, \mathbf{p})$ policy is computed only if 2^{ι}, $\iota \in \{1, 2, \cdots, \}$ epochs are met, or the determinant of the sample covariance

Algorithm 11 Dynamic inventory control and pricing with infrequent DP solutions

```
Input: problem parameters $k, c, h$, time horizon $T$, the regression-oracle-specific parameter $\gamma$.
Output: inventory and pricing decisions $y_t, p_t$ for each $t \in [T]$.
Initialize: $\hat{\theta}_0 = 0^{\mathfrak{d}}$, $\Lambda_1 = I_{\mathfrak{d}\times\mathfrak{d}}$ and $\zeta_1 = 1$;
for epoch $b = 1, 2, 3, \cdots$ do
    if $\det(\Lambda_b) \geq 2\zeta_b$ or $b = 2^{\iota}$ for some $\iota \in \mathbb{N}$ then
        Update $\zeta_{b+1} = \det(\Lambda_b)$ and compute the model estimate $\hat{\theta}_b$ using the regression oracle $\mathcal{O}$ and samples from $\mathcal{H}_{b-1}$;
        Construct upper-confidence bounds $\bar{D}_b$ as in Eqs. (12.28,12.31);
        Construct $\hat{\mu}_b = \frac{1}{|\mathcal{E}_{<b}|}\sum_{t\in\mathcal{E}_{<b}} \mathbb{I}[d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})]$, where $\mathcal{E}_{<b}$ is constructed in Eq. (12.30);
        For every $s, S \in [\underline{s}, \overline{S}]$ compute $\phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b)$ as in Eq. (12.32) and find $\bar{r}_b(s, S) = \inf\{r \in \mathbb{R} : \phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b) = k\}$;
        Select $(s_b, S_b) = \arg\max_{s,S} \bar{r}_b(s, S)$ and let $\mathbf{p}_b$ be the optimal pricing decisions associated with dynamic programming $\phi^{(s_b,S_b)}(\cdot; \bar{D}_b, \bar{r}_b(s_b, S_b), \hat{\mu}_b)$;
    else
        Set $\hat{\theta}_b = \hat{\theta}_{b-1}$, $\zeta_{b+1} = \zeta_b$, $\overline{D}_b = \overline{D}_{b-1}$, $\hat{\mu}_b = \hat{\mu}_{b-1}$,$s_b = s_{b-1}$, $S_b = S_{b-1}$ and $\mathbf{p}_b = \mathbf{p}_{b-1}$;
    end if
    If the current inventory level exceeds $S_b$, set $p_t = 0$ until inventory level falls below $S_b$; *
    For the first time period $t_b$ in epoch $\mathcal{B}_b$ set $y_{t_b} = S_b$ and $p_{t_b} = \mathbf{p}_b(S_b)$; for the rest of epoch $\mathcal{B}_b$ set $y_t = x_t$ and $p_t = \mathbf{p}_b(x_t)$; epoch $\mathcal{B}_b$ terminates once $x_t < s_b$;
    Update $\Lambda_{b+1} = \Lambda_b + \sum_{t\in\mathcal{B}_b} \eta(p_t)\eta(p_t)^{\top}$;
end for
```

* Note that this step may only happen when the policy changes. It does not belong to any epoch; and since it happens very infrequently, its incurred regret can be bounded separately.

Λ_b doubles. This greatly reduces the number of DP calculations from $O(T)$ to $O(\tau \log T)$.

Regret Convergence for Infrequent DP Updates For the algorithm with infrequent DP updates, the regret is upper bounded by $\widetilde{O}(T^{1/2})$ with probability $1 - O(T^{-1})$.

12.6 Other Models

Burnetas and Smith (2000) is one of the earliest papers, if not the first one, that studies joint pricing and inventory control with unknown demand distribution. They assume the lost-sales cost is zero and inventory perishes at the end of each period. The pricing mechanism is modeled as a multiarmed bandit problem, while the order quantity decision is made based on a stochastic approximation procedure. Burnetas and Smith (2000) proves policy convergence of their proposed algorithm. Katehakis et al. (2020) consider the joint optimization problem with discrete backlogged demand in different settings with or without a leading price. Keskin et al. (2021)

study the joint pricing and inventory control problem with learning in a changing environment under a parametric demand-rate function and assume lost sales are observable. They provide learning algorithms whose convergence rates match the theoretical lower bound.

References

Broder, J., & Rusmevichientong, P. (2012), Dynamic pricing under a general parametric choice model. *Operations Research, 60*(4), 965–980.

Burnetas, A. N., & Smith, C. E. (2000). Adaptive ordering and pricing for perishable products. *Operations Research, 48*(3), 436–443.

Chen, B., & Chao, X. (2019). Parametric demand learning with limited price explorations in a backlog stochastic inventory system. *IISE Transactions, 51*(6), 605–613.

Chen, X., & Simchi-Levi, D. (2004a). Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The finite horizon case. *Operations Research, 52*(6), 887–896.

Chen, X., & Simchi-Levi, D. (2004b). Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The infinite horizon case. *Mathematics of Operations Research, 29*(3), 698–723.

Chen, B., Chao, X., & Ahn, H. S, (2019). Coordinating pricing and inventory replenishment with nonparametric demand learning. *Operations Research, 67*(4), 1035–1052.

Chen, B., Chao, X., & Wang, Y. (2020a). Data-based dynamic pricing and inventory control with censored demand and limited price changes. *Operations Research, 68*(5), 1445–1456.

Chen, B., Wang, Y., & Zhou, Y. (2020b). Optimal policies for dynamic pricing and inventory control with nonparametric censored demands. Available at SSRN 3750413.

Chen, B., Chao, X., & Shi, C. (2021a). Nonparametric learning algorithms for joint pricing and inventory control with lost-sales and censored demand. *Mathematrics of Operations Research, 46*(2), 726–756.

Chen, B., Simchi-Levi, D., Wang, Y., & Zhou, Y. (2021b). Dynamic pricing and inventory control with fixed ordering cost and incomplete demand information. *Management Science, forthcoming*.

Chen, X., & Simchi-Levi, D. (2012). Pricing and inventory management. *The Oxford Handbook of Pricing Management, 1*, 784–824.

Cheung, W. C., Simchi-Levi, D., & Wang, H. (2017). Dynamic pricing and demand learning with limited price experimentation. *Operations Research, 65*(6), 1722–1731.

Elmaghraby, W., & Keskinocak, P. (2003). Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science, 49*(10), 1287–1309.

Katehakis, M. N., Yang, J., & Zhou, T. (2020). Dynamic inventory and price controls involving unknown demand on discrete nonperishable items. *Operations Research, 68*(5), 1335–1355.

Keskin, N. B., & Zeevi, A. (2014). Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research, 62*(5), 1142–1167.

Keskin, N. B., Li, Y., & Song, J. S. J. (2021). Data-driven dynamic pricing and ordering with perishable inventory in a changing environment. *Management Science, 68*(3), 1938–1958.

Kleywegt, A. J., Shapiro, A., & Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization, 12*(2), 479–502.

Lei, Y. M., Jasin, S., & Sinha, A. (2014). Near-optimal bisection search for nonparametric dynamic pricing with inventory constraint. Ross School of Business Paper (1252)

Levi, R., Roundy, R. O., & Shmoys, D. B. (2007). Provably near-optimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research, 32*(4), 821–839.

Levi, R., Perakis, G., & Uichanco, J. (2015). The data-driven newsvendor problem: New bounds and insights. *Operations Research, 63*(6), 1294–1306.

Petruzzi, N. C., & Dada, M. (1999). Pricing and the newsvendor problem: A review with extensions. *Operations Research, 47*(2), 183–194.

Schumaker, L. (2007). *Spline functions: Basic theory*. Cambridge, UK: Cambridge University Press.

Sobel, M. J. (1981). Myopic solutions of Markov decision processes and stochastic games. *Operations Research, 29*(5), 995–1009.

Wang, Z., Deng, S., & Ye, Y. (2014). Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research, 62*(2), 318–331.

Whitin, T. M. (1955). Inventory control and price theory. *Management Science, 2*(1), 61–68.

Wu, C. F. J., et al. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics, 14*(4), 1261–1295.

Yano, C. A., & Gilbert, S. M. (2005). Coordinated pricing and production/procurement decisions: A review. *Managing Business Interfaces* (pp. 65–103).