European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Production, Manufacturing, Transportation and Logistics

Nonparametric multi-product dynamic pricing with demand learning via simultaneous price perturbation

Xiangyu Yang [a], Jianghua Zhang [a,*], Jian-Qiang Hu [b], Jiaqiao Hu [c,*]

[a] School of Management, Shandong University, 250100, Jinan, China
[b] School of Management, Fudan University, 200433, Shanghai, China
[c] Department of Applied Mathematics & Statistics, State University of New York at Stony Brook, NY 11794-3600, Stony Brook, USA

ARTICLE INFO

Keywords:
Decision analysis
Dynamic pricing with demand learning
Online learning
Simultaneous perturbation stochastic approximation (SPSA)
Revenue management

ABSTRACT

We consider the problem of multi-product dynamic pricing with demand learning and propose a nonparametric online learning algorithm based on the simultaneous perturbation stochastic approximation (SPSA) method. The algorithm uses only two price experimentations at each iteration, regardless of problem dimension, and could be especially efficient for solving high-dimensional problems. Under moderate conditions, we prove that the price estimates converge in mean-squared error (MSE) to the optimal price. Furthermore, we show that by suitably choosing input parameters, our algorithm achieves an expected cumulative regret of order $O\left(\sqrt{T}\right)$ over T periods, which is the best possible growth rate in the literature. The exact constants in the rate can be identified explicitly. We investigate the extensions of the algorithm to application scenarios characterized by non-stationary demand and inventory constraints. Simulation experiments reveal that our algorithm is effective for a range of test problems and performs favorably compared with a recently proposed alternative method for high-dimensional problems.

1. Introduction

Dynamic pricing has become a prevalent strategy adopted by businesses across diverse industries to maximize their revenue. This pricing technique involves adjusting the price of a product in real time based on various factors, such as customer demand, competition, time of day, seasonality, and inventory levels. Among these factors, customer demand is perhaps the most critical. With advancements in technology and data analytics, sellers can now use statistical tools to learn customer demands by continuously updating their prices. This approach is known as demand learning, which entails collecting and analyzing customer data to identify patterns in their purchasing behavior. By leveraging these data, sellers can adjust their prices dynamically to maximize revenue. This procedure is called dynamic pricing with demand learning, which is also referred to as learning and earning.

The problem of dynamic pricing with demand learning has attracted considerable attention in the literature of operations research and management science over the years. Most papers study the problem under the setting where the demand function of the product follows a parametric model (e.g., linear model or generalized linear model) so that the demand learning problem boils down to the estimation of unknown parameters in the demand function. Arguably, the most straightforward pricing policy (i.e., learning algorithm) would typically involve establishing the estimate of the optimal price based on the prevailing parameter estimates. This approach can be viewed as a myopic or greedy policy, also known as passive learning or certainty-equivalence control. Though simple in its implementation, this approach may perform poorly. For instance, den Boer and Zwart (2014) show that the prices generated by the certainty-equivalence policy may converge with a positive probability to a non-optimal price. In fact, the certainty-equivalence pricing policy is an example of incomplete learning that describes the phenomenon where parameter estimates in a dynamic decision problem with parameter uncertainty can converge to an incorrect value with positive probabilities; cf. Keskin and Zeevi (2018).

Extensive findings from the literature underscore a crucial factor to avoid incomplete learning: the inclusion of adequate price dispersion within the pricing policy (Keskin & Zeevi, 2014). This insight highlights the need to integrate a dedicated price experimentation step into the learning algorithm, preventing prices from converging too rapidly. By doing so, an abundance of new and valuable information could be obtained, leading to significant improvement in the accuracy of parameter estimates. We briefly describe three pricing policies satisfying the

* Corresponding authors.
E-mail addresses: yangxiangyu@email.sdu.edu.cn (X. Yang), zhangjianghua@sdu.edu.cn (J. Zhang), hujq@fudan.edu.cn (J.-Q. Hu), jiaqiao.hu.1@stonybrook.edu (J. Hu).

https://doi.org/10.1016/j.ejor.2024.06.017
Received 28 June 2023; Accepted 12 June 2024
Available online 21 June 2024

above requirement. The first example is the MLE-CYCLE policy proposed in Broder and Rusmevichientong (2012). MLE-CYCLE operates in cycles, and each cycle is separated by an exploration phase and an exploitation phase to balance the tradeoff between demand learning (i.e., parameter estimation) and revenue optimization (i.e., the myopic price concerning the current estimate of the demand parameter). The second example is the controlled variance pricing proposed in den Boer and Zwart (2014). This policy charges a myopic price at each period unless this price leads to insufficient price dispersion to ensure the sample variance of prices is greater than or equal to a lower bound; in the latter case, a small perturbation term is added to the myopic price. The third example is the semimyopic pricing scheme given in Besbes and Zeevi (2015). The policy also combines estimation and optimization in a cycle-based manner, with the selling prices during a cycle being adjusted, either upward or downward, on the basis of the estimated optimal price. The exploration and exploitation dilemma is resolved by appropriately tuning the length of cycles and the magnitude of perturbations. All the aforementioned pricing policies can be applied to the multi-product setting when the seller has multiple products to price and sell. For example, den Boer (2014) introduces an adaptive pricing policy, extending the controlled variance pricing to multiple dimensions.

In practice, the functional relationship between demand and price is not easily accessible to the sellers. This is particularly evident when a new product is introduced into the market for initial sale or when the market conditions undergo alterations. In such cases, it is plausible to take an approach that avoids specifying the explicit form of the demand function, so that the seller can employ a more flexible model for analyzing the demand–price relationship. Such an approach enables better representation of consumer behavior by taking into account the dynamic nature of the market, and thereby facilitates well-informed decision-making in an obscure market environment. Note that this type of pricing policy deals with quite general multi-product settings that allows different cross elasticities among the demands of products, either complement or substitute, and it includes the parametric models (e.g., den Boer & Zwart, 2014; Broder & Rusmevichientong, 2012; Keskin & Zeevi, 2014) as special cases. However, there are relatively few papers on nonparametric algorithms for dynamic pricing with demand learning. To the best of our knowledge, the Kiefer–Wolfowitz (KW) pricing policy proposed in Hong, Li, and Luo (2020) is the only existing multi-product nonparametric solution algorithm in the literature. The method applies the classical KW stochastic approximation (SA) algorithm, one of the most widely-used model-free stochastic optimization algorithms within an online pricing context. At each price experimentation step, the KW pricing policy uses $d+1$ revenue function evaluations to collect information on dispersed prices, where d is the price dimension. It has been shown in Hong et al. (2020) that the price estimates converge in mean-squared error (MSE) to the optimal price and that an upper bound on the regret (i.e., the relative loss due to not using the optimal price) is of order $\sqrt{T}$, which is the best possible growth rate (see, e.g., Broder & Rusmevichientong, 2012; Keskin & Zeevi, 2014). Nevertheless, because the per-iteration complexity of the KW pricing policy grows with the price dimension, the algorithm may not be very efficient on high-dimensional problems. In particular, when the quantity of products is much larger than the total decision periods, an (online) implementation of a KW pricing policy may not even be feasible.

In this paper, following the nonparametric formulation of Hong et al. (2020), we propose a novel pricing policy based on the simultaneous perturbation stochastic approximation (SPSA) technique introduced by Spall (1992). Such a technique allows the algorithm to explore different prices with only two revenue function evaluations, regardless of the price dimension, avoiding the need for expending a significant amount of computational effort for price experimentation at each step. The revenue estimates are then exploited to construct a stochastic gradient estimator for updating the price. Under certain smoothness and regularity conditions on the revenue function, we show that the regret of our policy can also achieve the best possible growth rate under appropriate choices of algorithmic parameters. Our regret analysis reveals no significant performance gap between parametric and nonparametric models. Consequently, our algorithm can help reduce the effort needed in searching for the correct parametric forms of demand functions. For high-dimensional pricing problems, we propose two additional strategies to further enhance the algorithm's empirical performance. The extensions of the algorithm to application scenarios involving non-stationary demand and finite inventory constraints are also investigated.

To summarize, we view our work as providing the following research contributions:

- We introduce a novel online learning algorithm for nonparametric multi-product dynamic pricing with demand learning, characterize the growth rate of the regret, and present enhancement strategies to improve the algorithm's practical performance.
- From a theoretical point of view, we relate the multivariate simulation/stochastic optimization approach to the dynamic pricing problem and show that such a problem can indeed be effectively solved by SPSA-based algorithms that achieve the optimal learning rate, both in stationary and non-stationary demand environments.
- In terms of practical implications, SPSA-based algorithms could be particularly useful for solving high-dimensional problems, even in the presence of inventory constraints, because the number of revenue function evaluations required per iteration is independent of the number of pricing decisions.
- Finally, the new algorithm serves as a valuable complement to existing dynamic pricing policies that rely primarily on parametric approaches to single-product (or low-dimensional) problems.

The rest of this paper is structured as follows. Related literature is reviewed in Section 2. We present the proposed pricing policy under the basic model in Section 3 and analyze its regret in Section 4. Two significant extensions of the model that incorporate non-stationary demand and finite inventory constraints are investigated in Section 5. Numerical experiments are provided in Section 6. We conclude the paper with some future research topics in Section 7. All proofs and additional numerical results are included in the Appendix.

2. Related literature

The basic model of this paper assumes that the market environment (demand function) remains unchanged and the seller possesses an infinite inventory (i.e., no limit on the amount of resources that can be used). Each of these conditions is then relaxed in Section 5. We begin by reviewing the related literature focusing on the basic model setting and then discuss existing studies that consider non-stationary market environments and limited inventory supplies.

Under the basic model setting, in addition to the studies (Besbes & Zeevi, 2015; den Boer & Zwart, 2014; Broder & Rusmevichientong, 2012) mentioned in Section 1, some earlier works include, e.g., Bertsimas and Perakis (2006), Carvalho and Puterman (2005), Lobo and Boyd (2003), suggesting that pricing policies incorporating active exploration tend to outperform myopically greedy policies. This observation underscores the importance of engaging in price experimentation. Cheung, Simchi-Levi, and Wang (2017) establish asymptotically optimal policies when the seller is constrained to a finite number of allowable price changes. den Boer and Keskin (2020) consider the cases where discontinuities are permitted in the demand function. From a Bayesian perspective, Harrison, Keskin, and Zeevi (2012) address a stationary, two-hypothesis dynamic pricing problem. Robust optimization approaches to address demand uncertainty are also investigated, as discussed in Bergemann and Schlag (2011). Interested readers may consult the survey papers (e.g., Chen & Chen, 2015; Den Boer, 2015a) for

comprehensive reviews on various dynamic pricing and learning problem formulations. Also closely related with this topic is the stream of literature on multi-armed bandit (MAB) problems, initially introduced by Thompson (1933). While the original emphasis was on medical trials, the applications of MAB algorithms have undergone substantial expansions in contemporary research (see, e.g., Bubeck et al., 2012; Lattimore & Szepesvári, 2020). The approach closest to our work in the MAB literature is the continuum-armed bandit problems, in which there is an infinite number of arms (i.e., pricing decisions). Similar to the setting of our basic model, the literature on continuum-armed bandits does not consider resource constraints and presents various policies for learning the maximum of an objective function (e.g., Auer, Ortner, & Szepesvári, 2007; Bubeck, Munos, Stoltz, & Szepesvári, 2011; Grant & Szechtman, 2021; Kleinberg, 2004; Kleinberg, Slivkins, & Upfal, 2008; Misra, Schwartz, & Abernethy, 2019). These policies can be employed to maximize the expected revenue as a function of price but are much more complex to implement than ours. In contrast, we employ a direct stochastic gradient-based approach to maximize the revenue function.

For dynamic pricing problems with unknown non-stationary demand, Besbes and Zeevi (2011) apply a change-point detection approach to identify temporal shifts in the demand function. They investigate demand learning within an uncapacitated Bernoulli demand model, wherein the seller is initially aware of the demand curve. At an undisclosed point in time, the demand curve undergoes a switch to a different function of the price. The authors show that under the well-separated condition of demand curves, a myopically greedy policy turns out to be optimal. Keskin and Zeevi (2017) investigate a generalized version of the problem, which allows multiple change points. They examine the scenario where the seller lacks knowledge about the potential demand function, the timing of changes, and the total number of changes. A joint learning-and-detection policy is proposed, which attains a T-period regret on the order of $\sqrt{T}$, up to a logarithmic factor. For more general non-stationary demand settings, Den Boer (2015b) considers a demand environment characterized by unknown and non-stationary market size, with known price sensitivity of demand. The author formulates policies that hedge against potential demand changes and derives upper bounds on the long-run average regret for these policies.

In contrast to the setting of unlimited inventory, determining optimal prices becomes more challenging when inventory constraints are incorporated. Based on the foundational work on canonical revenue management models formalized in Gallego and Van Ryzin (1994) (single-product single-resource problems) and Gallego and Van Ryzin (1997) (multiple products using common resources, i.e., network revenue management problems), there is an array of literature focusing on nonparametric algorithms for dynamic pricing in the presence of inventory constraints; (see, e.g., Besbes & Zeevi, 2009, 2012; Chen & Gallego, 2022; Wang, Deng, & Ye, 2014; Yang & Xiong, 2020). The salient feature of these models is that the demand is given by a Poisson process whose intensity is controlled by the price, and the seller does not know the relationship between the price and the demand rate. We remark that the structure of the above solution algorithms significantly differs from that used in the basic setting of our paper. They consider the fluid approximation of the original continuous-time dynamic program and find an upper bound for the expected cumulative revenue based on the deterministic optimization counterpart. The objective is to design an algorithm with small asymptotic regret of the so-called "size-k" problem with both the initial inventory and the demand function scaled by k. In addition to the canonical revenue management formulation, the problem is connected to the so-called bandits with knapsacks model. Badanidiyuru, Kleinberg, and Slivkins (2013) is among the first to consider such a problem and modifies the celebrated upper confidence bound algorithm to accommodate the setting with inventory constraints. Agrawal and Devanur (2016) propose an algorithm for linear contextual bandits with knapsacks. (Ferreira, Simchi-Levi, & Wang, 2018) develop Thompson sampling algorithms tailored for both the linear demand case and the bandits with knapsacks problem.

On a technical level, the construction of our pricing policy hinges on SA methods, which are among the most commonly used online learning tools, dating back to two pioneering papers, Robbins and Monro (1951) and Kiefer and Wolfowitz (1952). SA was originally introduced to solving stochastic root-finding problems. When the function of interest is the gradient of the objective function in an optimization problem, SA can be interpreted as a stochastic version of the steepest ascent/descent algorithm for finding the first-order critical points. The theoretical properties of SA such as the local/global convergence and the rate of convergence can be analyzed by the ordinary differential equation (ODE) approach; (see, e.g., Borkar, 2009; Kushner & Yin, 2003; Meyn, 2022). However, the ODE-based convergence analysis is often conducted in an asymptotic form. In this study, we adopt the general bounds for the SA algorithms developed in Broadie, Cicek, and Zeevi (2011) and apply them to our variant of the SPSA algorithm to perform a finite-time analysis.

3. SPSA pricing policy for dynamic pricing with demand learning

3.1. Basic model

We consider a stylized dynamic pricing problem where a seller has d distinct products for sale over T discrete time periods, denoted as $t = 1, 2, \dots, T$. At the beginning of period t, the seller must select a price (vector), denoted by $\mathbf{p}_t = (p_{t,1}, \dots, p_{t,d})^{\mathrm{T}}$, for its products. After setting the price, the seller observes the demand during that period, denoted by $\mathbf{D}(\mathbf{p}_t) = (D_1(\mathbf{p}_t), \dots, D_d(\mathbf{p}_t))^{\mathrm{T}}$. Given any $\mathbf{p}_t$, $\mathbf{D}(\mathbf{p}_t)$ is a random vector, and we assume that its functional relationship with respect to $\mathbf{p}_t$ does not change over time but is unknown to the seller. We also assume that all products are nonperishable, and that all demands are fulfilled by the seller. Each of these assumptions will later be relaxed in Section 5. In particular, the non-stationary demand case will be addressed in Section 5.1, whereas in Section 5.2, we discuss the finite inventory setting. The marginal costs of all products are equal to 0 so that the revenue is equal to the profit. In period t, the seller's expected one-period revenue under any advertised price $\mathbf{p} \in \mathbb{R}^d$ is given by

$$f(\mathbf{p}) := \mathsf{E}\left[F(\mathbf{p}_t) | \mathbf{p}_t = \mathbf{p}\right], \tag{1}$$

where $F(\mathbf{p}) := \mathbf{p}^{\mathrm{T}}\mathbf{D}(\mathbf{p})$, $\forall \mathbf{p}$. After receiving the realized revenue $F(\mathbf{p}_t)$, the seller moves to the next period. The product price in the next period is determined by a stationary random pricing policy Ψ and the history of prices and demands collected through the end of period t, that is, $\mathbf{p}_{t+1} = \Psi(\mathbf{p}_1, \mathbf{D}(\mathbf{p}_1), \dots, \mathbf{p}_t, \mathbf{D}(\mathbf{p}_t))$. Such a class of policies is usually referred to as non-anticipating policies, inducing $\{\mathbf{p}_t\}$ to be a sequence of random vectors. Accordingly, the seller's expected cumulative revenue under policy Ψ across T periods is defined as

$$r(T, \Psi) := \mathsf{E}\left[\sum_{t=1}^{T} f(\mathbf{p}_t)\right].$$

Note that the expectation above is taken with respect to both the randomness in revenue outcomes and any randomization in the pricing policy. As is standard in the online learning literature, the performance metric we adopt is the T-period cumulative (static) regret, which is defined as follows:

$$R(T, \Psi) := T f^* - r(T, \Psi), \tag{2}$$

where $f^* = \sup_{\mathbf{p}} f(\mathbf{p})$, representing the maximal expected revenue under the true optimal price in each period. For instance, consider the commonly used linear demand model. The expected revenue at every period is a quadratic function of the posted price, i.e., $f(\mathbf{p}) = \mathbf{p}^{\mathrm{T}}\mathsf{E}[\mathbf{D}(\mathbf{p})] = \mathbf{p}^{\mathrm{T}}(\mathbf{a} + \mathbf{B}\mathbf{p})$, where $\mathbf{a} \in \mathbb{R}^d$ and $\mathbf{B}$ is a $d \times d$ matrix. Under some appropriate conditions, the true optimal price $\mathbf{p}^*$ is given by $\mathbf{p}^* = -(\mathbf{B}+\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{a}$ due to the first-order optimality condition (see Keskin

& Zeevi, 2014). Given any sub-optimal policy Ψ, the regret measures the expected revenue loss relative to a clairvoyant who always charges the optimal policy. Clearly, the regret is non-negative, and the lower the regret, the better the policy. The amount of the regret, especially how it grows over time, gives a perspective on how well a policy performs.

The crux of the nonparametric dynamic pricing model is that the demand $\mathbf{D}(\mathbf{p})$ is a stochastic "black-box" function of the posted price $\mathbf{p}$, i.e., there is minimal knowledge on how the aggregate market responds to the price. The seller is confronted with the challenge of adaptively balancing the acquisition of demand information while maximizing revenue in an online fashion. To address this issue, we adopt a stochastic gradient-based black-box optimization approach instead of assuming any particular form of the demand function. This approach is relatively simple to implement and offers greater flexibility in practical applications.

3.2. Algorithm description

The proposed algorithm operates in stages. At the beginning of each stage, which we index by $n \in \{1, 2, \ldots\}$, the seller has an estimate of the optimal price, namely, $\mathbf{p}_n$. Then the seller conducts two price experimentations by randomly perturbing $\mathbf{p}_n$, where the magnitude of the perturbation is determined based on a random vector $\boldsymbol{\Delta}_n = (\Delta_{n,1}, \ldots, \Delta_{n,d})^{\mathrm{T}} \in \mathbb{R}^d$, with $\Delta_{n,i}$ being the ith component of $\boldsymbol{\Delta}_n$ for all $i \in \{1, 2, \ldots, d\}$. We will discuss the regularity conditions on the choice of $\boldsymbol{\Delta}_n$ in Section 4. Each perturbed price will be used for one period, and the corresponding revenue is earned under this price. At the end of stage n, the price estimate is updated through a stochastic gradient ascent method in the spirit of the SPSA. The process continues until the time horizon comes to a close.

Let $\{a_n\}$ and $\{c_n\}$ be two real-valued parameter sequences, which will be used in the algorithm. The detailed algorithmic steps are presented below.

SPSA pricing policy: Ψ^{SPSA}

Step 0: Initialization

Select a step-size sequence $\{a_n\}$, a perturbation-size sequence $\{c_n\}$, and a starting price $\mathbf{p}_1$. Set period counter $t = 0$ and stage counter $n = 1$.

Step 1: Price Experimentation

Generate a realization of the random vector $\boldsymbol{\Delta}_n$.

- ▷ Set $t = t + 1$. Put $\tilde{\mathbf{p}}_t^+ = \mathbf{p}_n + c_n\boldsymbol{\Delta}_n$ and obtain a revenue observation F^+, where $F^+ = F(\tilde{\mathbf{p}}_t^+)$.
- ▷ Set $t = t + 1$. Put $\tilde{\mathbf{p}}_t^- = \mathbf{p}_n - c_n\boldsymbol{\Delta}_n$ and obtain a revenue observation F^-, where $F^- = F(\tilde{\mathbf{p}}_t^-)$.

Step 2: Price Updating

Compute

$$\mathbf{p}_{n+1} = \mathbf{p}_n + a_n \frac{F^+ - F^-}{2c_n} \boldsymbol{\Delta}_n^{-1}, \tag{3}$$

where $\boldsymbol{\Delta}_n^{-1}$ is the component-wise reciprocal of $\boldsymbol{\Delta}_n$. Set $n = n + 1$. If $t \le T$, then go to Step 1; otherwise, the algorithm outputs all price estimates $\{\tilde{\mathbf{p}}_t\}$ and terminates.

In Step 1, we use F^+ and F^- to denote the realized revenue per period under the perturbed prices $\tilde{\mathbf{p}}_t^+$ and $\tilde{\mathbf{p}}_t^-$. These terms play a crucial role in constructing the gradient estimate in Step 2. Notice that Ψ^{SPSA} is an online version of the classical SPSA. The classical SPSA is designed to solve offline optimization problems where only the terminal solutions of the iterations (i.e., $\{\mathbf{p}_n\}$) are relevant. On the other hand, our algorithm considers all evaluated solutions (i.e., $\{\tilde{\mathbf{p}}_t\}$) since they all lead to increased regrets. This distinction between offline and online optimizations sets the two algorithms apart. The online nature of Ψ^{SPSA} allows the seller to learn demand and/or revenue functions on the fly by leveraging past experiences, thus continuously improving its decision-making as the sale proceeds. For instance, online retailing platforms are natural application scenarios of our algorithms because one can benefit from adapting the service to the individual sequence of customers. This methodology is consistent with the fundamental principles of reinforcement learning (Sutton & Barto, 2018), where the agent (seller) must balance exploration (Step 1) and exploitation (Step 2) in consecutive stages to optimize the objective function.

The structure of Ψ^{SPSA} is similar to the KW pricing policy proposed by Hong et al. (2020), denoted by Ψ^{KW}. The common goal is to approximate the maximum value of the expected revenue function Eq. (1) in cases where direct gradient information cannot be easily acquired from sales data. Both algorithms fall under the umbrella of the zeroth-order optimization methods (see, e.g., Chen & Liu, 2019; Ghadimi & Lan, 2013). The primary difference between Ψ^{SPSA} and Ψ^{KW} resides in their respective approaches to tackle the issue of incomplete learning. Specifically, we utilize an SPSA-based updating scheme, while Ψ^{KW} relies on the forward finite difference method. For high-dimensional price vectors, i.e., a large number of products whose prices need to be determined by the seller, Ψ^{KW} may become computationally demanding. In particular, each price experimentation step in Ψ^{KW} requires d revenue function evaluations at the perturbed price vectors. This, together with the current price estimate, results in a total of $d + 1$ different price evaluations at each stage. In contrast, our algorithm takes a different approach by simultaneously altering all components of the price vector in random directions (i.e., $\pm c_n\boldsymbol{\Delta}_n$). We show in Section 4 that this achieves the same optimal regret bound as the forward-finite-difference-based approach but with only two revenue function evaluations needed during each stage, regardless of the price dimension. The algorithm is also much easier to implement than the parametric approaches, especially in high-dimensional settings. There are very few hyperparameters to tune, and we will give the optimal choice of parameter structures in the next section.

In addition to the SPSA pricing policy, we further propose the following two modified versions to improve the empirical performance of the algorithm:

Adding a projection operator.

According to the original recursion Eq. (3), there is no "truncation" when updating the price. However, in practical applications, it is always necessary to establish price limits, e.g., both lower and upper bounds, for any given product. That is, there exist two non-negative constants l_i and u_i such that $p_{n,i} \in [l_i, u_i]$ for all $n \in \{1, 2, \ldots\}$ and $i \in \{1, \ldots, d\}$. Thus the truncated Ψ^{SPSA} policy uses the following recursion when updating price estimates:

$$\mathbf{p}_{n+1} = \mathcal{P}_{[l_1,u_1]\times\cdots\times[l_d,u_d]}\left(\mathbf{p}_n + a_n \frac{F^+ - F^-}{2c_n} \boldsymbol{\Delta}_n^{-1}\right),$$

where recall that $\mathcal{P}$ is the projection operator with

$$\mathcal{P}_{[l_1,u_1]\times\cdots\times[l_d,u_d]}(\mathbf{x}) = (\min\{u_1, \max\{l_1, x_1\}\}, \ldots, \min\{u_d, \max\{l_d, x_d\}\})^{\mathrm{T}}$$

for any $\mathbf{x} = (x_1, \ldots, x_d)^{\mathrm{T}}$. Incorporating such a projection constrains the search of the algorithm within a predetermined price range, which typically results in enhanced algorithm stability; cf., e.g., Andradóttir (1995) and Nemirovski, Juditsky, Lan, and Shapiro (2009).

Using common random numbers.

Another enhancement is to use a variance reduction technique for reducing the variance of the finite-difference term in Eq. (3). In particular, because Eq. (3) involves the difference of two random variables, namely $F^+ - F^-$, it is natural to consider the use of common random numbers (CRN), i.e., the same stream of random numbers, for generating F^+ and F^- (see, e.g., Law, 2015, Chapter 11). Specifically, let $F_{CRN}^+ = F(\mathbf{p}_n + c_n\boldsymbol{\Delta}_n; U_n)$ and $F_{CRN}^- = F(\mathbf{p}_n - c_n\boldsymbol{\Delta}_n; U_n)$ be the output random revenue obtained using the same input random number stream U_n under the perturbed price $\mathbf{p}_n + c_n\boldsymbol{\Delta}_n$ and $\mathbf{p}_n - c_n\boldsymbol{\Delta}_n$, respectively. The CRN version of Ψ^{SPSA} simply works by replacing F^+ and F^- in (3) with F_{CRN}^+ and F_{CRN}^-, thus the price updating scheme is given by

$$\mathbf{p}_{n+1} = \mathbf{p}_n + a_n \frac{F_{CRN}^+ - F_{CRN}^-}{2c_n} \boldsymbol{\Delta}_n^{-1}.$$

In the context of dynamic pricing with demand learning, this means that the seller only modifies the price during the price experimentation step while keeping all other conditions unchanged, with the aim of achieving a variance reduction effect based on CRN.

In light of the discussion on the two aforementioned modifications, we denote the enhanced variant of the Ψ^{SPSA} algorithm by $\mathscr{E}\Psi^{\text{SPSA}}$. Its performance will be discussed through numerical experiments in Section 6.

4. Finite-time regret analysis

In this section, we carry out a rigorous regret analysis of the SPSA pricing policy. To begin with, we define $(\otimes, \mathscr{F}, \mathsf{P})$ as the probability space induced by Ψ^{SPSA}, where $\otimes$ refers to the set of all potential sample trajectories that can be observed during the execution of the algorithm, $\mathscr{F}$ is the σ-field of subsets of $\otimes$, and P is a probability measure on $\mathscr{F}$. For a vector $\mathbf{v}$, let $\|\mathbf{v}\|$ be the Euclidean norm of $\mathbf{v}$. For a matrix A, we use $\|A\|$ and $\|A\|_\infty$ to denote the matrix norm induced by the Euclidean norm (also known as the spectral norm) and the infinite norm (i.e., $\|A\|_\infty = \max_i \sum_j |A_{i,j}|$), respectively. Let $\partial_j f(\mathbf{p}) = \partial_j f(\mathbf{x})|_{\mathbf{x}=\mathbf{p}}$ where $\partial_j f$ denotes the partial derivative of the function f with respect to the jth argument and the definition is for notational convenience. The gradient and Hessian matrix of the function f at $\mathbf{p}$ is denoted as $\nabla f(\mathbf{p})$ and $\nabla^2 f(\mathbf{p})$, respectively. For any two real-valued functions $x(n)$ and $y(n)$, we write $x(n) = O(y(n))$ if $\limsup_{n\to\infty} x(n)/y(n) < \infty$, $x(n) = o(y(n))$ if $\lim_{n\to\infty} x(n)/y(n) = 0$, and $x(n) = \Omega(y(n))$ if $\liminf_{n\to\infty} x(n)/y(n) > 0$. Let $\lceil\cdot\rceil$ denote the ceiling function. For any $\mathbf{p} \in \mathbb{R}^d$, let $\xi = F(\mathbf{p}) - f(\mathbf{p})$. Note that the noise term ξ may depend on $\mathbf{p}$, but we drop this dependency to simplify the notation. To analyze the regret of the proposed pricing policy, $R(T, \Psi^{\text{SPSA}})$, we impose the following conditions on the model and algorithm parameters:

Assumptions.

A1 (Concavity and Smoothness) *The function f is twice differentiable. There exist positive constants K, L_1, and L_2 such that $K < L := \max\{L_1, L_2\}$ and*

(a)

$$(\mathbf{p} - \mathbf{p}^*)^{\mathsf{T}} \nabla f(\mathbf{p}) \le -K\|\mathbf{p} - \mathbf{p}^*\|^2, \quad \forall \mathbf{p},$$

where $\mathbf{p}^* = (p_1^*, \ldots, p_d^*)^{\mathsf{T}}$ is the unique optimal price satisfying $\nabla f(\mathbf{p}^*) = 0$.

(b)

$$\|\nabla f(\mathbf{p})\| \le L_1 \|\mathbf{p} - \mathbf{p}^*\|, \quad \forall \mathbf{p}.$$

(c) For all $t_1, t_2 \in (0, 1)$, with probability one

$$\max\left\{ \sup_n \|\nabla^2 f(\mathbf{p}_n + t_1 c_n \boldsymbol{\Delta}_n)\|_\infty, \sup_n \|\nabla^2 f(\mathbf{p}_n - t_2 c_n \boldsymbol{\Delta}_n)\|_\infty \right\} \le L_2.$$

A2 (Bounded Variance) *There exists a positive constant σ such that with probability one,*

$$\sigma^2 = \sup_n \mathsf{Var}[F(\mathbf{p}_n + c_n \boldsymbol{\Delta}_n) - F(\mathbf{p}_n - c_n \boldsymbol{\Delta}_n) | \mathbf{p}_n, \boldsymbol{\Delta}_n] < \infty.$$

A3 (Random Direction Sequence) *(a) The components of $\boldsymbol{\Delta}_n$ are d mutually independent zero-mean random variables. The inverse moment $\mathsf{E}[|1/\boldsymbol{\Delta}_{n,i}|]$ is finite for all n and i, and the sequence of $\boldsymbol{\Delta}_n$'s are independent and identically distributed (i.i.d.) with $\boldsymbol{\Delta}_n$ independent of $\{\mathbf{p}_i, i = 1, \ldots, n\}$ for all n.*

(b) There exist positive constants $B_1 \ge 1$ and $B_2 \ge 1$ such that $\|\boldsymbol{\Delta}_n\|^2 \le B_1$ and $\mathsf{E}[\|\boldsymbol{\Delta}_n^{-1}\|^2] \le B_2^2$ for all n.

A4 (Step Size and Perturbation Size) *$\{a_n\}$ and $\{c_n\}$ are positive and bounded sequences. There exist positive constants C, τ_1, and τ_2 such that*

(a) $a_n/c_n^2 \le \left(a_{n+1}/c_{n+1}^2\right)(1 + Ca_{n+1})$ for all $n \ge 1$.
(b) $c_n^2 \le c_{n+1}^2 (1 + Ca_{n+1})$ for all $n \ge 1$.
(c) $a_n \to 0$ as $n \to \infty$.
(d) either (i) $c_n^4/a_n \le \tau_1$ or (ii) $c_n^4/a_n \ge \tau_2$ for all $n \ge 1$.

We briefly comment on these assumptions. A1(a) is satisfied if f is strongly concave so that the gradient-based iterative algorithm may take successive steps in the direction where the revenue function is increasing. A1(b) imposes a linearly increasing envelope on the gradient, which holds if the domain is compact and the function f is twice continuously differentiable. A1(c) relaxes the bounded third derivative assumption made by Spall (1992) and only requires the boundedness of second order derivatives.

A2 assumes a bounded variance of the noise term when constructing the SPSA-based stochastic gradient, which frequently appears in the SA literature (Kushner & Yin, 2003). Both A3 and A4 are conditions on the input parameters. A3 precludes the use of common distributions such as the Gaussian and uniform distributions due to their concentration of probability mass in the proximity of 0. A common choice of $\boldsymbol{\Delta}_n$ is the Rademacher (or symmetric Bernoulli) random vectors, with each component of $\boldsymbol{\Delta}_n$ taking value $+1$ or -1 with equal probability $1/2$. A4 is taken from Broadie et al. (2011). The condition is quite general and includes not only polynomial-like sequences (e.g., for some $\alpha > 0$ and $\gamma > 0$, setting $a_n = \alpha/n^a$ and $c_n = \gamma/n^c$ where $0 < a \le 1$ and $c > 0$), but also allows for a much broader class of sequences such as $a_n = \alpha/n$ and $c_n = \gamma/\log(n)$; cf. Broadie et al. (2011, Remark 3). We note the assumption "for all $n \ge 1$" in A4 is primarily made for the sake of simplicity. It can be substituted with the statement "for n sufficiently large" by making minor adjustments.

Given the above assumptions, we can prove the following theorem, which gives a finite-time upper bound on the MSE of the pricing policy Ψ^{SPSA}. The analysis is based on that of Broadie et al. (2011) with appropriate modifications tailored to our setting.

Theorem 1. *Assume A1–A4 hold and $C < 2K$. Then there exist constants $C_1 > 0$ and $C_2 > 0$ such that for all $n \ge 1$,*

$$\mathsf{E}\left[\|\mathbf{p}_{n+1} - \mathbf{p}^*\|^2\right] \le \begin{cases} C_1 a_n/c_n^2 & \text{if } c_n^4/a_n \le \tau_1, \\ C_2 c_n^2 & \text{if } c_n^4/a_n \ge \tau_2. \end{cases} \tag{4}$$

Proof. See Appendix A for the proof. □

Note that the constants C_1 and C_2 can be identified explicitly (see Eq. (A.6) and Eq. (A.7) in the proof), both depend polynomially on the price vector dimension d. The theorem suggests that the rate of convergence of the price estimates is associated with the choice of the sequences $\{a_n\}$ and $\{c_n\}$. Clearly, if we specify the form of a_n and c_n such that $a_n = o\left(c_n^2\right)$ and $c_n \to 0$ (as well as satisfying Assumption A4), then the sequence of price estimates is guaranteed to converge to the true optimal price in terms of the MSE.

Based on the upper bounds presented in Theorem 1, we can adopt a methodology analogous to that employed by Hong et al. (2020, Theorem 2) for regret analysis. Specifically, we initiate the procedure by deriving an upper bound for the regret in each price experimentation step, and subsequently extend it to cover all periods. The regret upper bound is achieved through a careful choice of parameters a_n and c_n, followed by substituting the summation with an integral, as detailed in the proof of Corollary 1.

A finite-time upper bound on the regret of Ψ^{SPSA} is provided in Proposition 1 below; see Appendix B for a proof.

Proposition 1. *Assume A1–A4 hold and $C < 2K$. Then for all $T \ge 1$,*

$$R(T, \Psi^{SPSA}) = \begin{cases} \sum_{n=1}^{\lceil T/2 \rceil} O\left(a_n/c_n^2\right) + O\left(\sqrt{a_n}\right) & \text{if } c_n^4/a_n \le \tau_1, \\ \sum_{n=1}^{\lceil T/2 \rceil} O\left(c_n^2\right) & \text{if } c_n^4/a_n \ge \tau_2. \end{cases}$$

Note that all omitted constants contained in the big-O notations above can be identified explicitly in the proof. Proposition 1 reveals the relationship between the regret of Ψ^{SPSA} and tuning sequences $\{a_n\}$ and $\{c_n\}$. Roughly speaking, a_n and c_n reflect the degree of exploitation and exploration in the algorithm, respectively. In particular, a large a_n value implies that the price estimate $\mathbf{p}_n$ changes quickly over the iterations, in which case the perturbation size c_n should decay sufficiently slowly to ensure that the regret cannot increase too fast. Indeed, we can optimize the order of the regret bound by properly selecting the forms of a_n and c_n, which is given by Corollary 1.

Corollary 1. *Assume A1–A4 hold and $C < 2K$. Let $a_n = \alpha n^{-1}$ and $c_n = \gamma n^{-1/4}$ with $\alpha > \frac{\sqrt{2}-1}{K}$ and $\gamma > 0$. Then there exist constants $\tilde{C}_1 > 0$ and $\tilde{C}_2$ such that $R(T, \Psi^{SPSA}) \le \tilde{C}_1\sqrt{T} + \tilde{C}_2$ for all $T = 1, 2, \ldots$.*

Proof. See Appendix C for the proof. □

The constants $\tilde{C}_1$ and $\tilde{C}_2$ are explicitly identified in the proof; refer to Eq. (C.1). Combined with the definition of C_1 and C_2 in Theorem 1, it follows that the regret has a polynomial dimension dependency. In several dynamic pricing with demand learning settings, it has been proved that the lower bound of the expected cumulative regret is $\Omega(\sqrt{T})$ for arbitrary pricing policy (see, e.g., Broder & Rusmevichientong, 2012; Keskin & Zeevi, 2014), hence $O(\sqrt{T})$ is the best possible growth rate of the regret. Corollary 1 has shown that our pricing policy can achieve a finite-time regret bound of $O(\sqrt{T})$ by setting $a_n = \alpha n^{-1}$ and $c_n = \gamma n^{-1/4}$ for some positive constants α and γ. This is a finite-time result that is stronger than the asymptotic bound and hence implies the asymptotic result such as Cope (2009).

Remark 1. The expressions of $\{a_n\}$ and $\{c_n\}$ in Corollary 1, i.e., $a_n = \alpha n^{-1}$ and $c_n = \gamma n^{-1/4}$, are the same as the parameter selection employed in Ψ^{KW} presented in Hong et al. (2020). However, unlike Hong et al. (2020) (as well as essentially most previous literature), we refrain from explicitly specifying the exact forms of the two sequences in Theorem 1 and Proposition 1. Instead, we follow the methodology outlined in Broadie et al. (2011) and derive this parameter structure from general bounds.

Remark 2. From Eq. (4), we obtain that the order of MSE is $O(n^{-\frac{1}{2}})$ when assigning values to a_n and c_n as αn^{-1} and $\gamma n^{-1/4}$, respectively. Note that by imposing stronger smoothness conditions, e.g., third-order differentiability of f as stated in Spall (1992), we are able to derive the following result similar to Eq. (4):

$$\mathbf{E}\left[\|\mathbf{p}_{n+1} - \mathbf{p}^*\|^2\right] = \begin{cases} O\left(\frac{a_n}{c_n^2}\right) & \text{if } c_n^6/a_n \le \tau_1', \\ O\left(c_n^4\right) & \text{if } c_n^6/a_n \ge \tau_2', \end{cases} \tag{5}$$

where τ_1' and τ_2' are positive constants. Following the same argument as in the proofs of Proposition 1 and Corollary 1, we can set $a_n = \alpha' n^{-1}$ and $c_n = \gamma' n^{-1/6}$ for some positive constants α' and γ', and the optimal MSE given by Eq. (5) is then of order $O(n^{-\frac{2}{3}})$. However, this improvement does not affect the growth rate of the regret. By Eq. (B.1), the expected regret at each stage, denoted as $\varphi(n)$, is upper bounded in order by the maximum of the MSE and $O(c_n^2)$. If the function f has bounded third derivatives with $c_n = \gamma' n^{-1/6}$, then $\varphi(n)$ is dominated by $O(c_n^2)$ and $\varphi(n) = O(n^{-\frac{1}{3}})$, yielding $R(T, \Psi^{\text{SPSA}}) = O(T^{\frac{2}{3}})$. Therefore, even when the revenue function is thrice differentiable, we also need to set $a_n = \alpha n^{-1}$ and $c_n = \gamma n^{-1/4}$ to achieve the optimal regret order $O(\sqrt{T})$.

We note that all the convergence proofs are also applicable if a projection operator is used in the price updating step, as in $\mathscr{C}\Psi^{\text{SPSA}}$. In particular, if we assume the optimal price of each product i lies in the interior of $[l_i, u_i]$, i.e., $l_i < p_i^* < u_i$, then all theoretical results (Theorem 1, Proposition 1, and Corollary 1) still hold. The proof of truncated Ψ^{SPSA} differs from that of the projection-free Ψ^{SPSA} only in its use of the fact that the projection operator is non-expanding. Hence for all $\mathbf{p} \in \mathbb{R}^d$,

$$\|\mathscr{P}(\mathbf{p}) - \mathbf{p}^*\| = \|\mathscr{P}(\mathbf{p}) - \mathscr{P}(\mathbf{p}^*)\| \le \|\mathbf{p} - \mathbf{p}^*\|.$$

Using this observation and recalling the proof of Theorem 1, now we have

$$\begin{aligned} Z_{n+1} := \|\mathbf{p}_{n+1} - \mathbf{p}^*\|^2 &= \left\| \mathscr{P}\left(\mathbf{p}_n + a_n \frac{F^+ - F^-}{2c_n} \boldsymbol{\Delta}_n^{-1}\right) - \mathbf{p}^* \right\|^2 \\ &\le \left\| \mathbf{p}_n - \mathbf{p}^* + a_n \frac{F^+ - F^-}{2c_n} \boldsymbol{\Delta}_n^{-1} \right\|^2. \end{aligned}$$

The remainder of the proof is identical to that of the projection-free algorithm.

5. Extensions

5.1. Non-stationary demand

The demand for a seller's product may become non-stationary under various business settings, as influenced by exogenous factors such as macroeconomic issues and fashion trends. The aim of this subsection is to expand the foregoing discussion by formulating and exploring a dynamic, time-varying demand environment, where the seller confronts a non-stationary black-box online learning problem. Under such a setting, the static regret employed in Eq. (2) to assess the algorithm's performance is no longer appropriate. Specifically, instead of a single unknown revenue function f as in the stationary demand setting, there is now a sequence of functions $\{f_t, t = 1, 2, \ldots, T\}$ such that at every period t, the seller selects a price $\mathbf{p}_t$ and then observes a (unbiased) noisy feedback $F_t(\mathbf{p}_t)$ to the true function value $f_t(\mathbf{p}_t)$. This gives rise to the so-called dynamic regret (Besbes, Gur, & Zeevi, 2015):

$$\mathscr{R}(T, \Psi) := \sup_{\{f_t\} \in \mathscr{V}} \left\{ \sum_{t=1}^{T} f_t(\mathbf{p}_t^*) - \mathbf{E}\Big[\sum_{t=1}^{T} f_t(\mathbf{p}_t)\Big] \right\}, \tag{6}$$

where $\mathbf{p}_t^* \in \arg\max_{\mathbf{p}} f_t(\mathbf{p})$ for all $t \in \{1, 2, \ldots, T\}$, and $\mathscr{V}$ is the set of admissible revenue function sequences whose precise definition is given later in Eq. (8). The performance metric adopted in our analysis is closely linked to the dynamic regret, which has been employed in adversarial online convex optimization (OCO) problems (see, e.g., Hazan et al., 2016). In particular, the efficacy of a policy Ψ in the OCO context is gauged through the single best action in hindsight, referred to as OCO regret. This evaluation is defined by the following expression:

$$\mathscr{R}(T, \Psi) := \sup_{\{f_t\} \in \mathscr{V}} \left\{ \max_{\mathbf{p}} \Big\{ \sum_{t=1}^{T} f_t(\mathbf{p}) \Big\} - \mathbf{E}\Big[\sum_{t=1}^{T} f_t(\mathbf{p}_t)\Big] \right\}, \tag{7}$$

which mainly differs from the dynamic regret (6) in the order of the sum and max operators on the right-hand sides.

A key insight gleaned from Besbes et al. (2015) is that a policy exhibiting favorable performance regarding the single best action in hindsight within the adversarial OCO framework, as measured by Eq. (7), can be adapted to the stochastic non-stationary environment to yield good performance, as evaluated by Eq. (6). Such an adaptation can be facilitated through a simple "restarting" procedure. Therefore, for analytical tractability, we focus on presenting the algorithm's performance with respect to the OCO regret. Dynamic regret can be readily obtained by applying the established result in the literature (see Besbes et al., 2015, Proposition 2), thereby providing a comprehensive understanding of the algorithm's efficacy.

Based on the estimated gradient step (EGS) method described by Besbes et al. (2015, Section 5.2), we introduce an SPSA-type algorithm that is rate optimal in the adversarial setting. Throughout this section, we assume the decision space $\mathscr{P}$ is a convex, compact, nonempty set in $\mathbb{R}^d$. Denote by $\mathscr{P}_\delta$ the δ-interior of $\mathscr{P}$ for any $\delta > 0$, i.e., $\mathscr{P}_\delta = \{\mathbf{p} : \mathbb{B}_\delta(\mathbf{p}) \subseteq \mathscr{P}\}$, where $\mathbb{B}_\delta(\mathbf{p})$ is a ball with radius δ,

centered at $\mathbf{p}$. For any $\mathbf{y} \in \mathbb{R}^d$, let $\mathscr{P}_{\mathscr{X}}(\mathbf{y}) = \arg\min_{\mathbf{x}\in\mathscr{X}} \|\mathbf{x}-\mathbf{y}\|$ signify the projection operator on any given set $\mathscr{X}$. For $k \in \{1,2,\ldots,d\}$, we denote by $\mathbf{e}^{(k)}$ the unit vector with 1 in the kth element. The algorithm description of SPSA-type EGS is provided as follows:

SPSA-type EGS algorithm : $\Psi^{\text{SPSA-EGS}}$

Step 0: Initialization

Select three decreasing sequences of positive real numbers $\{\alpha_t\}$, $\{h_t\}$, and $\{\delta_t\}$. Set the initial price $\mathbf{p}_1 = \mathbf{Z}_1$ in $\mathscr{P}$ and period counter $t = 0$.

Step 1: Price Experimentation

Generate a realization of the Rademacher random vector $\phi_t \in \mathbb{R}^d$.

- ▷ Compute a stochastic gradient estimate $\hat{\nabla}_{h_t} f_t(\mathbf{Z}_t) := \frac{F_t(\mathbf{Z}_t+h_t\phi_t)}{h_t}\phi_t$.
- ▷ Update $\mathbf{Z}_{t+1} = \mathscr{P}_{\mathscr{P}_{\delta_t\sqrt{d}}}\left(\mathbf{Z}_t + \alpha_t \hat{\nabla}_{h_t} f_t(Z_t)\right)$.

Step 2: Price Updating

Compute

$$\mathbf{p}_{t+1} = \mathbf{Z}_{t+1} + h_{t+1}\phi_t.$$

Set $t = t+1$. If $t < T$, then go to Step 1; otherwise, the algorithm outputs all price estimates $\{\mathbf{p}_t\}$ and terminates.

The price experimentation step of $\Psi^{\text{SPSA-EGS}}$ resembles the classical one-sided SPSA (Spall, 1997), with the random perturbation vectors selected as Rademacher vectors. This departs significantly from the EGS algorithm presented in Besbes et al. (2015). In their work, the perturbation vector is a single unit vector $\mathbf{e}^{(k)}$ with $k \in \{1,2,\ldots,d\}$, perturbing only one component of the price vector in each period. This difference is analogous to the distinction between our SPSA pricing policy and the KW pricing policy under the basic model setting.

To investigate the theoretical performance of $\Psi^{\text{SPSA-EGS}}$, we first transform the revenue maximization problem into its equivalent minimization counterpart by considering the cost function $g_t(\cdot) := -f_t(\cdot)$. Thus, the OCO regret of $\Psi^{\text{SPSA-EGS}}$ can be expressed as

$$\mathscr{R}(T,\Psi^{\text{SPSA-EGS}}) = \sup_{\{g_t\}\in\mathscr{V}} \left\{ \mathbf{E}\Big[\sum_{t=1}^{T} g_t(\mathbf{p}_t)\Big] - \min_{\mathbf{p}} \Big\{\sum_{t=1}^{T} g_t(\mathbf{p})\Big\} \right\}.$$

We focus on the class of strongly convex revenue functions $\mathscr{F}_s$ and define the set of admissible function sequences as:

$$\mathscr{V} = \left\{ \{g_t, t = 1,2,\ldots,T\} \in \mathscr{F}_s : \sum_{t=2}^{T} \sup_{\mathbf{p}\in\mathscr{P}^*} |g_t(\mathbf{p}) - g_{t-1}(\mathbf{p})| \le V_T \right\}, \quad (8)$$

where $\mathscr{P}^*$ is the convex hull of the minimizers, i.e., $\mathscr{P}^* = \{\mathbf{p} \in \mathbb{R}^d : \mathbf{p} = \sum_{t=1}^{T} \lambda_t \mathbf{p}_t^*, \sum_{t=1}^{T} \lambda_t = 1, \lambda_t \ge 0, \forall t\}$ and V_T is the variation budget satisfying $1 \le V_T \le T$ for all $T \ge 1$. We impose the following conditions on the model:

Assumptions.

A5 (Boundness, Strongly Convexity and Smoothness) *There exist positive constants G and H such that for all $\mathbf{p} \in \mathscr{P}$ and period $t \in \{1,\ldots,T\}$,*

(a)

$$|g_t(\mathbf{p})| \le G, \qquad \|\nabla g_t(\mathbf{p})\| \le G.$$

(b)

$$H\mathbf{I}_d \preceq \nabla^2 g_t(\mathbf{p}) \preceq G\mathbf{I}_d,$$

where $\mathbf{I}_d$ denotes the d-dimensional identity matrix.

A6 (Bounded Variance) *There exists a positive constant ρ such that for all $t \in \{1,\ldots,T\}$,*

$$\sup_{\mathbf{p}\in\mathscr{P}} \mathbf{E}\left[\left(F_t(\mathbf{p})\right)^2\right] \le G^2 + \rho^2,$$

where G is defined in A5.

We note that the "universal" constant G in A5 primarily serves as the purpose to reduce notational burden. A5(b) indicates that the function g_t is H-strongly convex and G-smooth. A6 is slightly more stringent than A2 and is expected to hold under many practical situations.

Given the above conditions, we have the following theorem, which shows that under appropriate choices of the parameters α_t, δ_t, and h_t, the SPSA-type EGS algorithm achieves a regret of order $\sqrt{T}$ with respect to a single best action in the adversarial setting. The analysis is partially based on the work of Besbes et al. (2015) but employs different techniques in the regret analysis, which could be of independent interest.

Theorem 2. *Assume A5 and A6 hold. Let $a_t = \frac{2}{Ht}$ and $\delta_t = h_t = a_t^{1/4}$ for all $t \in \{1,2,\ldots,T\}$. Then there exist positive constants $\bar{C}_1$ and $\bar{C}_2$, independent of T, such that for all $T \ge 1$,*

$$\mathscr{R}(T,\Psi^{\text{SPSA-EGS}}) \le \bar{C}_1\sqrt{T} + \bar{C}_2.$$

Proof. See Appendix D for the proof. □

By leveraging established results in the literature, the result of Theorem 3 can be carried over from the adversarial OCO setting to the non-stationary stochastic setting. Such a rate can be shown to be the best possible under additional mild conditions; cf. Besbes et al. (2015, Theorem 5). In particular, the dynamic regret of the proposed algorithm in the context of a changing demand environment is provided in Proposition 2. As previously noted, the proof directly applies (Besbes et al., 2015, Proposition 2) and is hence omitted.

Proposition 2. *Consider the policy π defined by the restarting procedure, as described in Besbes et al. (2015, Section 3), with the SPSA-type EGS algorithm as a subroutine using a batch size of $\lceil (T/V_T)^{2/3} \rceil$ in the restarting procedure. Assume that all conditions in Theorem 2 are satisfied. Then, there exists a constant $\bar{C} > 0$ such that for all $T \ge 2$,*

$$\mathscr{R}(T,\pi) \le \bar{C} V_T^{1/3} T^{2/3}.$$

5.2. Finite inventory

Our previous discussion is based on the assumption that there is no resource constraint and an infinite inventory is available to the seller. This assumption, however, is unlikely to be satisfied in many practical situations such as airline ticket booking, hotel room reservation, as well as the selling of perishable commodities. Thus, the goal of this section is to examine how the SPSA pricing policy proposed in Section 3 could be modified/extended to handle resource constraints. We consider the classical network revenue management model framework proposed by Gallego and Van Ryzin (1997). In particular, there are m types of resources used to produce d products. Let the resource consumption matrix be

$$\mathbf{A} = \begin{bmatrix} A_{11} & \cdots & A_{1d} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{md} \end{bmatrix},$$

which each element $A_{ij} \in \mathbb{N}$ indicates the units of resource $i \in \{1,2,\ldots,m\}$ that need to be consumed to produce one unit of product $j \in \{1,2,\ldots,d\}$. We denote by $\mathbf{x}_0 = (x_{0,1}, x_{0,1}, \ldots, x_{0,d})^{\mathrm{T}}$ the initial capacity levels of all resources at the beginning of the selling season. Under the standard model assumption, inventory replenishment is precluded throughout the entire season. This can often be justified in various industrial settings such as in hotels and airlines, where the

replenishment of resources during the selling season is often deemed excessively costly or, in some cases, infeasible. The selling season is divided into T discrete periods, with one customer arriving in each period. Each customer purchases at most one unit of a product from d alternatives. The seller discontinues the sale of all products for the remainder of the season when the inventory of at least one resource is depleted. Let $\mathbf{Q}(\mathbf{p}) = (Q_1(\mathbf{p}), Q_2(\mathbf{p}), \ldots, Q_d(\mathbf{p}))^{\mathrm{T}}$ represent the purchase probability vector under the price vector $\mathbf{p}$. For each $j \in \{1, 2, \ldots, d\}$, $Q_j(\mathbf{p})$ can be interpreted as the demand rate or market share for product j. Thus $\mathbf{Q}(\mathbf{p})$ lies in a d-dimensional simplex, that is, $\mathbf{Q}(\mathbf{p}) \in \mathcal{D}_q := \left\{(x_1, x_2, \ldots, x_d)^{\mathrm{T}} \middle| \sum_{i=1}^{d} x_i \le 1, x_i \ge 0, i = 1, \ldots, d\right\}$. At the end of time period t, the seller acquires the actual sales data denoted as $\mathbf{Y}_t = (Y_{t,1}, Y_{t,2}, \ldots, Y_{t,d})^{\mathrm{T}} \in \left\{(y_1, y_2, \ldots, y_d)^{\mathrm{T}} \middle| \sum_{i=1}^{d} y_i \le 1, y_i \in \{0, 1\}, i = 1, \ldots, d\right\}$. Note that the exact form of the demand rate function is unknown to the seller, and only the realized demand is available. Due to this uncertainty, Besbes and Zeevi (2012) refer to the class of problems as "blind" network revenue management.

Denote by $J^*(\mathbf{x}_0, T)$ the optimal expected total revenue that could be attained (while adhering to the resource constraints) if the seller had full information about the market share function $\mathbf{Q}(\cdot)$. A pricing policy performs well if its T-period expected revenue, $\mathsf{E}\left[\sum_{t=1}^{T} \mathbf{p}_t^{\mathrm{T}} \mathbf{Y}_t\right]$, can be made as close to $J^*(\mathbf{x}_0, T)$ as possible. Unfortunately, achieving this would require solving a stochastic dynamic program, which could be computationally intractable. So we instead develop a heuristic method based on the following result. Specifically, consider the (deterministic) optimization problem

$$\begin{aligned} \underset{\mathbf{p}}{\text{maximize}} \quad & T\mathbf{p}^{\mathrm{T}}\mathbf{Q}(\mathbf{p}) \\ \text{subject to} \quad & T\mathbf{A}\mathbf{Q}(\mathbf{p}) \le \mathbf{x}_0, \\ & \mathbf{Q}(\mathbf{p}) \in \mathcal{D}_q. \end{aligned} \tag{9}$$

It has been shown (see, Gallego & Van Ryzin, 1997, Lemma 1) that the optimal value of (9) serves as an upper bound on $J^*(\mathbf{x}_0, T)$, i.e.,

$$J^*(\mathbf{x}_0, T) \le T\mathbf{p}_*^{\mathrm{T}}\mathbf{Q}(\mathbf{p}_*), \tag{10}$$

where $\mathbf{p}_*$ is the (unknown) optimal solution to (9). The idea is thus to develop an SPSA-based algorithm for approximately solving (9) to obtain a well-performing learning policy.

Due to the black-box nature of $\mathbf{Q}(\cdot)$, neither the objective nor the constraints are known in the optimization problem (9). Instead, one can only obtain noisy responses when querying their values at a given $\mathbf{p}$. To the best of our knowledge, developing a constrained variant of the SPSA algorithm for addressing such problems remains an open challenge. Existing studies predominantly concentrate on situations where the constraints are known (see, e.g., Shi & Spall, 2021). So we propose two variants of constrained SPSA-based heuristics and conduct numerical tests to evaluate their effectiveness in tackling blind revenue management problems. The detailed descriptions of these heuristics are presented below.

Projection SPSA heuristic : $\Psi^{\mathrm{SPSA-Proj.}}$

Step 0: Initialization

Select a step-size sequence $\{a_n\}$, a perturbation-size sequence $\{c_n\}$, a negative sequence $\{h_n\}$, and a starting price $\mathbf{p}_1$. Choose two positive integers n' and s' such that $n' \times s' < T$.

Step 1: Demand Rate Function Regression

Select s' sample points in the domain of prices, denoted as $\{S_1, S_2, \ldots, S_{s'}\}$.

- ▷ Calculate the empirical demand rate $D'_i = (D'_{i,1}, D'_{i,2}, \ldots, D'_{i,d})^{\mathrm{T}}$ for each price S_i by considering its occurrence over n' time periods:

$$D'_{i,j} = \frac{\#\text{times the } j\text{th product is selected under price } S_i}{n'}, \qquad j = 1, 2, \ldots, d.$$

- ▷ Apply a regression model to fit a regression function $\hat{\mathbf{Q}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$, train it with independent variables $\{S_1, S_2, \ldots, S_{s'}\}$ and dependent variables $\{D'_1, D'_2, \ldots, D'_{s'}\}$.

Set period counter $t = n' \times s'$ and stage counter $n = 1$.

Step 2: Price Experimentation

Generate a realization of the random vector $\boldsymbol{\Delta}_n$.

- ▷ Set $t = t + 1$. Put $\tilde{\mathbf{p}}_t^+ = \mathbf{p}_n + c_n\boldsymbol{\Delta}_n$, obtain a demand observation $\mathbf{Y}_t^+$ and a realized revenue R^+, where $R^+ = (\tilde{\mathbf{p}}_t^+)^{\mathrm{T}}\mathbf{Y}_t^+$.
- ▷ Set $t = t + 1$. Put $\tilde{\mathbf{p}}_t^- = \mathbf{p}_n - c_n\boldsymbol{\Delta}_n$, obtain a demand observation $\mathbf{Y}_t^-$ and a realized revenue R^-, where $R^- = (\tilde{\mathbf{p}}_t^-)^{\mathrm{T}}\mathbf{Y}_t^-$.

Step 3: Price Updating

Compute

$$\begin{aligned} \bar{\mathbf{p}}_{n+1} &= \mathbf{p}_n + a_n \frac{R^+ - R^-}{2c_n}\boldsymbol{\Delta}_n^{-1}, \\ \mathbf{p}_{n+1} &= \mathcal{P}_{\Theta_{n+1}}(\bar{\mathbf{p}}_{n+1}), \end{aligned}$$

where the projection sub-problem used to estimate $\mathcal{P}_{\Theta_n}(\bar{\mathbf{p}}_n)$ is defined as

$$\begin{aligned} \underset{\mathbf{p}}{\text{minimize}} \quad & \|\mathbf{p} - \bar{\mathbf{p}}_n\|^2 \\ \text{subject to} \quad & T\mathbf{A}\hat{\mathbf{Q}}(\mathbf{p}) - \mathbf{x}_0 \le h_n, \\ & \hat{\mathbf{Q}}(\mathbf{p}) \in \mathcal{D}_q. \end{aligned} \tag{11}$$

Set $n = n+1$. If $t \le T$, then go to Step 2; otherwise, the algorithm outputs price estimates for all periods and terminates.

To utilize the constrained SPSA method effectively, it is imperative to have knowledge of the constraints. In Step 1, a regression-based method is used to fit the empirical data in order to provide an approximation of $\mathbf{Q}$. The subsequent steps 2 and 3 remain essentially unchanged compared to the SPSA pricing policy Ψ^{SPSA} in scenarios without inventory constraints. A main difference lies in the use of a projection operator, which is implemented in (11) to ensure the feasibility of the estimated price vectors. Note that the variable h_n is incorporated in (11) to account for the approximation error of $\hat{\mathbf{Q}}$. Also, it helps to project an iterate onto the strict interior of the constraint $T\mathbf{A}\hat{\mathbf{Q}}(\mathbf{p}) \le \mathbf{x}_0$, so that the perturbed price vectors $\tilde{\mathbf{p}}_t^{\pm} = \mathbf{p}_n \pm c_n\boldsymbol{\Delta}_n$ are well-defined. This could be achieved, for example, by setting $h_n = -2c_n \sup_n \|\boldsymbol{\Delta}_n\|$.

Penalty SPSA heuristic : $\Psi^{\mathrm{SPSA-Pena.}}$

Step 0: Initialization

Select a step-size sequence $\{a_n\}$, a perturbation-size sequence $\{c_n\}$, a positive sequence $\{r_n\}$, a bounded vector sequence $\{\lambda_n\}$ in $\mathbb{R}^d$, and a starting price $\mathbf{p}_1$. Choose two positive integers n' and s' such that $n' \times s' < T$.

Step 1-2: Same as $\Psi^{\mathrm{SPSA-Proj.}}$

Step 3: Price Updating

Compute

$$\mathbf{p}_{n+1} = \mathbf{p}_n + a_n \frac{R^+ - R^-}{2c_n}\boldsymbol{\Delta}_n^{-1} - a_n r_n \nabla p_n(\mathbf{p}_n),$$

where $p_n(\cdot)$ is a penalty function for all n. This is defined as

$$p_n(\cdot) = \frac{1}{2r_n^2}\sum_{j=1}^{m}\left\{\left[\max\left\{0, \lambda_{n,j} + r_n\left(TA_j^{\mathrm{T}}\hat{\mathbf{Q}}(\cdot) - x_{0,j}\right)\right\}\right]^2 - \lambda_{n,j}^2\right\},$$

where $\lambda_{n,j}$ is the jth component of λ_n and A_j is the jth row of the resource consumption matrix A.

Set $n = n+1$. If $t \le T$, then go to Step 2; otherwise, the algorithm outputs price estimates for all periods and terminates.

The primary distinction between $\Psi^{\mathrm{SPSA-Pena.}}$ and $\Psi^{\mathrm{SPSA-Proj.}}$ becomes evident in the price updating step. Following the approach outlined in Wang and Spall (2008), the basic concept of the penalty SPSA involves transforming the originally constrained optimization problem (9) into an unconstrained one by introducing a sequence of augmented Lagrangian functions p_n. During the price updating step, we

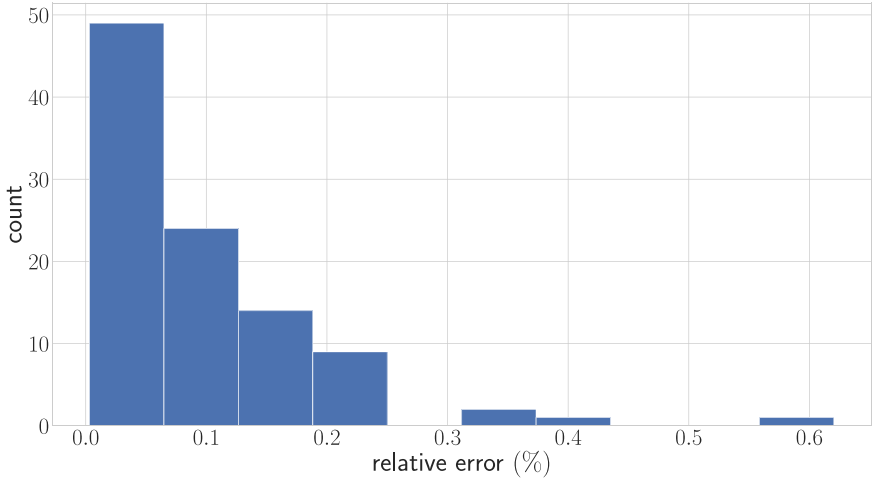

Fig. 1. Histogram of relative errors with $\alpha, \gamma \in \{1, 2, \ldots, 10\}$.

use the gradient of the penalty function p_n. However, when applying the simultaneous perturbation method to derive gradient estimates of p_n, an additional bias is typically introduced. This bias arises from the limitation that we can only measure the noisy value of $\mathbf{Q}(\cdot)$ but not the penalty function itself. As a result, we opt to fit a demand function with analytical expressions in Step 1, facilitating the derivation of the closed-form gradient of the penalty function. The numerical performance of these heuristics are presented in the next section.

6. Numerical experiments

This section presents numerical experiments that showcase the practical effectiveness of our pricing policy Ψ^{SPSA}. In Section 6.1, we empirically illustrate that the regret indeed grows at a rate of $\sqrt{T}$ and the price estimate converges to the optimal price for the linear demand model. In Section 6.2, we test the empirical performance of Ψ^{SPSA} and its enhanced version on different (possibly nonlinear) demand models. In Section 6.3 we compare the performance of our algorithms with that of Ψ^{KW}, the multi-product nonparametric policy suggested in Hong et al. (2020), under both low-dimensional and high-dimensional settings. We would like to note that our work does not aim to propose an algorithm that outperforms all existing algorithms. Rather, our goal is to develop an algorithm that combines a desirable theoretical convergence rate with effective facilitation of practical applications, even in high-dimensional problems. Finally, we conduct numerical experiments on the non-stationary demand setting and finite-inventory setting proposed in Section 5.

6.1. Illustration of the rate optimality of Ψ^{SPSA}

We follow the same multi-product linear demand model used in Keskin and Zeevi (2014) and Hong et al. (2020). The random demand vector in period t is expressed as

$$\mathbf{D}(\mathbf{p}_t) = \mathbf{a} + \mathbf{B}\mathbf{p}_t + \epsilon_t, \tag{12}$$

where $\mathbf{a} \in \mathbb{R}^d$ is an unknown vector with strictly positive components, $\mathbf{B} = [b_{ij}]$ is an unknown $d \times d$ matrix with strictly negative diagonal elements, and $|b_{ii}| > \sum_{j \neq i} |b_{ij}|$ for all $i \in \{1, \ldots, d\}$, and $\epsilon_t \in \mathbb{R}^d$ is a random demand vector whose components are i.i.d. normal random variables with mean zero and variance ς^2. Then the unique optimal price is given by $\mathbf{p}^* = -(\mathbf{B} + \mathbf{B}^{\mathrm{T}})^{-1}\mathbf{a}$ as a result of the first-order optimality condition (see Keskin & Zeevi, 2014).

We consider a simple two-product problem (i.e., $d = 2$). All model parameters are set according to Hong et al. (2020), in which $\mathbf{a} = (1.1, 0.7)$, $\mathbf{B} = \begin{bmatrix} -0.5 & 0.05 \\ 0.05 & -0.3 \end{bmatrix}$, and $\varsigma^2 = 0.01$. Note that in this case, we have set $\mathbf{B}$ as a symmetric matrix, signifying that all products share a symmetric sensitivity to the prices of other products. However, in general, $\mathbf{B}$ may take on an asymmetric form. The optimal price is given by $\mathbf{p}^* \approx (1.237, 1.373)^{\mathrm{T}}$. In our algorithm, we set the initial price $\mathbf{p}_0 = (0.75, 2.1)^{\mathrm{T}}$ and $a_n = 3n^{-1}$ and $c_n = n^{-1/4}$. We remark that while Corollary 1 provides the forms of a_n and c_n, determining the constant factors (i.e., α and γ) may require a trial-and-error approach. Similar to the majority of SA algorithms, the practical efficacy of the algorithm heavily relies on judicious selections of the iteration step size and perturbation size. A systematic approach involves initial testing of multiple constants over a wide range, establishing a reasonable guess on the parameter range, and subsequently fine-tuning the constants. In this context, we present our pilot runs with $\alpha, \gamma \in \{1, 2, \ldots, 10\}$—resulting in 100 parameter combinations. For each combination, the average price over 100 replications upon algorithm termination is recorded, and the relative error between the estimated price and the optimal price is calculated. The distribution of the 100 relative errors is depicted in Fig. 1, which shows promising performance under such a parameter range.

Fig. 2 plots the average cumulative regret, along with its 95% confidence interval (CI) over 100 simulation replications, as a function of $\sqrt{T}$, where the maximum value of T is 3000. The results presented in Fig. 2 confirm that, except for values of T that are very small (approximately $T \leq 25$ in our experiment), the regret of Ψ^{SPSA} increases linearly with $\sqrt{T}$. This observation empirically supports the theoretical growth rate of the regret bound established in Corollary 1. To further assess the applicability of Corollary 1, we have computed the values of $\tilde{C}_1$ and $\tilde{C}_2$ in Appendix E. Our calculation indicates that the theoretical values of $\tilde{C}_1$ and $\tilde{C}_2$ (see Eq. (E.1)) are significantly larger than those implied in Fig. 2. This discrepancy suggests that the constants given in Corollary 1 might be overly conservative, leaving room for further refinement and improvement. However, addressing this issue might require completely different proof techniques, which is beyond the scope of this paper. Additionally, Fig. 3 shows sample paths of the two sequences of the product prices generated at successive iterations (averaged over 100 replications), which clearly indicate the algorithm's fast convergent behavior.

In Appendix F, the performance of the algorithm is further tested on a 50-dimensional version of the example. Test results suggest that the algorithm scales well with problem dimension and has the potential to be a useful tool for solving high-dimensional practical pricing problems.

6.2. Ψ^{SPSA} and $\mathscr{C}\Psi^{SPSA}$ under nonlinear demand models

Motivated by Besbes and Zeevi (2015) and Hong et al. (2020), we proceed to investigate the empirical performance of Ψ^{SPSA} when the

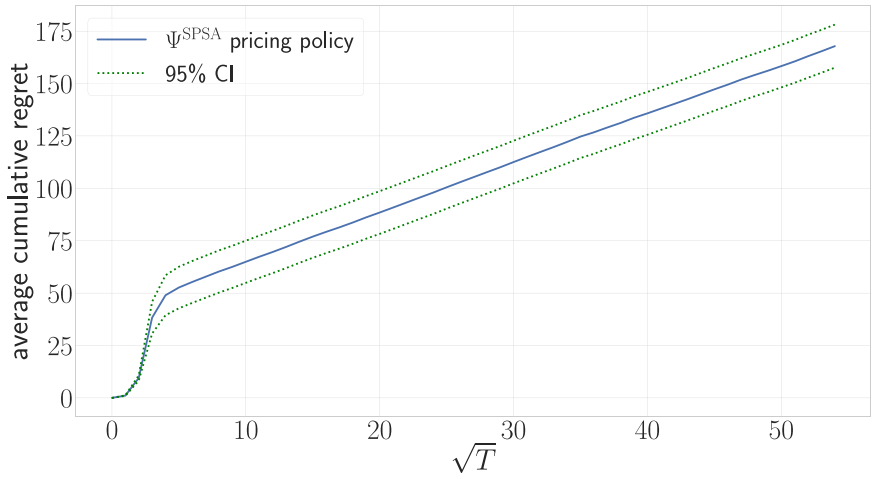

Fig. 2. Average cumulative regret of the SPSA pricing policy.

Fig. 3. Sample paths of the averaged prices over 100 simulation replications of the SPSA pricing policy.

underlying demand curve deviates from linearity. With a slight abuse of notation, let $\{\epsilon_t\}$ be i.i.d. normal random variables with zero mean and variance ς^2. We consider the following three one-dimensional demand models:

- Linear$^+$: $D(p_t) = \max\{a + bp_t, 0\} + \epsilon_t$, where $a = 1$, $b = -0.5$, and $\varsigma^2 = 0.0025$. The optimal price is $p^* = 1$. Let initial price $p_0 = 0.4$, $a_n = 3n^{-1}$, and $c_n = n^{-1/4}$ for the SPSA pricing policy.
- Exponential: $D(p_t) = \exp(a + bp_t) + \epsilon_t$, where $a = 1$, $b = -0.3$, and $\varsigma^2 = 0.0025$. The optimal price is $p^* = 3$. Let initial price $p_0 = 0.7$, $a_n = 0.55n^{-1}$, and $c_n = n^{-1/4}$ for the SPSA pricing policy.
- Logit: $D(p_t) = \exp(a + bp_t)/(1 + \exp(a + bp_t)) + \epsilon_t$, where $a = 1$, $b = -0.3$, and $\varsigma^2 = 0.0025$. The optimal price is $p^* \approx 5.224$. Let initial price $p_0 = 4.5$, $a_n = 10n^{-1}$, and $c_n = n^{-1/4}$ for the SPSA pricing policy.

The Linear$^+$ model is basically equivalent to the linear model Eq. (12), with the difference being that it ensures the expected demand to be nonnegative. However, such a requirement is not always essential from a mathematical perspective. The Exponential and Logit models deviate significantly from the linear model considered in Section 6.1. The performance of Ψ^{SPSA}, including the accumulated regret produced and its convergence behavior for each of the above demand models, is reported in Appendix G.

We have also implemented the $\mathscr{E}\Psi^{\text{SPSA}}$, the enhanced version of Ψ^{SPSA}. The projection operation in our experiments is implemented by setting the low bound on the price to 0 and the upper bound to twice the maximum value of the optimal price vector's components. In addition, when simulating F^+ and F^-, the same random seed is used at each step of the algorithm. The experiment is performed under the same parameter setting as the original algorithm Ψ^{SPSA}. We report the mean prices (with the corresponding standard errors in parentheses) obtained upon the algorithm's termination under four different demand models in Table G.1; see Appendix G. We have tested three different variance cases for both algorithms to investigate the effect of the variance on algorithm performance. The numerical results indicate that the enhanced algorithm surpasses its predecessor in both estimation accuracy and statistical efficiency in most instances, especially under high variance scenarios. For a theoretical analysis on the effect of noise variance on finite-difference-based stochastic approximation algorithms, we refer the reader to the recent work by Hu and Fu (2024).

6.3. Comparison with the KW pricing policy

In the section, we carry out simulation experiments to compare the performance of our proposed Ψ^{SPSA} and $\mathscr{E}\Psi^{\text{SPSA}}$ algorithms with that of Ψ^{KW}. First, we consider the low-dimensional example described in Section 6.1 and use the same parameter setting as presented in that section. Then, we evaluate the three algorithms on a high-dimensional problem with a dimension of 200 and a total of 50 000 periods. Model parameters in the high-dimensional problem are randomly generated,

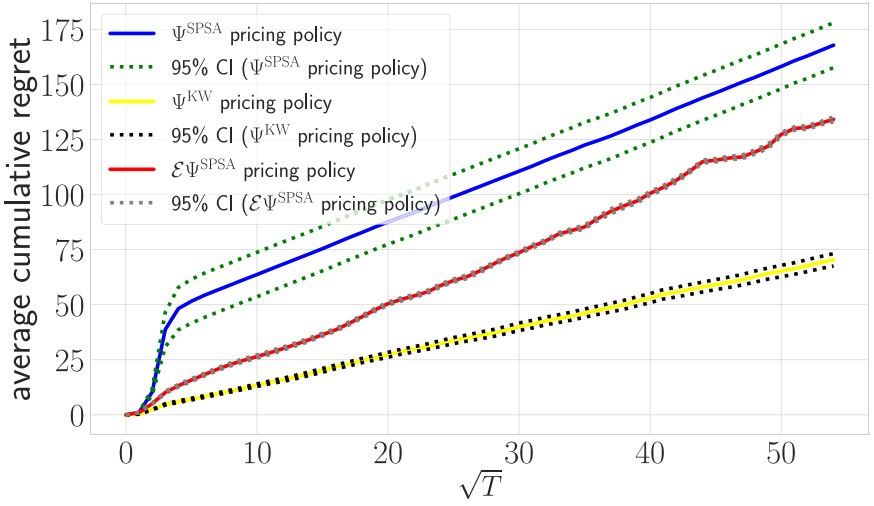

Fig. 4. Average cumulative regret of three different pricing policies on the 2-dimensional problem.

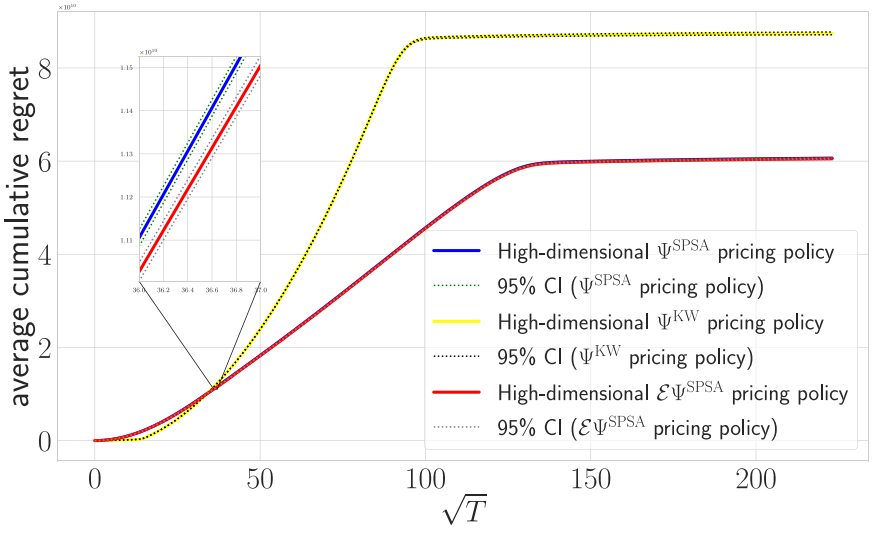

Fig. 5. Average cumulative regret of three different pricing policies on the 200-dimensional problem.

in which each component of **a** is uniformly selected from $[0, 10]$, **B** is randomly generated and satisfy the condition that diagonal elements are strictly negative with $|b_{ii}| > \sum_{j\neq i} |b_{ij}|$ for all $i \in \{1, \ldots, d\}$, and $a_n = 0.02n^{-1}$ and $c_n = 0.01n^{-1/4}$. The cumulative regrets averaged over 100 replications in the two experiments are plotted in Fig. 4 and Fig. 5, respectively.

From the figures, it can be observed that the KW pricing policy exhibits superior performance in low dimensions, but its performance becomes less competitive compared to our policies in high dimensions. Additionally, in both cases, the KW algorithm has a smaller regret value when the number of stages is relatively small. These outcomes are as expected, because the KW pricing policy alters one component of the price vector, i.e., the price of a single product, at each stage while holding all other components constant. Since all components of the price vector are fully explored deterministically, the constructed (KW-based) stochastic gradient estimator has a lower bias than that of simultaneous perturbation (Kushner & Yin, 2003). However, the KW pricing policy requires $d + 1$ periods to carry out one price updating step, leading to fewer updates within the fixed total number of periods allowed in high-dimensional problems. In contrast, the SPSA pricing policy requires only two periods to experiment with prices regardless of problem dimension, thereby ensuring a large number of price updates to the optimal price.

Remark 3. A close examination of the pricing policies employed by KW and SPSA reveals the respective complexities of both algorithms. In particular, for a given $\epsilon > 0$, it can be seen (theorem 1 and appendix in Hong et al., 2020) that the number of periods (i.e., function evaluations) required by KW to achieve an MSE that falls below ϵ is of order $O(d^3)/\epsilon^2$. In contrast, according to Theorem 1 and Appendix A, $O(d^2)/\epsilon^2$ periods is sufficient for our SPSA policy to achieve the same level of accuracy, provided that the parameters $a_n = \alpha n^{-1}$ and $c_n = \gamma n^{-1/4}$ are set with some positive constants α and γ.

Further, based on the findings illustrated in Figs. 4 and 5, which demonstrate the superior performance of the SPSA algorithm over the KW algorithm, especially under conditions involving a linear demand function and a substantial number of periods, we proceed to evaluate the efficacy of both algorithms in handling nonlinear demand scenarios, as elaborated in Section 6.2, while considering a relatively small number of periods. All parameters are set as the same as that of Hong et al. (2020). Fig. 6 presents the averaged cumulative regrets of the three policies under Linear$^+$, Exponential and Logit demand models, respectively, over 100 independent replications for $T = 1, 2, \ldots, 200$. From the figure, it is clear that the KW pricing policy outperforms our approach when the underlying model is approximately linear. Nevertheless, in scenarios characterized by nonlinear demand models,

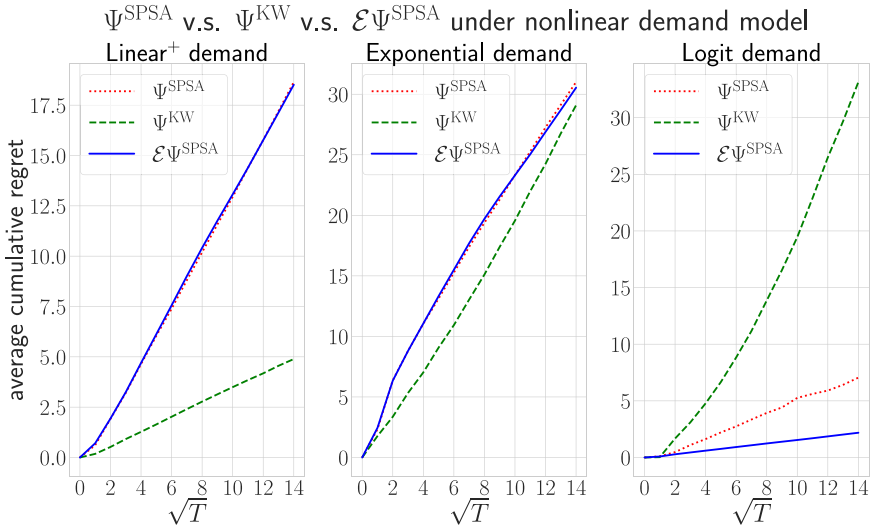

Fig. 6. Comparison policies under nonlinear demand models.

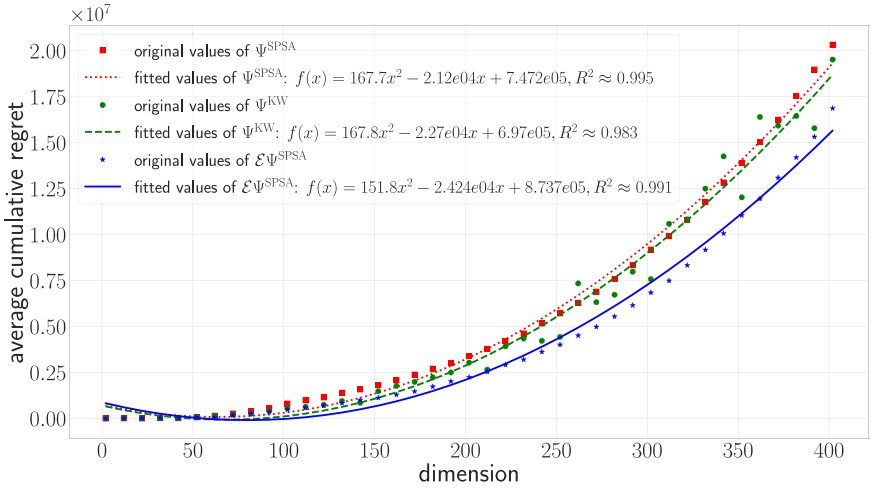

Fig. 7. Polynomial dimension dependency of three different pricing policies.

our policy's performance is either comparable or superior to that of the KW pricing policy, which indicates that our policies may offer advantages when employed in dynamic pricing contexts with nonlinear demand models.

Finally, we conduct a numerical experiment to study the impact of problem dimension on the regret by running the three algorithms on a set of test problems with dimensions varying from 2 to 402, where the total number of periods is fixed at 3000. This allows us to assess the terminal regret exhibited by the three algorithms across varying dimensions. For each dimension, model parameters are randomly generated, with each component of $\mathbf{a}$ being uniformly selected from $[0, 0.01]$. The matrix $\mathbf{B}$ is also randomly generated and satisfies the condition that diagonal elements are strictly negative with $|b_{ii}| > \sum_{j\neq i} |b_{ij}|$, $b_{ij} \in [0, 0.01]$ for all $j \neq i \in \{1, \ldots, d\}$, and $\varsigma^2 = 0.0025$. The input parameters are set as $a_n = 0.2n^{-1}$ and $c_n = 0.1n^{-1/4}$. Fig. 7 shows the cumulative regrets generated by the three algorithms, each averaged over 100 replications, as functions of problem dimensions. The figure clearly indicates that the regret values of both our algorithms and the KW algorithm demonstrate a tendency to increase polynomially with the dimension d. This observation is consistent with the theoretical findings derived from Corollary 1, as presented in Section 4.

6.4. Numerical experiment on the extensions

First, we illustrate the dynamic regret by measuring the average cumulative regret incurred under various patterns of changing demands. Following the numerical example used in Besbes et al. (2015), we construct a simple two-dimensional non-stationary environment. In particular, under any price vector $\mathbf{p} = (p_1, p_2)$, the expected revenue function at period $t \in \{1, 2, \ldots, T\}$ is expressed as $f_t(\mathbf{p}) = b_1 p_1^2 + b_2 p_2^2 + a_{1,t} p_1 + a_{2,t} p_2$. This is a quadratic function with its optimal solution equal to $(\frac{-a_{1,t}}{2b_1}, \frac{-a_{2,t}}{2b_2})$. Such a revenue function represents a linear demand function with no correlations between different products. Note that $a_{1,t}$ and $a_{2,t}$ are time-varying. For every $i \in \{1, 2\}$, we consider the following three types of variation patterns of $a_{i,t}$, named as shock, decay, and linear:

$$a_{i,t}^{shock} = \begin{cases} 1 & \text{if } t \le T/4, \\ 0 & \text{otherwise.} \end{cases} \qquad a_{i,t}^{decay} = \begin{cases} 1 & \text{if } t \le T/4, \\ \mathrm{e}^{-10(t-T/4)/T} & \text{otherwise.} \end{cases}$$

$$a_{i,t}^{linear} = \begin{cases} 1 & \text{if } t \le T/4, \\ \frac{T-t}{T-T/4} & \text{otherwise.} \end{cases}$$

It can be verified that the variation can be bounded by the budget $V_T = O(\sqrt{T})$ in all the considered patterns. In our numerical experiment,

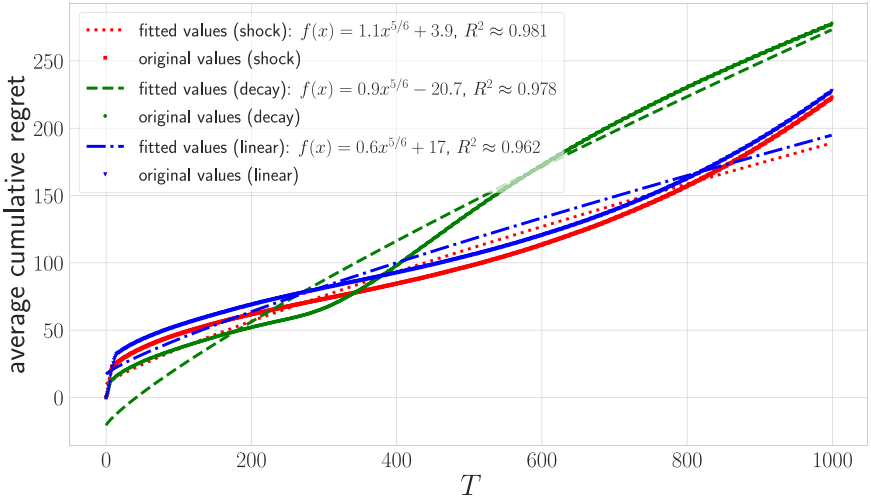

Fig. 8. Average cumulative regret of three different changing demand functions: shock, decay, and linear.

we set $b_1 = -0.5$ and $b_2 = -0.7$. Recall from Theorem 2, we set $a_t = \frac{2}{Ht}$ and $\delta_t = h_t = a_t^{1/4}$ where $H = \max\{-2b_1, -2b_2\} = 1.4$. The noise at every period is set to be independent normal random variables with zero mean and variance equal to 0.3. We assume the price vector lies in $[0, 10] \times [0, 10]$. Let $T = 1000$. For each of the considered variation patterns, we simulated the proposed policy with the SPSA-type EGS algorithm as a subroutine using a batch size of $T^{1/3}$ in the restarting procedure, replicating each instance 100 times and calculating the average regret. The results are reported as in Fig. 8. One can observe that a 5/6-degree function fits well with the regrets, which is consistent with the theoretical bounds in Proposition 2.

Second, to investigate the efficacy of our constrained SPSA-based heuristics, we consider the example used in Chen and Shi (2023, section 6) where the seller sells two products ($d = 2$) with three resources ($m = 3$). The resource consumption matrix $\mathbf{A}$ is defined by its first, second, and third rows as $(1, 1)$, $(3, 1)$, and $(0, 5)$. The purchase probabilities for each customer follow a multinomial logit model with a base utility vector of $(2, 2)$. Specifically, given any price vector $\mathbf{p} = (p_1, p_2)^{\mathrm{T}}$, the customer either selects product $j \in \{1, 2\}$ with probability $\exp(p_j)/\big(1 + \exp(p_1) + \exp(p_2)\big)$ or chooses not to purchase anything with probability $1/\big(1 + \exp(p_1) + \exp(p_2)\big)$. The feasible region is set to $\mathbf{p}_1 \in [0.5, 5]$ and $\mathbf{p}_2 \in [0.5, 5]$. Note that we examine a sequence of "increasing" problems, which is standard for evaluating the algorithm's performance in finite inventory settings (e.g., Chen & Gallego, 2022; Chen, Jasin, & Duenyas, 2019; Chen & Shi, 2023). Under such a context, both the initial resource capacity and the demand rate are scaled by a factor $k > 0$. Since we consider the case where there is exactly one customer arrival in every discrete time period, scaling the length of the selling season is equivalent to scaling the demand rate. In our experiments, the initial inventories are set to $(1.5T, 1.2T, 3.0T)$.

For the projection SPSA, we employ two regression techniques: linear regression and Gaussian process regression (Rasmussen et al., 2006). In the case of penalty SPSA, we examine two types of penalty functions—quadratic penalty function and absolute value penalty function. Here, we simply let λ_n be a zero vector (cf. Wang & Spall, 2008, Section 4). The utilization of linear regression demand in penalty SPSA is motivated by the linear nature of $\hat{\mathbf{Q}}(\cdot)$ concerning $\mathbf{p}$. Hence we can derive the gradient of the penalty function analytically. In both heuristics, we fix a_n and c_n as $3n^{-0.501}/T$ and $2n^{-0.101}$, respectively. In Step 1, we employ a set of 128 low-discrepancy Sobol points in the two-dimensional space (see, e.g., Glasserman, 2004, Chapter 5), i.e., $s' = 128$. We allocate $T/3$ periods for regression model fitting, and set $n' = T/(3s')$. In Step 2, $\boldsymbol{\Delta}_n$ is chosen as the 2-dimensional Rademacher vector. At Step 3 of $\Psi^{\mathrm{SPSA-Proj.}}$, the sub-problem (11) transforms into a quadratic programming problem when $\hat{\mathbf{Q}}(\cdot)$ is linear, which can be efficiently solved. On the other hand, if $\hat{\mathbf{Q}}(\cdot)$ is a (possibly non-convex) Gaussian process, we employ the sequential quadratic programming (SQP) approach to estimate the optimal solution to (11). It is worth noting that SQP can be relatively easily implemented using the built-in solvers in scientific programming languages, such as the SciPy library in Python (e.g., `scipy.optimize.minimize(method='SLSQP')`). Regarding Step 3 of $\Psi^{\mathrm{SPSA-Pena.}}$, following Wang and Spall (2008), we set $r_n = 10n^{0.1}$ for the quadratic penalty function and a constant penalty of $r_n = 10$ for the absolute value penalty function.

We gauge the performance of each heuristic by calculating the percentage of cumulative revenue generated compared to the "optimal revenue", averaged over 30 replications; see Fig. 9. By "optimal revenue", we refer to the upper bound on optimal revenue where the retailer knows the demand–price function prior to the selling season, representing the optimal value of (9). In light of Eq. (10), the percentage compared to the true optimal revenue is at least as high as the numbers shown in Fig. 9. The figure illustrates that the projection SPSA with linear regression shows only marginal growth in the number of periods. This phenomenon is believed to stem from the model misspecification inherent in linear demand models. Utilizing a nonlinear Gaussian process to model the demand leads to significant improvements in algorithmic performance. In contrast, the penalty SPSA exhibits comparable efficacy under the two penalty functions, surpassing the projection SPSA with linear regression in both instances. Nevertheless, it should be noted that penalty SPSA introduces a sequence of penalty parameters and necessitates additional hyperparameter tuning work.

7. Conclusion

We have introduced a novel nonparametric gradient-based online algorithm called SPSA pricing policy for dynamic pricing with demand learning. This algorithm addresses the challenges of solving multi-product problems with noisy black-box demand/revenue functions. In the context of nonparametric multi-product settings, our proposed pricing policy makes a significant methodological contribution to the literature on dynamic pricing with demand learning. The SPSA pricing policy is especially promising for high-dimensional problems, as it only requires two price experimentations per iterative update. In comparison to methods that rely on component-wise finite differences (such as the KW pricing policy), our algorithm has the potential to achieve substantial computational savings. Incorporating enhancements such as projection and CRN may further improve the numerical stability of the algorithm, reduce the variance in the gradient estimates, and therefore lead to its enhanced empirical performance in practical applications.

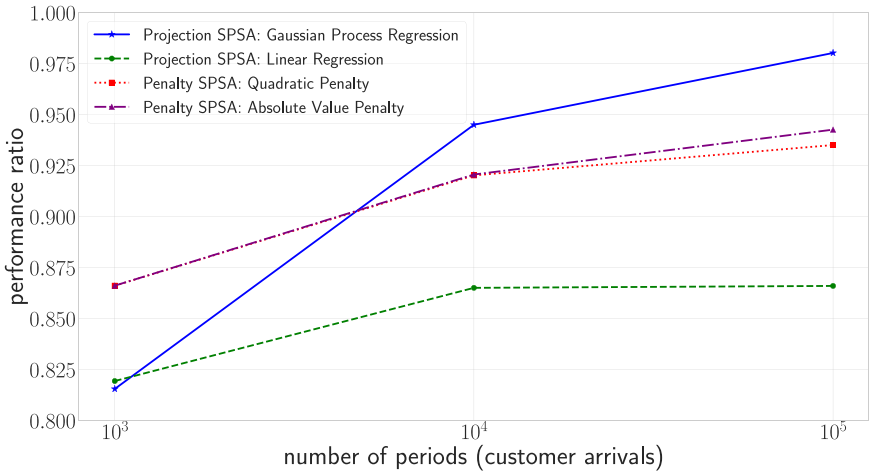

Fig. 9. Performance comparison of three constrained SPSA-based dynamic pricing heuristics.

Under the assumption of concavity and smoothness of the revenue function and other appropriate conditions, we have analyzed the MSE of the SPSA pricing policy and shown its capability to achieve the optimal order of regret $O(\sqrt{T})$. Additionally, through theoretical analysis and numerical experiments, we have further investigated the performance of the SPSA method under non-stationary and limited inventory scenarios. Simulation experiments indicate that our algorithm performs well, especially on high-dimensional problems, in terms of the total time periods required to achieve reasonable performance.

Throughout the paper, the demand function in each period is solely affected by the current advertised price (with stochastic demand shocks). However, it is important to acknowledge that in real-world situations, demand may be influenced not only by the present price but also by past pricing instances. Under such a scenario, demands across various periods are in general correlated. The corresponding analysis could be more challenging than those addressed in our current work. For instance, when considering the reference price effect (den Boer & Keskin, 2022), demand becomes contingent on the entire historical pricing trajectory. Exploring this aspect could be an interesting avenue. Another potential future extension of this study will be to consider the setting involving multiple types of consumers and investigate personalized dynamic pricing strategies tailored to different consumer segments. Lastly, it will also be interesting to explore whether the regularity conditions on the unknown demand/revenue functions could be relaxed by considering more general multi-modal revenue functions; cf., e.g., Wang, Chen, and Simchi-Levi (2021).

CRediT authorship contribution statement

Xiangyu Yang: Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Jianghua Zhang:** Supervision, Investigation, Funding acquisition. **Jian-Qiang Hu:** Writing – review & editing, Supervision, Investigation, Conceptualization. **Jiaqiao Hu:** Writing – review & editing, Supervision, Methodology, Investigation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant numbers: 72293582, 72033003, 72350710219, 72342006, 72293565), the China Postdoctoral Science Foundation (Grant number: 2023M732054), the Shandong Provincial Natural Science Foundation (Grant number: ZR2023QG159), the Shandong Postdoctoral Science Foundation (Grant number: SDCX-RS-202303004), and the National Science Foundation (Grant number: CMMI-2027527).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejor.2024.06.017.

References

Agrawal, S., & Devanur, N. (2016). Linear contextual bandits with knapsacks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems: vol. 29*, Curran Associates, Inc.

Andradóttir, S. (1995). A stochastic approximation algorithm with varying bounds. *Operations Research*, *43*(6), 1037–1048.

Auer, P., Ortner, R., & Szepesvári, C. (2007). Improved rates for the stochastic continuum-armed bandit problem. In *Learning theory: 20th annual conference on learning theory, COLT 2007, San Diego, CA, USA; June 13-15, 2007. Proceedings 20* (pp. 454–468). Springer.

Badanidiyuru, A., Kleinberg, R., & Slivkins, A. (2013). Bandits with knapsacks. In *2013 IEEE 54th annual symposium on foundations of computer science* (pp. 207–216). Los Alamitos, CA, USA: IEEE Computer Society.

Bergemann, D., & Schlag, K. (2011). Robust monopoly pricing. *Journal of Economic Theory*, *146*(6), 2527–2543.

Bertsimas, D., & Perakis, G. (2006). Dynamic pricing: A learning approach. *Mathematical and Computational Models for Congestion Charging*, 45–79.

Besbes, O., Gur, Y., & Zeevi, A. (2015). Non-stationary stochastic optimization. *Operations Research*, *63*(5), 1227–1244.

Besbes, O., & Zeevi, A. (2009). Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, *57*(6), 1407–1420.

Besbes, O., & Zeevi, A. (2011). On the minimax complexity of pricing in a changing environment. *Operations Research*, *59*(1), 66–79.

Besbes, O., & Zeevi, A. (2012). Blind network revenue management. *Operations Research*, *60*(6), 1537–1550.

Besbes, O., & Zeevi, A. (2015). On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, *61*(4), 723–739.

den Boer, A. V. (2014). Dynamic pricing with multiple products and partially specified demand distribution. *Mathematics of Operations Research*, *39*(3), 863–888.

den Boer, A. V., & Keskin, N. B. (2020). Discontinuous demand functions: Estimation and pricing. *Management Science*, *66*(10), 4516–4534.

den Boer, A. V., & Keskin, N. B. (2022). Dynamic pricing with demand learning and reference effects. *Management Science*, *68*(10), 7112–7130.

den Boer, A. V., & Zwart, B. (2014). Simultaneously learning and optimizing using controlled variance pricing. *Management Science*, *60*(3), 770–783.

Borkar, V. S. (2009). Stochastic approximation: a dynamical systems viewpoint. *vol. 48*, Springer.

Broadie, M., Cicek, D., & Zeevi, A. (2011). General bounds and finite-time improvement for the Kiefer-Wolfowitz stochastic approximation algorithm. *Operations Research*, *59*(5), 1211–1224.

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

Bubeck, S., Cesa-Bianchi, N., et al. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, *5*(1), 1–122.

Bubeck, S., Munos, R., Stoltz, G., & Szepesvári, C. (2011). X-Armed bandits. *Journal of Machine Learning Research*, *12*(5).

Carvalho, A. X., & Puterman, M. L. (2005). Learning and pricing in an internet environment with binomial demands. *Journal of Revenue and Pricing Management*, *3*, 320–336.

Chen, M., & Chen, Z.-L. (2015). Recent developments in dynamic pricing research: multiple products, competition, and limited demand information. *Production and Operations Management*, *24*(5), 704–731.

Chen, N., & Gallego, G. (2022). A primal–dual learning algorithm for personalized dynamic pricing with an inventory constraint. *Mathematics of Operations Research*, *47*(4), 2585–2613.

Chen, Q., Jasin, S., & Duenyas, I. (2019). Nonparametric self-adjusting control for joint learning and optimization of multiproduct pricing with finite resource capacity. *Mathematics of Operations Research*, *44*(2), 601–631.

Chen, P.-Y., & Liu, S. (2019). Recent progress in zeroth order optimization and its applications to adversarial robustness in data mining and machine learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 3233–3234).

Chen, Y., & Shi, C. (2023). Network revenue management with online inverse batch gradient descent method. *Production and Operations Management*.

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

Cope, E. W. (2009). Regret and convergence bounds for a class of continuum-armed bandit problems. *IEEE Transactions on Automatic Control*, *54*(6), 1243–1253.

Den Boer, A. V. (2015a). Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in Operations Research and Management Science*, *20*(1), 1–18.

Den Boer, A. V. (2015b). Tracking the market: Dynamic pricing and learning in a changing environment. *European Journal of Operational Research*, *247*(3), 914–927.

Ferreira, K. J., Simchi-Levi, D., & Wang, H. (2018). Online network revenue management using thompson sampling. *Operations Research*, *66*(6), 1586–1602.

Gallego, G., & Van Ryzin, G. (1994). Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, *40*(8), 999–1020.

Gallego, G., & Van Ryzin, G. (1997). A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, *45*(1), 24–41.

Ghadimi, S., & Lan, G. (2013). Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, *23*(4), 2341–2368.

Glasserman, P. (2004). *Monte Carlo methods in financial engineering: vol. 53*, Springer.

Grant, J. A., & Szechtman, R. (2021). Filtered poisson process bandit on a continuum. *European Journal of Operational Research*, *295*(2), 575–586.

Harrison, J. M., Keskin, N. B., & Zeevi, A. (2012). Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Science*, *58*(3), 570–586.

Hazan, E., et al. (2016). Introduction to online convex optimization. *Foundations and Trends® in Optimization*, *2*(3–4), 157–325.

Hong, L. J., Li, C., & Luo, J. (2020). Finite-time regret analysis of Kiefer-Wolfowitz stochastic approximation algorithm and nonparametric multi-product dynamic pricing with unknown demand. *Naval Research Logistics*, *67*(5), 368–379.

Hu, J., & Fu, M. C. (2024). On the convergence rate of stochastic approximation for gradient-based stochastic optimization. *Operations Research*, http://dx.doi.org/10.1287/opre.2023.0055, ePub ahead of print March 8.

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., & Zeevi, A. (2017). Chasing demand: Learning and earning in a changing environment. *Mathematics of Operations Research*, *42*(2), 277–307.

Keskin, N. B., & Zeevi, A. (2018). On incomplete learning and certainty-equivalence control. *Operations Research*, *66*(4), 1136–1167.

Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 462–466.

Kleinberg, R. (2004). Nearly tight bounds for the continuum-armed bandit problem. *Advances in Neural Information Processing Systems*, *17*.

Kleinberg, R., Slivkins, A., & Upfal, E. (2008). Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on theory of computing* (pp. 681–690).

Kushner, H. J., & Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications* (2nd ed.). Springer New York, NY.

Lattimore, T., & Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Law, A. M. (2015). *Simulation modeling & analysis* (5th ed.). New York, NY, USA: McGraw-Hill.

Lobo, M. S., & Boyd, S. (2003). Pricing and learning with uncertain demand. In *INFORMs revenue management conference* (pp. 63–64). Citeseer.

Meyn, S. (2022). *Control systems and reinforcement learning*. Cambridge University Press.

Misra, K., Schwartz, E. M., & Abernethy, J. (2019). Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, *38*(2), 226–252.

Nemirovski, A., Juditsky, A., Lan, G., & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, *19*(4), 1574–1609.

Rasmussen, C. E., Williams, C. K., et al. (2006). *Gaussian processes for machine learning: vol. 1*, Springer.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 400–407.

Shi, J., & Spall, J. C. (2021). SQP-based projection SPSA algorithm for stochastic optimization with inequality constraints. In *2021 American control conference* (pp. 1244–1249). IEEE.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, *37*(3), 332–341.

Spall, J. C. (1997). A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, *33*(1), 109–112.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, *25*(3–4), 285–294.

Wang, Y., Chen, B., & Simchi-Levi, D. (2021). Multimodal dynamic pricing. *Management Science*, *67*(10), 6136–6152.

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.

Wang, I.-J., & Spall, J. C. (2008). Stochastic optimisation with inequality constraints using simultaneous perturbations and penalty functions. *International Journal of Control*, *81*(8), 1232–1238.

Yang, C., & Xiong, Y. (2020). Nonparametric advertising budget allocation with inventory constraint. *European Journal of Operational Research*, *285*(2), 631–641.