Expert Systems With Applications 244 (2024) 122948

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Data-driven dynamic pricing and inventory management of an omni-channel retailer in an uncertain demand environment

Shiyu Liu [a], Jun Wang [a,*], Rui Wang [b], Yue Zhang [c,*], Yanjie Song [d], Lining Xing [e]

[a] School of Economics and Management, Beihang University, Beijing 100191, China
[b] College of Systems Engineering, National University of Defense Technology, Changsha, Hunan 410073, China
[c] School of Reliability and System Engineering, Beihang University, Beijing 100191, China
[d] National Defense University, Haidian District, Beijing 100091, China
[e] School of Electronic Engineering, Xidian University, Xi'an 710126, China

ARTICLE INFO

Keywords:
Dynamic pricing
Inventory management
Omni-channel retailing
Deep reinforcement learning

ABSTRACT

In recent years, omni-channel retailing has become immensely popular among both retailers and consumers. In this approach, retailers often leverage their brick-and-mortar stores to fulfill online orders, leading to the need for simultaneous decision-making on replenishment and inventory rationing. This inventory strategy presents significant complexities in traditional dynamic pricing and inventory management problems, particularly in unpredictable market environments. Therefore, we have developed a dynamic pricing, replenishment, and rationing model for omni-channel retailers using a two-level partially observed Markov decision process to visualize the dynamic process. We propose to use a deep reinforcement learning algorithm, called Maskable LSTM-Proximal Policy Optimization (ML-PPO), which integrates the current observations and future predictions as input to the agent and uses the invalid action mask to guarantee the allowable actions. Our simulation experiments have demonstrated the ML-PPO's efficiency in maximizing retailer profit and service level, along with its generalized ability to tackle dynamic pricing and inventory management problems.

1. Introduction

Since the beginning of the COVID-19 epidemic, the consumption scenario of retail has accelerated to online channels. Consumers who find it difficult to go out have chosen to consume through online scenarios such as delivery-to-home business and live shopping. To meet consumer demand, the traditional retail industry continues to transform into new retail, from traditional single-channel retailing to multi-channel retailing, until omni-channel retailing. Based on a report by Statista on omni-channel retailing in the United States, it was found that as of 2022, more than 80 % of retailers have transitioned to omni-channel retailing. The adoption of omni-channel retailing presents both opportunities and challenges for retailers. On the one hand, it allows them to optimize performance by effectively coordinating operations across different channels (Cao & Li, 2015). On the other hand, it presents significant challenges for retailers in decision-making.

Faced with the omni-channel retailing pattern, retailers must reconsider their inventory strategies (Jalilipour Alishah, Moinzadeh, & Zhou, 2015). To ensure efficient and timely distribution, stores often resort to store-warehouse integration methods, such as the ship-from-store strategy (Mou, Robb, & DeHoratius, 2018). However, this inventory strategy can result in conflicts when online and offline consumers attempt to purchase the same item concurrently. To solve this problem, retailers choose to store a certain amount of inventory to fulfill the online demand. For instance, a supermarket chain in China fulfills online orders using the inventory in its physical store, employing a proportional distribution scheme for online and offline inventory to reserve a portion of online inventory. As a result, the physical store is not authorized to sell locked online products. To fulfill the consumer demand in both online and offline sales channels, retailers must make timely inventory rationing decisions. Additionally, dynamic pricing plays a crucial role in enabling retailers to balance demand and inventory. Appropriate coordination between pricing and inventory decisions can reduce the risk of inventory and demand mismatch (Feng, Luo, & Shanthikumar, 2020) and enable companies to maximize profits (Lei, Jasin, & Sinha, 2018).

* Corresponding authors.
E-mail addresses: liushiyu@buaa.edu.cn (S. Liu), king.wang@buaa.edu.cn (J. Wang), ruiwangnudt@gmail.com (R. Wang), zhangyue1127@buaa.edu.cn (Y. Zhang), songyj_2017@163.com (Y. Song), lnxing@xidian.edu.cn (L. Xing).

https://doi.org/10.1016/j.eswa.2023.122948
Received 12 March 2023; Received in revised form 29 November 2023; Accepted 12 December 2023
Available online 20 December 2023

With the continuous development of omni-channel retailing, there has been an increase in research focused on pricing and inventory management, such as Goedhart, Haijema, & Akkerman (2022a) and Qiu, Ma, & Sun (2023). However, to the best of our knowledge, there is still a research gap in addressing dynamic decision-making encompassing joint pricing, ordering, and inventory rationing under uncertain demand in the context of omnichannel retailing. Hence, motivated by the existing gap in the literature and the practical necessity, this study aims to investigate the **d**ynamic **p**ricing, **r**eplenishment, and **r**ationing (DPRR) problem of an omnichannel retailer, particularly with imperfect information concerning future demand and consumer distribution across channels.

In terms of methodology, current studies in dynamic pricing and inventory control primarily rely on the Markov decision problem (MDP) to obtain numerical solutions (e.g., Goedhart, Haijema, & Akkerman, 2022b; Zhou, Yang, & Fu, 2022). Since solving MDP is time-consuming because of the curse of dimensionality, several algorithms are proposed to obtain a faster solution. However, in the real world, retailers cannot get an accurate demand state due to the uncertainty in the demand environment (Aviv & Pazgal, 2005). Thus, we formulate a finite horizon partially observed Markov decision problem (POMDP) on the DPRR problem of an omni-channel retailer in an uncertain demand market. Considering that the unobservable state further increases the difficulty of solving, we propose to use a deep reinforcement learning algorithm to solve the DPRR problem.

The main contributions of this paper are summarized as follows:

i. We constructed a model of the dynamic pricing, replenishment, and rationing problem (DPRR) for an omni-channel retailer to maximize the retailer's profit in an uncertain demand market.
ii. We proposed to use a Maskable Long-Short-Term-Memory (LSTM)-Proximal Policy Optimization (ML-PPO) algorithm that concatenates the observation state and the predictions of the future state from LSTM as input into the PPO agent and uses the invalid action mask to constraint the action space.
iii. We conducted experiments to evaluate the performance of the ML-PPO algorithm and demonstrated that it is highly effective and robust in various environment settings, leading to improved total profits and service levels for omni-channel retailers.

The rest of this paper is structured as follows. Section 2 reviews the related work. Section 3 sets up a model of the DPRR problem. Section 4 proposes an improved deep reinforcement learning algorithm. Section 5 introduces the simulation experiments to evaluate the performance of the designed algorithm. Section 6 concludes the paper and future research orientations.

2. Related work

In this section, we reviewed the prior literature about the joint pricing and inventory management problem and the application of deep reinforcement learning in supply chain management.

2.1. Joint pricing and inventory management

Joint pricing and inventory management are essential issues in supply chain management (Elmaghraby & Keskinocak, 2003; Simchi-Levi & Agrawal, 2004). The early studies mainly concentrated on single-channel supply chains with deterministic demands (Chen & Hu, 2012). As the research progressed, the problem's scenario gradually broadened. Scholars have proposed dependent demand functions to portray demand, such as the price-dependent demand function (Fang, Nguyen, & Currie, 2021; Sepehri, Mishra, Tseng, & Sarkar, 2021) and inventory-dependent demand function (Bardhan, Pal, & Giri, 2019; Cárdenas-Barrón, Shaikh, Tiwari, & Treviño-Garza, 2020). A part of scholars assume that the parameters are known and constant, a situation usually referred to as full-information demand (He, Huang, & Li, 2020; Wang, Gan, Li, & Yan, 2021), while other parts of scholars, in conjunction with actual sales scenarios, argue that demand information is not completely known and demand learning is required through parameter estimation and data-driven approaches. Chen, Chao, & Wang (2020) designed a data-driven algorithm and performed parameter estimation using great likelihood estimation. Keskin, Li, & Song (2022) argued that retailers do not have all the information about demand and used a data-driven approach to parameter estimation of the demand function. Neghab, Khayyati, & Karaesmen (2022) considered a single-period inventory problem with random demand with both directly observable and unobservable features.

Furthermore, joint pricing and inventory management in multi-channel supply chains have attracted more attention with the development of supply chains. Batarfi, Jaber, & Glock (2019) investigated a dual-channel supply chain's pricing and inventory decisions. He et al. (2020) studied a single-retailer-single-vendor dual-channel supply chain model and the pricing and inventory decisions simultaneously. Gupta, Ting, & Tiwari (2019) considered a retailer with many offline stores but then added an online channel and developed a decision support model to optimize pricing and inventory control. Liu & Xu (2020) studied joint decisions on pricing and ordering for omnichannel BOPS retailers. Qiu et al. (2023) considered a joint pricing and ordering optimization problem of an omnichannel retailer and proposed a data-driven robust optimization approach to handle the demand uncertainty.

Recently, the study of the problem transformed from a static problem to a dynamic problem with multi-periods. Feng et al., (2020) integrated dynamic pricing with inventory decisions where loss is allowed. Li & Mizuno (2022) used different power structures to study the joint dynamic pricing and inventory problem in the dual channel. Keskin et al. (2022) established a model describing dynamic pricing and ordering for perishable items.

In inventory management, inventory rationing is an essential topic. Topkis (1968) first proposed inventory rationing work as a static strategy. Since then, research has shifted from static rationing strategies to dynamic ones, first studied by Teunter & Klein Haneveld (2008). Turgay, Karaesmen, & Örmeci (2015) investigated a dynamic inventory rationing problem with random replenishment opportunities. In early studies, rationing is often used to link different types of demand to different ways of satisfying them. Recently, with the development of multi-channel supply chains, these different demand types are similar to different channels in the supply chain. Goedhart et al. (2022a) considered a store whose inventory fulfills in-store demand and online orders. They modeled it by a two-level Markov Decision Problem to maximize the expected profit.

2.2. Application of deep reinforcement learning (DRL)

Reinforcement learning is an unsupervised learning method that learns and updates itself by interacting with the environment, i.e., a machine learning algorithm that continuously learns by trial and error and modifies its behavior through the evaluative information it obtains. Deep reinforcement learning is an algorithm that combines deep learning and reinforcement learning methods and enables direct output of actions by introducing neural network structures.

Proximal Policy Optimization (PPO) is an on-policy deep reinforcement learning algorithm (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017). On-policy means that it explores by sampling actions according to the latest version of its stochastic strategy. The amount of randomness in action selection depends on the initial conditions and the training procedure. During the training process, the randomness of the strategy usually decreases gradually as the update rules encourage it to exploit the rewards it has found. PPO is based on Trust Region Policy Optimization (TRPO). Compared with TRPO, PPO is a family of first-order methods and is easier to implement.

With the widespread use of deep reinforcement learning, the PPO

Table 1
Position of our work in the existing literature.

Research Paper	Dynamic Pricing	Inventory replenishment	Inventory rationing	Channel	Partially observed state	DRL
Our work	√	√	√	Omni	√	√
Wu et al., 2023	√			Single		√
De Moor et al., 2022		√		Single		√
Goedhart et al., 2022a		√	√	Omni		
Keskin et al., 2022	√	√		Single	√	
Neghab et al., 2022		√		Single	√	√
Li & Mizuno, 2022	√	√		Dual		
Oroojlooyjadid et al., 2022		√		Single		√
Wang et al., 2022		√		Single		√
Zhou et al., 2022	√	√		Single		√
Wang et al., 2021	√	√		Single		√
He et al., 2020	√	√		Dual		
Chen et al., 2020	√	√		Single		
Feng et al., 2020	√	√		Single		

algorithm is widely used to solve dynamic pricing and inventory management problems. Ding et al. (2022) proposed an efficient algorithm based on PPO called CD-PPO to solve the inventory management problem. Yang, Feng, & Whinston (2022) adopted the PPO algorithm to make pricing and information disclosure decisions over time. Goedhart et al. (2022b) used the PPO algorithm to optimize the rationing and ordering decisions of an omni-channel retailer. Wu, Bi, & Liu (2023) used the PPO algorithm to derive optimal dynamic pricing strategies with online reviews.

Furthermore, there are various other DRL algorithms used in dynamic pricing and inventory management problems. Wang et al. (2021) designed a DRL algorithm based on the deep Q-network (DQN) to obtain an approximate optimal strategy for pricing and ordering decisions for perishable products. Oroojlooyjadid, Nazari, Snyder, & Takáč (2022) proposed a DQN algorithm to play the beer game, where agents select order quantities to minimize total costs. De Moor, Gijsbrechts, & Boute (2022) constructed an improved DQN algorithm with reward shaping to optimize the inventory management problem for perishable goods. Wang et al. (2022) proposed a hybrid simulation and reinforcement learning method to find a superior dynamic inventory replenishment strategy. Zhou et al. (2022) proposed an improved DQN method to solve the single-channel dynamic pricing and inventory management problem with reference price effect.

Table 1 summarizes how our work compares to the existing literature and its position in the existing literature. To fill the knowledge gap, we focus on the joint dynamic pricing and inventory management consisting of order quantity and inventory rationing decisions in an omni-channel supply chain and consider the unobservable state due to the uncertain demand market. Meanwhile, we use a deep reinforcement learning algorithm based on the PPO algorithm to improve the accuracy by adding the LSTM algorithm and invalid action mask.

3. Problem formulation

In this section, we set up a DPRR model for an omni-channel retailer in an uncertain demand market. We consider an omni-channel retailer with both online and offline channels using an integrated warehouse-storage inventory management approach and a uniform pricing

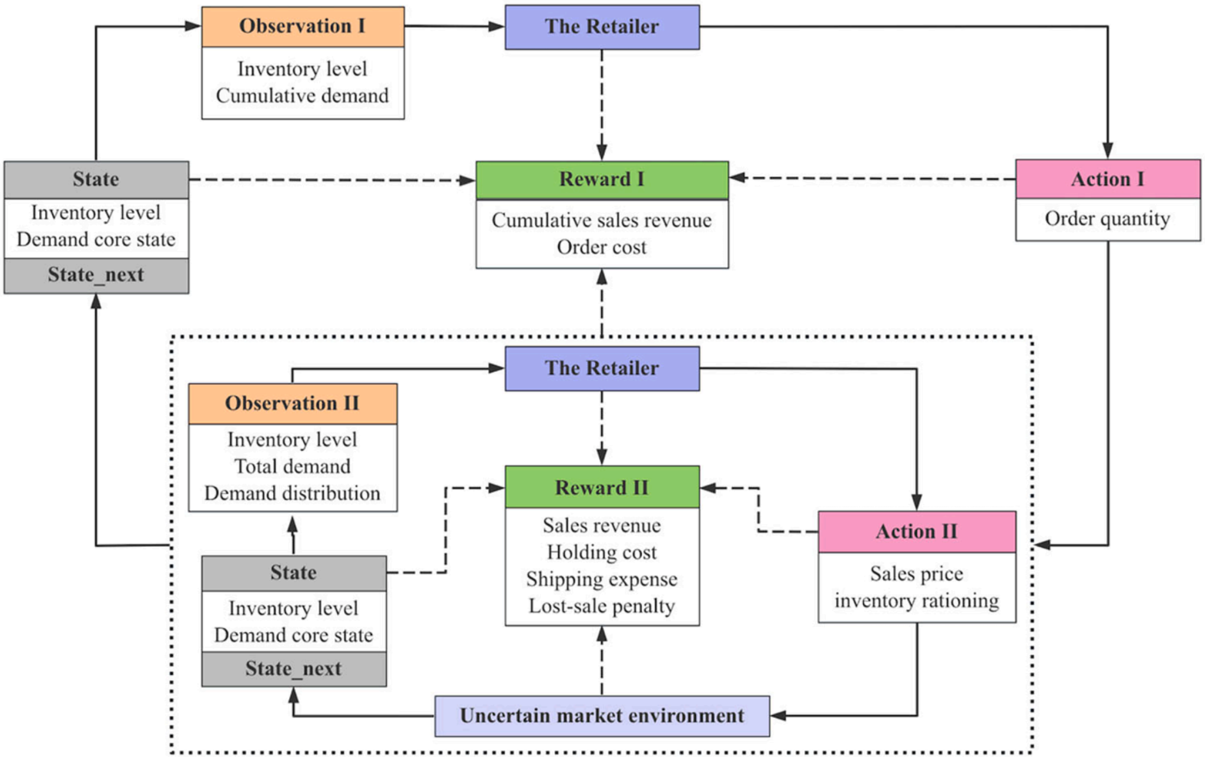


Fig. 1. The interaction of an omni-channel retailer and the environment.

Table 2
Description of symbols.

Parameters	
Γ	Order Cycle
c^h	Holding cost per item per period
c^l	Lost-sales penalty per item
c^s	Shipping expense per item purchased online
c^o	Order cost per item
I_t	Inventory level at the beginning of period t
d_t	Total demand of period t
λ	Demand distribution rate in online channels of period t
Φ	Demand core state
$\mathscr{R}_t^I$	Reward function of Level I for each order cycle
$\mathscr{R}_t^{II}$	Reward function of Level II at time t ($\mathscr{R}_t^{II} = \mathscr{R}_t^{on} + \mathscr{R}_t^{off}$)
$\mathscr{R}_t^{on}$	Online-channel profit of period t
$\mathscr{R}_t^{off}$	Offline-channel profit of period t
γ	Discount factor
Decision variables	
$Q_{n\Gamma}$	Order-up-to level at an order point
p_t	Non-negative discrete variables indicating uniform price at time t
y_t^{on}	Non-negative integer variables indicating online inventory rationing of period t
y_t^{off}	Non-negative integer variables indicating online offline-channel inventory rationing of period t ($y_t^{off} = I_t - y_t^{on}$)

strategy for online and offline channels, which is widely used in omni-channel retailing (Wu, Zhao, Yan, & Wang, 2020). The online and offline inventory is proportionally rationed and the fulfillment rate of online orders is ensured by online inventory blocking. We assume that the order cycle is positive and the lead time is considered to be zero. For out-of-stock items, we consider their full loss.

A typical setup for an omni-channel retailer is to replenish once in each order cycle, and the inventory rationing can be dynamically adjusted daily (Goedhart et al., 2022a). Thus, the process of the DPRR problem can be divided into two levels. We assume that $t = 0$ is the first order point, and $t = n^*\Gamma\,(n = 0, 1, 2, \cdots)$ represents a series of order points, where Γ represents the retailer's order cycle and is a positive constant. I_0 represents the retailer's initial existing inventory at the beginning of the planning horizon. In Level I, the replenishment decision is made at each order point. In Level II, the inventory rationing decisions y_t^{on} and the uniform pricing decision p_t are made at the beginning of period t. The Level II decisions are made once per period in the order cycle.

Fig. 1 illustrates the interaction between the retailer and the environment in the two-level DPRR model. At Level I, the retailer decides the order quantity at the beginning of each order cycle, which serves as the initial state of the inventory level in the Level II environment. At Level II, the retailer interacts with the environment by making Action II at the beginning of each period and receiving rewards and observations at the end of each period within one order cycle. The cumulative demand and reward obtained at Level II are then fed back to the Level I environment to inform the decision-making process for Action I. This iterative process continues throughout the decision-making cycle, enabling the retailer to optimize its dynamic pricing and inventory management strategies. The symbols of the parameters involved in the model and the decision variables are shown in Table 2.

We formulate the total demand function as price-dependent and the demand of each channel relies on the consumer distribution rate $\lambda \in [0, 1]$ in the online channel. Thus, the total demand and the demand in each channel under price p_t in period t are shown in Equations (1) - (3), where α represents the market size, β represents the price sensitivity and ε_t is the error term with $E(\varepsilon_t) = 0$.

$$d_t(p_t) = \alpha - \beta p_t + \varepsilon_t \tag{1}$$

$$d_t^{on}(p_t) = \lambda d_t \tag{2}$$

$$d_t^{off}(p_t) = (1 - \lambda)d_t \tag{3}$$

To introduce the uncertainty of the market into the demand function, we set up the demand core state Φ indicates the various demand statistical features of the market, following Aviv & Pazgal (2005). When the state at period t is $C_t = k$, we say the demand pattern of period t is k. In our model, each state $k \in \Phi$ of the demand core state is characterized by a pair of price-dependent demand parameters (α, β) and an online demand distribution rate λ.

The objective of the DPRR model is to maximize the retailer's expected profit over the finite horizon at each level. In Level II, the retailer's expected profit in period t is the sum of the profits in the online and offline channels, which are determined by the price decision p_t and the online inventory rationing decision y_t^{on} as follows:

$$\mathscr{R}_t^{\rm II}\left(y_t^{on}, p_t, C_t, I_t\right) = \mathscr{R}_t^{on}\left(y_t^{on}, p_t, C_t, I_t\right) + \mathscr{R}_t^{off}\left(y_t^{on}, p_t, C_t, I_t\right) \tag{4}$$

$$\mathscr{R}_t^{on}\left(y_t^{on}, p_t, C_t, I_t\right) = \begin{cases} (p_t - c^s)\min\left(\lambda d_t, y_t^{on}\right) \\ -c^h\left(y_t^{on} - \min\left(\lambda d_t, y_t^{on}\right)\right) \\ -c^l\left(\min\left(\lambda d_t, y_t^{on}\right) - \lambda d_t\right) \end{cases} \tag{5}$$

$$\mathscr{R}_t^{off}\left(y_t^{on}, p_t, C_t, I_t\right) = \begin{cases} p_t\min((1-\lambda)d_t, (I_t - y_{on})) \\ -c^h((I_t - y_{on}) - \min((1-\lambda)d_t, (I_t - y_{on}))) \\ -c^l(\min((1-\lambda)d_t, (I_t - y_{on})) - (1-\lambda)d_t) \end{cases} \tag{6}$$

where c^h, c^l and c^s are constants that represents the holding cost per item per period, lost-sales penalty per item, and shipping expense per item sold online, respectively, and are assumed not to vary over time. For the single-period problem in Level II, we present the concavity properties of the optimal pricing and inventory rationing policy in Theorem 1. As shown, the expected profit in any period is jointly concave with respect to (y_t^{on}, p_t), which confirms the existence of at least one extremum point and verifies the feasibility of our optimization problem.

Theorem 1. *For any period t, the expected profit* $\mathscr{R}_t^{II}(y_t^{on}, p_t, C_t, I_t)$ *under given core state* C_t *and inventory level* I_t*, is jointly concave with respect to* (y_t^{on}, p_t)*.*

Proof. We first prove the expected profit $\mathscr{R}_t^{on}\left(y_t^{on}, p_t\right)$ is jointly concave with respect to (y_t^{on}, p_t). We can conduct a categorical analysis:

i If $\lambda d_t \geq y_t^{on}$:

$$\mathbb{E}\mathscr{R}_t^{on}\left(y_t^{on}, p_t\right) = (p_t - c^s)y_t^{on} - c^l\left(\lambda(\alpha - \beta p_t) - y_t^{on}\right)$$

In this case, $\mathbb{E}\mathscr{R}_t^{on}\left(y_t^{on}, p_t\right)$ is a bilinear function of (y_t^{on}, p_t).

ii If $\lambda d_t < y_t^{on}$:

$$\mathbb{E}\mathscr{R}_t^{on}\left(y_t^{on}, p_t\right) = (p_t - c^s)\lambda(\alpha - \beta p_t) - c^h\left(y_t^{on} - \lambda(\alpha - \beta p_t)\right)$$

In this case, the first term is a quadratic function of p_t, and the coefficient of the quadratic term is negative, making it a concave function with respect to (y_t^{on}, p_t) and the second term is a linear function with respect to (y_t^{on}, p_t).

Thus, for both cases, the expected online profit $\mathscr{R}_t^{on}\left(y_t^{on}, p_t\right)$ is jointly concave with respect to (y_t^{on}, p_t). Using the same approach, we can demonstrate that the expected offline profit $\mathscr{R}_t^{off}\left(y_t^{on}, p_t, C_t, I_t\right)$ is also jointly concave with respect to (y_t^{on}, p_t). Therefore, based on the Equation (4), we can deduce that the single-period expected profit is jointly concave with respect to (y_t^{on}, p_t).

The retailer's objective in Level II is to jointly optimize inventory rationing and pricing policy to maximize the total expected discounted profits over the order cycle. The Level II profit-to-go value function $V_t^{II}(I_t, C_t)$, which represents the expected discounted total profit from period t to the end of the current order cycle that can be achieved from the current state onwards, satisfies the following dynamic programming recursion:

$$V_t^{II}(I_t, C_t) = \max_{p_t, y_t^{on}} \mathbb{E}\left(\mathscr{R}_t^{\rm II}\left(y_t^{on}, p_t, C_t, I_t\right) + \gamma \mathbb{E} V_{t+1}^{II}(I_{t+1}, C_{t+1})\right) \tag{7}$$

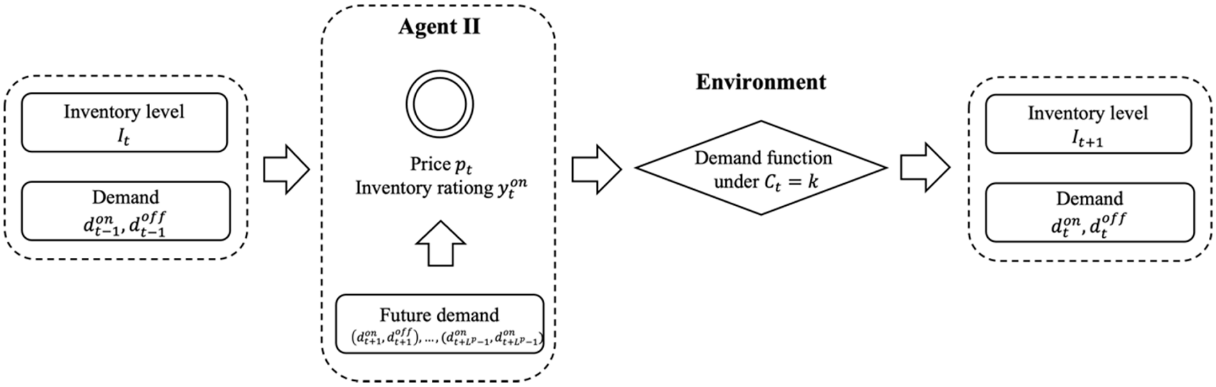


Fig. 2. Schematic diagram of agent II and environment in ML-PPO.

where $\gamma \in [0, 1]$ is the discount factor and I_{t+1} is the next state of inventory level under action (p_t, y_t^{on}) as $I_{t+1} = \left(y_t^{on} - \lambda d_t\right)^+ + \left(I_t - y_{on} - (1-\lambda) d_t\right)^+$.

In the Level I problem, the retailer's profit is determined by the cumulative profit earned from Level II decisions throughout an order cycle, along with the order cost per item, denoted as c^o, for the order quantity decided at the beginning of each order cycle. The expected profit in Level I of an order cycle can be calculated using Equation (8). The objective of the Level I problem is to optimize the ordering quantity to maximize the total expected discounted profits over the entire decision-making horizon. The Level I profit-to-go value function $V_{n\Gamma}^I(I_{n\Gamma})$ represents the expected total profit from cycle n to the end of decision-making horizon that can be achieved under the current state $I_{n\Gamma}$, as shown in Equation (9):

$$\mathscr{R}_{n\Gamma}^{\mathrm{I}}(I_{n\Gamma}, Q_{n\Gamma}) = \sum_{t=n\Gamma}^{(n+1)\Gamma-1} \mathscr{R}_t^{\mathrm{II}} - c^o * (Q_{n\Gamma} - I_{n\Gamma}) \tag{8}$$

$$V_{n\Gamma}^I(I_{n\Gamma}) = \max_{Q_{n\Gamma}} \mathbb{E}\left(\mathscr{R}_{n\Gamma}^{\mathrm{I}}(I_{n\Gamma}, Q_{n\Gamma}) + \gamma^\Gamma \mathbb{E} V_{(n+1)\Gamma}^I\left(I_{(n+1)\Gamma}\right)\right) \tag{9}$$

where $Q_{n\Gamma}$ satisfy $I_{n\Gamma} \le Q_{n\Gamma} \le Q_{max}$. Q_{max} is the maximize order constraints, such as the transport capacity.

4. Methodology

There are several methods proposed to deal with dynamic programming. However, as the state and action space dimensions increase, it becomes difficult to solve in a reasonable computational time. Thus, DRL is introduced to solve it by a near-optimal approach. The PPO algorithm has been shown to be efficient in dealing with dynamic pricing and inventory management problems. For our multi-dimensional discrete action space problem, we improve the basic PPO algorithm by introducing the LSTM algorithm as the prediction algorithm and an invalid action mask as an enhancement to solve the state-dependent action space, which is called ML-PPO.

4.1. POMDP model

According to the DPRR model, the whole process is divided into two levels of decision-making. The first level is the replenishment problem at the point of order, and the second is the daily pricing and inventory rationing problem. Since the demand core state and consumer distribution in each state are not observable, the problem is defined as a Partially Observable Markov Decision Process. The details of the POMDP model are given as follows.

4.1.1. State and observation spaces

$\mathscr{S}$ is a set of all possible environment states consisting of the current inventory level I_t and demand core state C_t, represented as $s = \{I, C, t\} \in \mathscr{S}$.

$\mathscr{O}$ is the observation space, which differs from the state space in the POMDP. For omni-channel retailers, the demand core state, including the relationship between price and demand and the distribution rate across online and offline channels, are vital points for pricing and inventory control. However, the retailer cannot observe the demand core state directly. Instead, at the beginning of period t, the retailer can observe the previous demand in each channel. Since the retailer has different concerns at two levels, each level has a different observation state. For Level I, the retailer pays more attention to the whole state of an order cycle, so the observation state of Level I consists of the inventory level at the end of an order cycle and the cumulative demand during the order cycle as $D_{n\Gamma} = \sum_{t=n\Gamma}^{(n+1)\Gamma-1} d_t$. Thus, the observation state of Level I is $o^{\mathrm{I}} = \{I, D, n\}$. For Level II, the observation state consists of the current

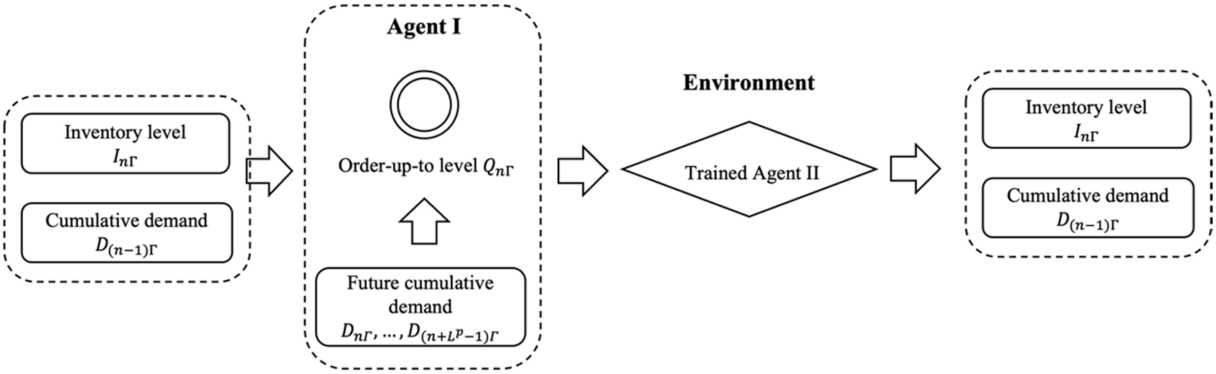


Fig. 3. Schematic diagram of agent I and environment in ML-PPO.

inventory level I_t and the demand $(d_{t-1}^{on}, d_{t-1}^{off})$ of the previous period in each channel, as $o^{\text{II}} = \{I, d^{on}, d^{off}, t\}$.

4.1.2. Action spaces

For Level I, the action is the order-up-to level $Q_{n\Gamma}$ at each order point. The order-up-to level cannot be less than the current inventory level and cannot be more than the maximum storage capacity. Thus, the Level I action space is $Q_{n\Gamma} \in [I_{n\Gamma}, Q_{max}]$

For Level II, the actions include pricing and inventory rationing as $a_t = \{p_t, y_t^{on}\} \in \mathscr{A}_t$. The first action p_t represents the uniform price of both channels and the second action y_t^{on} represents the inventory rationing to the online channel. From a real-world perspective, they are both discrete variables. Since there are costs and market normative prices for items, there is a range constraint on product pricing, i.e., $p_t \in \mathscr{P} = [p_{min}, p_{max}]$. The inventory rationing decision is based on the current inventory level at the beginning of each period and there is a need to satisfy $y_t^{on} \in \mathscr{Y}_t = (0, 1, \cdots, I_t)$ and y_t^{off} can be calculated as $y_t^{off} = I_t - y_t^{on}$.

4.2. ML-PPO algorithm

4.2.1. Agent and environment

To solve this two-level POMDP model, we set two agents to learn the strategies respectively. The structure of each agent and the environment are shown in Fig. 2 and Fig. 3. We begin with training the Level II agent by interacting with the Level II environment and updating the network parameters to obtain the near-optimal policy. Then, we use the trained Level II agent as a part of the Level I environment, which can give a near-optimal policy under the replenishment quantity and get a cumulative reward of the order cycle to send back to the Level I agent.

4.2.2. Invalid action mask

The action space is discrete and state-dependent in the DPRR problems. Thus, not all actions are reasonable in certain states. If retailers try an unallowable action, computation time will be wasted and the agent may not receive an appropriate reward from the environment. In order to solve these actions, a technique named *Invalid action masking* has been proposed and used in recent works (Huang & Ontañón, 2022). Compared with an invalid action penalty, which means the agent will receive a negative reward when choosing invalid action, invalid action masking means that the agent will receive a mask so that only valid units can be selected. The core of invalid action masking is to put a 'mask' on the vector of the output action or value function, such as dot-multiply a vector $\{0, 1\}$ or $\{-\infty, 1\}$.

In the DPRR model, the order-up-to level $Q_{n\Gamma}$ is restricted to the range $[I_{n\Gamma}, Q_{max}]$ and the invalid action mask for the replenishment quantity is shown as follows:

$$mask^Q[i] = \begin{cases} 1 & if Q_{n\Gamma} \geq I_{n\Gamma} \\ 0 & if Q_{n\Gamma} < Q_{min} \end{cases} \quad \forall i \in [0, Q_{max}] \tag{10}$$

At each period t, the inventory rationing action y_t^{on} must be less than the current existing inventory level I_t and the price p_t must restrict to the range $[p_{min}, p_{max}]$. Thus, the invalid action mask for the inventory rationing and the price can be shown as follows:

$$mask_t^r[i] = \begin{cases} 1 & if y_t^{on} \leq I_t \\ 0 & if y_t^{on} > I_t \end{cases} \quad \forall i \in [0, I_0] \tag{11}$$

$$mask_t^p[i] = \begin{cases} 1 & if p_t \leq p_{min} \\ 0 & if p_t \geq p_{min} \end{cases} \quad \forall i \in [0, p_{max}] \tag{12}$$

4.2.3. Implementation of ML-PPO

PPO follows the Actor-Critic framework with an actor-network and a critic network. The actor selects the action based on the probability distribution, the critic judges the score based on the action generated by the actor, and the actor then modifies the probability of the selected action based on the critic's score. The input of both networks is the observation from the environment. The output of the actor-network is the action probability $\pi(a_t|s_t)$ for the discrete action, and the output of the critic network is an estimation of the expected future discounted profits of the current state. There are two primary variants of PPO. The first one is PPO-Penalty which approximates a KL-constrained, penalizes KL-divergence in the objective function instead of making it a hard constraint, and automatically adjusts the penalty factor during training. The second one is PPO-Clip which uses specialized clipping in the objective function to keep the new policy not far from the old one. In this paper, we mainly focus on PPO-Clip, which updates policy by Equation (13).

$$\theta_{k+1} = \text{arg}max\mathbb{E}_{s,a\sim\ \pi_{\theta_k}}(L(s, a, \theta_k, \theta)\ ) \tag{13}$$

The L is defined as

$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(s|a)}{\pi_{\theta_k}(s|a)}\widehat{A}_t, clip\left(\frac{\pi_\theta(s|a)}{\pi_{\theta_k}(s|a)}, 1 - \epsilon, 1 + \epsilon\right)\widehat{A}_t\right) \tag{14}$$

where ϵ is a hyperparameter that defines the maximum degree that the new policy can get away from the old one and $\widehat{A}_t$ is an estimator of the advantage function of period t.

To solve the unobservable state, we use the Long-Short-Term-Memory (LSTM) algorithm as a prediction model to forecast the demand in the future. LSTM is a special RNN, mainly solving gradient disappearance and gradient explosion problems while training long sequences (Hochreiter & Schmidhuber, 1997). For Level I, LSTM aims to forecast the sum demand during an order cycle to instruct the replenishment decisions. For Level II, LSTM is trained to forecast the online and offline demand in the future. L^p is defined as the prediction length.

Furthermore, a history data set $\mathscr{H}_t$ is collected during the training process of ML-PPO to update the LSTM network according to the current environment. The length of history data is set to be L^h, and it collects the observations of demand of each past period as Equation (15). If the data length exceeds L^h, the oldest data is deleted to accommodate the newest data.

$$\mathscr{H}_t = \left\{ (p_{t-1}, d_{t-1}^{on}, d_{t-1}^{off}), (p_{t-2}, d_{t-2}^{on}, d_{t-2}^{off}), \cdots, \left(p_{t-L^h}, d_{t-L^h}^{on}, d_{t-L^h}^{off}\right) \right\} \tag{15}$$

In the ML-PPO algorithm, we concatenate the current observation and the predictions of the future state as the input of the PPO agent and use the invalid action mask to filter the unallowable actions. We summarize our ML-PPO method in Algorithm 1.

Algorithm 1: ML-PPO

```
Initial policy parameters $\theta_0$, initial value function parameters $\phi_0$
Input LSTM parameters $\eta_0$
For episode $k = 0$ to $M$ do
  Reset the marketing environment
  Initial state $I_0$, initial history data set $\mathscr{H}_0$
  For $t = 0$ to $T$ do
    Agent observes the current observation $o_t$
    LSTM predict the future demand $\overline{D}_t = \{d_{t+1}, \cdots, d_{t+L^p}\}$
    Concatenate the observation $o_t$ and prediction $\overline{D}_t$ and input into the PPO agent
    Mask the action space based on the state
    Select a valid action $a_t$ based on the policy $\pi_{\theta_k}$
    Execute the action $a_t$ and observe $\mathscr{R}_t$ and $o_{t+1}$
    Collect the set $\mathscr{D}_k = \{\tau_t\} = \{(o_0, a_0, \mathscr{R}_0), \cdots, (o_t, a_t, \mathscr{R}_t)\}$
    Compute rewards-to-go $\widehat{\mathscr{R}}_t$
    Compute advantage estimates, $\widehat{A}_t$ based on the current value function $V_{\phi_k}$
    Update the history data set $\mathscr{H}_t$:
      If the history data length exceeds $L^h$ Then
        Update $\mathscr{H}_{t+1} = \left(\mathscr{H}_t - \left(p_{t-L^h}, d_{t-L^h}^{on}, d_{t-L^h}^{off}\right)\right) \cup \left(p_{t+1}, d_{t+1}^{on}, d_{t+1}^{off}\right)$
      Else
```

(continued on next page)

(*continued*)

Algorithm 1: ML-PPO

```
            Update $\mathscr{H}_{t+1} = \mathscr{H}_t \cup \left(p_{t+1}, d^{on}_{t+1}, d^{off}_{t+1}\right)$
        End for
        Update the LSTM parameters $\eta_{k+1}$
        Update the policy by maximizing the PPO-Clip objective in Equation (13) via SGD with Adam
        Fit value function by regression on mean-square error:
            $\phi_{k+1} = \underset{\phi}{\mathrm{argmin}} \frac{1}{|\mathscr{D}_k|T} \sum_{\tau \in \mathscr{D}_k} \sum_{t=0}^{T} \left(V_\phi(o_t) - \widehat{\mathscr{R}}_t\right)^2$
    End for
```

5. Simulation experiments

We conduct several simulation experiments to evaluate the proposed ML-PPO algorithm. We evaluate the performance of the algorithm by the average episode cumulative profit and the service level, where an episode is equal to an order cycle. We test ten consecutive order cycles to avoid randomness in the results and use the average cumulative profit as the final result. We use three kinds of value iteration algorithms, which can obtain the exact solution in small-scale instances, as the baselines to evaluate the accuracy of the ML-PPO algorithm. Furthermore, we use the PPO algorithm as the baseline to evaluate the degree of improvement of the ML-PPO algorithm.

5.1. Environment setup

5.1.1. Algorithm parameters setting

To implement the ML-PPO algorithm, we performed some experiments to improve the performance of our ML-PPO algorithm by tuning the parameters and investigating the results of a base test case. We evaluated different neural network architectures, learning rates, and batch sizes. The parameters we used are listed in Table 3.

In the ML-PPO algorithm, we follow the actor-critic framework of the PPO algorithm, which consists of two neural networks, actor and critic, with neurons in the input and output layers corresponding to the dimensions of the state and decision variables, respectively, and two hidden layers, with 64 neurons each. The learning rate is 0.0001, the batch size is 64, and the clipping parameter is 0.2. Furthermore, the prediction network consists of three LSTM cells. Each cell has two LSTM layers with a width of 64 and adopts the Adam optimizer. The batch size is 128, and MSE is adopted as the loss function. In the invalid action mask approach, we set the output of invalid actions in the actor neural network to a small negative value, which is 10^{-8} in our experiment setting, before the softmax activation layer so that the probability of selecting these actions becomes negligible.

To train the prediction network, we conduct pre-experiments to collect training data and the network is well-trained before being used in the ML-PPO. During the pre-experiments, retailers make pricing decisions randomly to interact with the environment and the actions and observations of previous demand are stored as training data. Using these training data, we train the prediction network to efficiently predict the future online and offline demand, which can be directly used in the ML-PPO algorithm.

Table 3
Value of parameters in ML-PPO algorithm.

	Parameter	Value
PPO	Depth of NN	2
	Width of NN	64
	Learning rate	0.0001
	Batch size	64
	Clipping parameter	0.2
LSTM	LSTM cell	3
	Width of cell	64
	Batch size	128
	Loss function	MSE

All the networks are set up by Pytorch 1.12.0. The open-source Python library Gym is used to establish the interaction between the learning algorithms and the environments (Brockman et al., 2016). All models are run on a server with an NVIDIA GeForce RTX 3060 GPU and 32 GB RAM. We follow the simulation procedure as Goedhart et al., (2022b) that all the following models are simulated for 1,000,000 periods because of a warm-up period.

5.1.2. Simulation parameters setting

To evaluate the performance of the ML-PPO algorithm, we construct simulations of the DPRR model. We set up a base test and the value of the parameters is shown in Table 4. The market size is assumed to be constant, so the demand core state consists of several potential values of the price sensitivity and demand distribution, which are randomly selected by the environment at the same probability. For example, in the base test, there are two potential values for both β and λ. The demand core state contains four different demand patterns and the environment selects each with a probability of 0.25. Furthermore, the lost-sale penalty c^l is set to be 50 percent of the sales price of each period.

5.2. Experiment results

Since there are two levels in the DPRR model with different decision variables, we conduct the ML-PPO algorithm in each level of the DPRR model respectively to evaluate the performance and conduct robustness checks on the results.

5.2.1. Evaluation of ML-PPO algorithm

In this section, we conduct the ML-PPO algorithm on both the dynamic pricing and rationing problems in Level II and the dynamic inventory replenishment in Level I to evaluate the performance of ML-PPO.

5.2.1.1. Level II: Dynamic pricing and rationing. We first compare the performance of ML-PPO with three value iteration algorithms to test its accuracy in small-scale experiments. The parameters of the experiments are shown in Table 5. We train the ML-PPO algorithm under the base test

Table 4
Value of parameters in the base test of the DPRR model.

Parameter	Value	Explanation
Γ	3	Order Cycle
c^h	0.5	Holding cost per item per period
c^l	50 %	Lost-sales penalty per item
c^s	3	Shipping expense per item purchased online
c^o	5	Order cost per item
α	40	Market size
Φ	$\beta \in \{1.5, 2\}, \lambda \in \{0.4, 0.6\}$	Demand core state
ε_t	$[-2, -1, 0, 1, 2]$	Demand error term
p_{min}	10	The lower bound of the sales price
p_{max}	15	The upper bound of the sales price
Q_{max}	60	The upper bound of the order quantity
γ	0.99	Discount factor

Table 5
Values of parameters in small-scale experiments.

Experiment	Demand core state	State spaces	Observation spaces	Action spaces
Base test	$\beta = \{1.5, 2\}, \lambda = \{0.4, 0.6\}$	612	21,600	306
E-1	$\beta = 2, \lambda = 0.6$	153	12,150	306
E-2	$\beta = 2, \lambda = \{0.4, 0.6\}$	306	12,150	306
E-3	$\beta = \{1.5, 2\}, \lambda = 0.6$	306	21,600	306

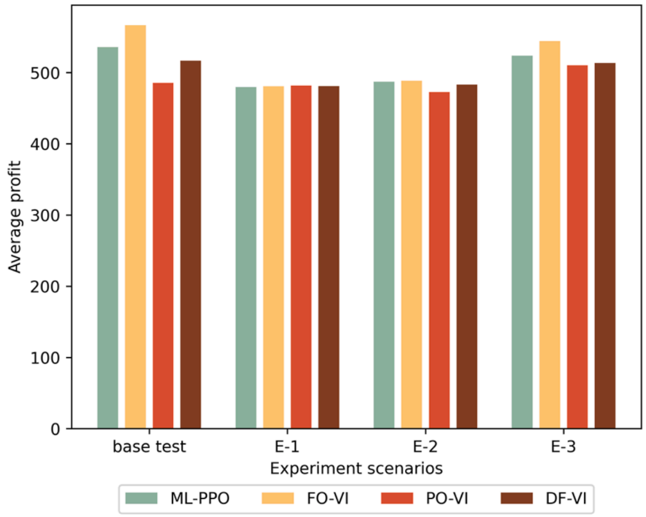


Fig. 4. Results of comparisons with FO-VI, PO-VI and DF-VI.

Table 6
Values of parameters in large-scale experiments.

Experiment	Demand core state	Prediction length	State spaces	Observation spaces	Action spaces
L-1	$\beta = 2, \lambda = 0.6$	1	2107	170,100	1806
L-2	$\beta = 2, \lambda \in \{0.4, 0.5, 0.6\}$	1	6321	170,100	1806
L-3	$\beta \in \{1.5, 2, 2.5\}, \lambda = 0.6$	1	6321	303,408	1806
L-4	$\beta \in \{1.5, 2, 2.5\}, \lambda \in \{0.4, 0.5, 0.6\}$	1	18,963	303,408	1806
L-5	$\beta \in \{1.5, 2, 2.5\}, \lambda \in \{0.4, 0.5, 0.6\}$	3	18,963	303,408	1806
L-6	$\beta \in \{1.5, 2, 2.5\}, \lambda \in \{0.4, 0.5, 0.6\}$	5	18,963	303,408	1806

parameters and discuss the impact of demand uncertainty by varying the number of potential values in the demand core state in Experiments E1-E3.

The three value iteration algorithms contain fully-observation value iteration (FO-VI), partially-observation value iteration (PO-VI) and demand forecast value iteration (DF-VI).[1] The solution of FO-VI can be seen as the exact solution to the DPR problem with full observation. The ML-PPO and DF-VI algorithms solve the DPR problem with partial observation and demand forecasting and PO-VI algorithms have no idea about the demand. Fig. 4 shows the results of the comparative experiments of the three value iteration algorithms and ML-PPO algorithms by the average episode profit of the three algorithms in each experiment. In all the tests, the performance of the FO-VI algorithm was consistently the best. In the four different demand environments, the ML-PPO algorithm performs second best, slightly outperforming the DF-VI algorithm and significantly outperforming the PO-VI algorithm. Thus, the results show that the uncertainty of the demand market brings more difficulty in the DPR problem, and the ML-PPO can obtain near-optimal solutions under the uncertain environment.

Then, we compare the results with different degrees of demand market uncertainty. When there is no demand market uncertainty in Experiment E-1, the average profits of the four algorithms are almost consistent. Comparing the results of Experiments E-1-E-3, it is obvious that as the demand market uncertainty increases, the optimal gap between ML-PPO and FO-VI becomes more significant, which indicates the negative effect of the demand uncertainty on the retailers' profit. Furthermore, the sources of demand market uncertainty are analyzed. In Experiment E-2, there is only uncertainty in the demand distribution and the average profits of ML-PPO and FO-VI do not have significant differences, which means that fluctuations in the demand distribution have little impact on pricing and rationing decisions. In Experiment E-3, demand price sensitivity varies randomly with a constant demand distribution, and the gaps between the three algorithms occur, which means that demand price sensitivity is a vital factor in retailers' decisions and profits.

Table 7
Average episode profit of PPO, ML-PPO and FM-GD.

Experiment	PPO	ML-PPO	FM-GD	PPO/ML-PPO	ML-PPO/FM-GD
L-1	3250.74	3279.97	3290.63	99.81 %	99.68 %
L-2	3342.23	3350.34	3392.86	99.76 %	98.75 %
L-3	3202.44	3236.26	3261.57	98.95 %	99.22 %
L-4	3286.85	3346.21	3355.97	98.22 %	99.71 %
L-5	3286.85	3364.42	3355.97	97.69 %	100.25 %
L-6	3286.85	3295.09	3355.97	99.75 %	98.19 %

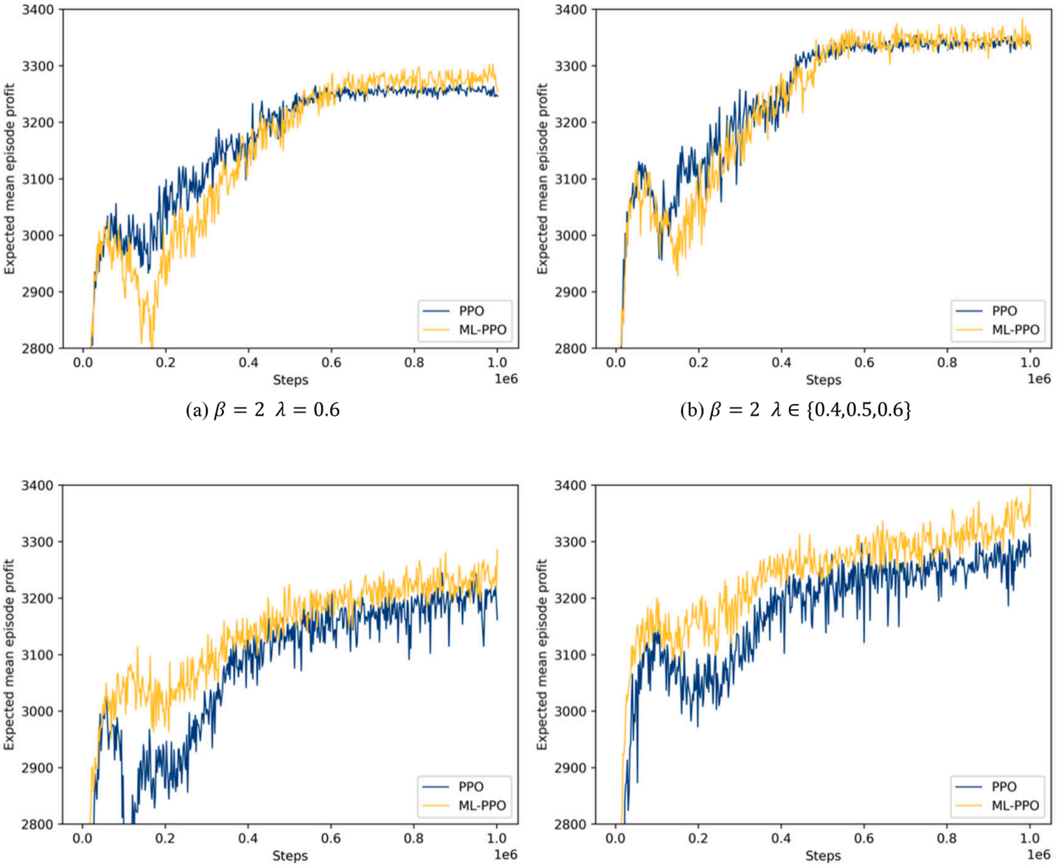
Subsequently, we scale up the problem scenarios to a large-scale setting: the order cycle consists of seven periods ($\Gamma = 7$), the market size α increases to 70, the initial inventory level I_0 increases to 300 and the uncertainty of the demand market increase to nine potential demand patterns in the demand core state Φ. Six sets of experiments are designed by varying the degree of environmental uncertainty and the demand forecasting length in the ML-PPO algorithm, as presented in Table 6. Given the suboptimal performance of the VI algorithm in large-scale settings, we employ the PPO algorithm and a full-information myopic grid search approach (FM-GD) for comparative evaluations in this section.

In the FM-GD approach, we assume that the retailer has access to full information at the beginning of each period, such as demand patterns and consumer preferences. Leveraging this full information, the retailer performs a grid search within the feasible range of decision variables to determine the myopic pricing and inventory rationing decisions that maximize the profit for the current period. Table 7 presents a comparison of the average episode profits for the three methods across the six experimental designs.

From Table 7, it can be observed that the average episode profits of ML-PPO are slightly lower than FM-GD approach, and surpass FM-GD when the prediction length is set to 3. This result suggests that ML-PPO, when dealing with partially observable problems, can utilize interactive learning and demand forecasting to acquire information and make relatively superior decisions. Additionally, ML-PPO outperforming FM-GD validates the advantage of long-term decision-making over short-term to obtain a higher profit. Therefore, when retailers make decisions, they should consider short-term and long-term benefits in a balanced manner to ensure decision efficiency while maintaining higher profits.

Moreover, when comparing ML-PPO and PPO, it is clear that ML-PPO outperforms in all experiments and the best improvement degree can reach about 2.4 % when the prediction length is 3. Additionally, the training processes of PPO and ML-PPO in all experiments are shown in Fig. 5 and Fig. 6. Fig. 5 shows the training process of PPO and ML-PPO under different demand markets. During the training process, in the four scenarios, the overall convergence trends of the algorithms do not change significantly, but ML-PPO outperforms the PPO algorithm in terms of average profits. The results reflect that compared to PPO,

[1] Details in Appendix A.



(a) $\beta = 2 \ \ \lambda = 0.6$

(b) $\beta = 2 \ \ \lambda \in \{0.4,0.5,0.6\}$

(c) $\beta \in \{1.5,2,2.5\} \ \ \lambda = 0.6$

(d) $\beta \in \{1.5,2,2.5\} \ \ \lambda \in \{0.4,0.5,0.6\}$

Fig. 5. Training process of PPO and ML-PPO under different demand market.

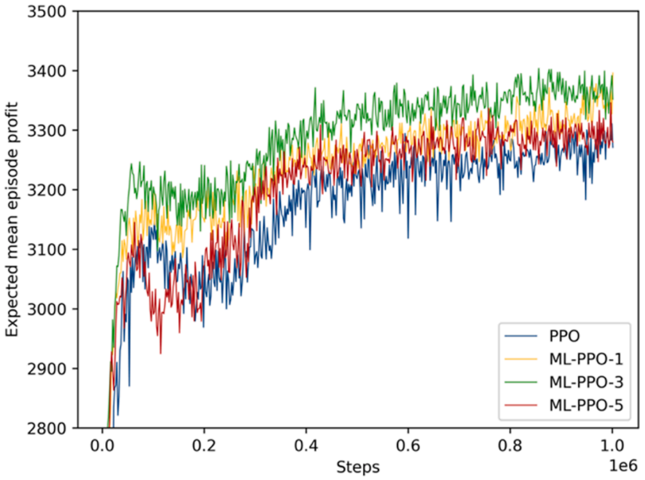


Fig. 6. Training process of algorithms in large-scale experiments.

although adding a prediction component as ML-PPO cannot accelerate the convergence speed of the algorithms, it can effectively improve the accuracy of a retailer's decisions in the face of both stable and uncertain markets. Furthermore, it is clear from the four figures in Fig. 5 that as the number of core demand states increases, the fluctuations of the average episode profit become larger even when the profit is nearly converged. This phenomenon is reasonable because of the uncertainty of demand, which greatly impacts the final profit and reflects the difficulty of decision-making under the uncertain environment from the side.

Fig. 6 shows the training process of PPO and ML-PPO with different prediction lengths L^p. It is clear that when the profits converge, the ML-PPO results are significantly better than the PPO, which means that the proposed algorithm can improve the accuracy of the solution. Furthermore, there are three different prediction lengths L^p varying from 1 to 5. Contrary to our speculation, it is not the case that the longer the prediction length provides, the better the results. The ML-PPO algorithm performs the best for $L^p = 3$, followed by $L^p = 1$, and finally $L^p = 5$. One of the reasons for this phenomenon is that the error of the prediction algorithm increases with the prediction length. Although a longer prediction period provides more information to the retailer, its reduced accuracy can also lead to biased decisions.

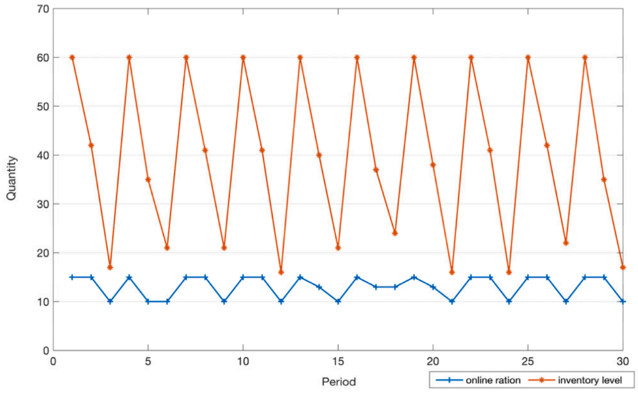


Fig. 7. Rationing strategy and inventory level change in base test.

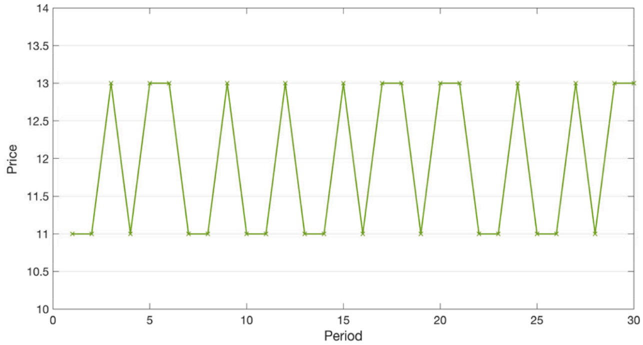


Fig. 8. Optimal pricing strategy in base test.

5.2.1.2. Level I: Dynamic inventory replenishment. Based on the trained ML-PPO algorithm of the DPR problem in the base test, we set up the dynamic inventory replenishment (DR) environment in Level I with the trained agent of Level II, as shown in Fig. 3 as the complete DPRR model, and conduct experiments on it in this sector. Inventory replenishment is made at each order point, and it concentrates on the cumulative profit during the order cycle instead of the instance profit. Although Level I and Level II have different action spaces and action periods, they face the same environment. With the extension from Level II to Level I, the state space is more complex, and it is difficult for the VI algorithms to solve in an acceptable time. Thus, we use ML-PPO and PPO algorithms to solve the DR problems. Additionally, except for the profit, we also consider the service level as a performance indicator of the solutions. A simple way to calculate the service level is to take the number of items sold and the number of items that could not be sold due to a lack of stock.

Figs. 7 and 8 illustrate the decision paths for dynamic ordering, rationing, and pricing obtained for the base test. At the ordering point, the optimal order quantity remains stable at the maximum order quantity ($Q = Q_{max} = 60$). Due to this stable order quantity, the initial inventory at the beginning of each ordering cycle is the same. As a result, the optimal pricing decisions and inventory rationing decisions exhibit relatively stable oscillations over ordering cycles, and they are dynamically adjusted in coordination based on the current remaining inventory level of each period.

According to the base test, we design four experiments that vary the uncertainty of the environments to evaluate the performance of the ML-PPO algorithm in solving the DPRR problems. The results are presented in Table 8.

Table 8 shows that the ML-PPO algorithm has a higher profit and service level than the PPO algorithm in the base test, and the performance of the ML-PPO algorithm is stable or even better when varying the demand uncertainty in the DPRR models. The optimal cumulative profit of ML-PPO exceeds PPO by up to 6 %. The service level of ML-PPO is about 1.4 % higher than the PPO algorithm, which means that the inventory quantity can fulfill more demand over an order cycle. The service levels can reach 1 with the ML-PPO algorithm when the demand market uncertainty is low. As the demand uncertainty increases, the solution difficulty increases, resulting in lower service levels and profits. These results indicate that the ML-PPO algorithm consistently outperforms the basic PPO algorithm across various demand environments.

5.2.2. Robustness checks

In this section, we conduct several additional experiments to examine the robustness of the numerical results. We achieve this by systematically varying the parameter settings in the proposed model. By exploring different scenarios, we can gain insights into the sensitivity of the model to different input parameters and better understand the implications of these variations on the overall outcomes.

5.2.2.1. Level II: Dynamic pricing and rationing. First, we test the robustness of our results of DPR problem by varying the values of the initial inventory level, and several costs of the base test to compare the average profit of ML-PPO, FO-VI, PO-VI and DF-VI algorithms. Experiments R1-R2 vary the initial inventory level I_0 from 40 to 60, and Experiments R3-R5 change the value of holding cost, lost sale penalty, and shipping expense, respectively. The experiment settings and the results are summarized in Table 9.

In Table 9, the average profits of the trained ML-PPO algorithm with different initial inventory levels are compared. When the initial inventory level is insufficient, the profits in all algorithms decrease significantly, and the gap between ML-PPO and FO-VI is stable, which means that ML-PPO can solve the insufficient inventory scenario efficiently. When the initial inventory level is sufficient, the average profits remain stable, and ML-PPO, PO-VI and DF-VI can achieve almost the same profit. On the other hand, the results reflect that the initial inventory level plays an important role in the profit, and the initial inventory level is the decision variable of the Level I problem. Therefore, conducting the latter experiments on Level I problems is meaningful.

Then, we vary the economic parameters, including holding cost, lost-sales penalty and shipping expense parameters. The gap between ML-PPO and FO-VI becomes larger when the holding cost increases, which

Table 8
Profit and service level for DPRR problems of PPO and ML-PPO.

Experiments	Cumulative Profit			Service Level		
	ML-PPO	PPO	PPO/ML-PPO	ML-PPO	PPO	PPO/ML-PPO
$\beta = 2, \lambda_t = 0.6$	2860.81	2732.23	95.51 %	1	0.986	98.60 %
$\beta = 2, \lambda_t = \{0.4, 0.6\}$	2975.22	2938.04	98.75 %	1	0.991	99.10 %
$\beta = \{1.5, 2\}, \lambda_t = 0.6$	3089.93	2903.22	93.96 %	0.979	0.969	98.98 %
$\beta = \{1.5, 2\}, \lambda_t = \{0.4, 0.6\}$ [a]	3183.75	3005.88	94.41 %	0.984	0.972	98.78 %

[a] Base test.

Table 9
Robustness check results of DPR problems.

Experiment	I_0	c^h	c^l	c^s	Average profit			
					ML-PPO	FO-VI	PO-VI	DF-VI
Base test	50	0.5	50 %	3	535.9	566.5	485.61	516.73
R-1	40	0.5	50 %	3	451.5	467.17	393.1	449.5
R-2	60	0.5	50 %	3	539.9	566.22	540.64	542.56
R-3	50	1	50 %	3	494	541.17	441.71	490.26
R-4	50	0.5	70 %	3	518.7	550.72	471.43	513.42
R-5	50	0.5	50 %	5	486.1	521.85	439.53	474.93

Table 10
Robustness check results of DPRR problems.

Experiments	Cumulative Profit			Service Level		
	ML-PPO	PPO	PPO/ML-PPO	ML-PPO	PPO	PPO/ML-PPO
$c^h = 0.5$ [a]	3183.75	3005.88	94.41 %	0.984	0.972	98.78 %
$c^h = 1$	3144.89	2914.02	92.66 %	0.976	0.968	99.18 %
$c^l = 50\%$ [a]	3183.75	3005.88	94.41 %	0.984	0.972	98.78 %
$c^l = 70\%$	3486.32	3269.16	93.77 %	0.988	0.976	98.79 %
$c^s = 1$ [a]	3183.75	3005.88	94.41 %	0.984	0.972	98.78 %
$c^s = 3$	2596.04	2176.86	83.85 %	0.983	0.973	98.98 %
$c^o = 3$	3844.64	3558.56	92.56 %	0.987	0.978	99.09 %
$c^o = 5$ [a]	3183.75	3005.88	94.41 %	0.984	0.972	98.78 %
$c^o = 7$	2408.24	1964.33	81.57 %	0.963	0.911	94.60 %

[a] Base test.

means that the rationing decisions have a significant impact on the inventory quantity and then reflect in the final profit of the retailers. Meanwhile, the increase in lost-sale cost and shipping cost causes a decrease in profit, but the performance of ML-PPO is stable and efficient. The main results remain qualitatively the same in the robustness check experiments, which shows that the proposed ML-PPO can solve problems efficiently, even under more negative conditions.

5.2.2.2. Level I: Dynamic inventory replenishment. According to the base test, we design another four experiments that vary the values of the holding cost, the lost-sale cost, the shipping expense, and the order cost per item to test the robustness of our results. The experiment settings and the results are summarized in Table 10.

From Table 10, it is observed that when the holding cost increases, both the profit and the service level decrease. This is mainly the result of the balance between profit and service level: if more items are ordered at the order point, more items are likely to fulfill the demand and be stored in storage. More items fulfilled means an increase in service level, and more items in storage mean a high holding cost, which reduces the profit. Thus, the increased holding cost increases the pressure on inventory management, leading to reduced profits and lower service levels. Under this pressure, the ML-PPO algorithm can still achieve a higher profit and service level than the PPO algorithm.

In contrast, the increased lost-sale cost leads to higher profit and service levels in both algorithms. The lost-sale cost is an index of possible losses caused by out-of-stocks, and a higher lost-sale cost means that the retailer will pay more penalty for out-of-stocks. Thus, the higher lost-sale cost will raise the attention of retailers and avoid the loss of demand for this item and the service level will improve at the same time. As for the profit, although the increase in cost is more likely to cause a reduction in profit, the service level is high and the reduction can be covered by the income of meeting more demand. The increase in the shipping expense will result in a decrease in profit but will have little impact on the service level.

With the changes in the order cost, the profit and service level have also changed. The base level is set at 5, and when it is reduced to 3, the profit has a significant increase, especially in the ML-PPO algorithm, and the service level increases to a higher level of 0.987. Meanwhile, when the ordering cost increases to 7, the profit reduces rapidly, and the reduction degree of the ML-PPO algorithm is less than that of the PPO algorithm, which means that the ML-PPO algorithm has more ability to deal with more terrible conditions. It is the same with the reduction of the service level. The changing pattern is easy to interpret that the ordering cost is the most direct parameter associated with the inventory replenishment. The more items are ordered, the more costs have to be paid. Thus, under this condition, the retailer has no incentive to order more items to fulfill more demand at the cost of losing more profit. In general, our main results remain qualitatively the same across all the experiments.

5.3. Discussion

It is worth further discussing that, in our simulation experiments, the training data is obtained through online interactions between the agent and the simulated environment. However, in real business scenarios, retailers cannot engage in extensive interactions with the real environment to acquire data and train models. Therefore, how to train and apply this method in real-world situations is a challenge we need to address further.

Based on our research findings, we suggest that employing a 'market simulator' to assist training can facilitate the practical implementation of the method, which is also proposed by (Afshar, Rhuggenaath, Zhang, & Kaymak, 2023; Qiao, Huang, Gao, & Wang, 2024). In this approach, retailers are not required to engage extensively with the real market environment to collect information and train models. Instead, they can derive a market simulator based on historical sales data and achieve model training through extensive interactions with the simulator.

Undeniably, the market simulator may exhibit certain disparities compared to the actual market environment. Hence, when retailers interact with the real environment using decision results from the model, continuous optimization and adjustments to the market simulator can be implemented by comparing expected outcomes from the simulator with actual results. The real interaction data, in turn, serves as training data for further refining the model. This approach thus facilitates effective training and optimization of the DRL algorithm within a simulated environment, aligning with the intricacies of real-world business scenarios.

6. Conclusion

In this paper, we studied the dynamic pricing and inventory management problem of omni-channel retailers who make daily pricing and inventory rationing decisions across online and offline channels and replenish at order points in an uncertain demand market. To maximize the retailer's profit, we developed a DPRR model and used a two-level POMDP to describe the dynamic process since the decisions have different cycles and some states of the market environment are unobservable. We proposed to use an ML-PPO algorithm, which concatenates the observation of the environment and the predictions of the future state as the input of the PPO agent and uses the invalid action mask to filter the unallowable actions. Several simulation experiments were conducted to evaluate the performance of the ML-PPO algorithm by comparing it to the FO-VI algorithm, PO-VI algorithm, and PPO algorithm. The results reveal that the ML-PPO algorithm can obtain near-optimal solutions to the DPRR problems and outperform the PPO algorithm in terms of the retailer's profit and service level. Meanwhile, the generalization ability of the ML-PPO algorithm is verified across various market environments, making it practical for omni-channel retailers to deal with DPRR problems. These findings have significant implications for the retail industry and offer a new approach to optimizing pricing and inventory strategies in the era of omni-channel retailing. Furthermore, while we discussed how to implement the proposed algorithm in real-world, there is still room for a more detailed exploration of this issue to arrive at a better solution.

Furthermore, the results of our study also provide valuable management insights for omni-channel retailers. Firstly, the significant improvement in ML-PPO algorithms imply the importance of using historical data to predict future demand accurately. Thus, retailers should also pay more attention to research on accurate prediction of future demand to support better decision-making. Additionally, when retailers make decisions about which products to sell or which markets to target, they should take the uncertainty of demand into account. Focusing on products or markets with more predictable demand patterns can significantly reduce decision-making complexity and increase the likelihood of achieving expected profits. Moreover, our findings highlight that retailers' profit is not only dependent on pricing and inventory

Table A1
Linear regression results of each experiment.

			Unstandardized Coefficients		Standardized Coefficients	t-value	p-value
			B	Std. Error	Beta		
Base test	Online demand	Constant	20.690	0.189	–	109.420	0.000**
		price	−0.901	0.015	−0.515	−60.105	0.000**
	Offline demand	Constant	20.343	0.191	–	106.468	0.000**
		price	−0.868	0.015	−0.497	−57.304	0.000**
E-1	Online demand	Constant	24.395	0.067	–	363.130	0.000**
		price	−1.200	0.005	−0.914	−225.384	0.000**
	Offline demand	Constant	16.410	0.045	–	366.276	0.000**
		price	−0.801	0.004	−0.914	−225.588	0.000**
E-2	Online demand	Constant	16.286	0.104	–	157.252	0.000**
		price	−0.691	0.008	−0.644	−84.158	0.000**
	Offline demand	Constant	20.361	0.128	–	159.436	0.000**
		price	−0.999	0.010	−0.703	−98.912	0.000**
E-3	Online demand	Constant	24.165	0.157	–	154.243	0.000**
		price	−1.029	0.012	−0.638	−82.902	0.000**
	Offline demand	Constant	16.286	0.104	–	157.252	0.000**
		price	−0.691	0.008	−0.644	−84.158	0.000**

$*p < 0.05$ $**p < 0.01$.

Table A2
Demand functions in each experiment.

	Online demand	Offline demand
Base test	$d_{on} = 20.690 - 0.901{*}p$	$d_{off} = 20.343 - 0.868{*}p$
E-1	$d_{on} = 24.395 - 1.2{*}p$	$d_{off} = 16.410 - 0.801{*}p$
E-2	$d_{on} = 20.361 - 0.999{*}p$	$d_{off} = 20.422 - 1.001{*}p$
E-3	$d_{on} = 24.165 - 1.029{*}p$	$d_{off} = 16.286 - 0.691{*}p$

management decisions but also influenced by various costs incurred during the process. Thus, retailers should not only focus on pricing and inventory strategies but also consider optimizing management costs, such as holding costs, shipping costs, and lost sales costs, to improve their profitability.

With respect to future studies, first, while this paper has considered some factors within the context of omni-channel retailing, there are still several other factors that could be incorporated into the model. For example, positive lead time, substitution or competition effects between products or channels, and omni-channel retailers with more than two channels are all aspects that could be further considered in the model. Second, the parameters of the experiments are subjective, although they refer to previous literature and several data from a real supermarket to be more practical. It is more meaningful to test the algorithm in a real market with an omni-channel retailer in the future. Finally, although the design of the prediction idea is tested to be helpful, the choice of the prediction algorithm can be further discussed, and the performance may be improved by using other machine learning algorithms.

Funding

This work was supported by the National Natural Science Foundation of China [grant numbers 72171008].

CRediT authorship contribution statement

Shiyu Liu: Conceptualization, Methodology, Writing – original draft. **Jun Wang:** Supervision, Writing – review & editing. **Rui Wang:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Yue Zhang:** Conceptualization, Writing – review & editing, Supervision. **Yanjie Song:** Resources, Formal analysis, Supervision. **Lining Xing:** Formal analysis, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix A

In small-scale experiments, we conducted a linear regression analysis of demand data, resulting in significant linear regression equations for both online and offline demand in relation to price. The explanatory variable for linear regression is the unified price p, and separate linear regressions were conducted for offline demand d_{off} and online demand d_{on} under several experiment settings. Using this as the model foundation, we employed value iteration to obtain approximate optimal strategies for dynamic pricing and inventory rationing under demand forecasting, as DF-VI algorithm. To obtain the demand data, we randomly selected prices within the price range and interacted with the market environment, observing the online and offline demand quantities corresponding to each price. Through 10^4 interactions, we obtained demand observation datasets under different demand scenarios for the purpose of fitting linear regression equations. We conducted linear regression analysis on the data using SPSS and selected the key indicators, which are summarized in the table below. The data in the Table A1 indicates the significance of the regression and the demand equations fitted for each experimental scenario are shown in Table A2.

References

Afshar, R. R., Rhuggenaath, J., Zhang, Y., & Kaymak, U. (2023). An automated deep reinforcement learning pipeline for dynamic pricing. *IEEE Transactions on Artificial Intelligence, 4*(3), 428–437. https://doi.org/10.1109/TAI.2022.3186292

Aviv, Y., & Pazgal, A. (2005). A partially observed markov decision process for dynamic pricing. *Management Science, 51*(9), 1400–1416. https://doi.org/10.1287/mnsc.1050.0393

Bardhan, S., Pal, H., & Giri, B. C. (2019). Optimal replenishment policy and preservation technology investment for a non-instantaneous deteriorating item with stock-dependent demand. *Operational Research, 19*(2), 347–368. https://doi.org/10.1007/s12351-017-0302-0

Batarfi, R., Jaber, M. Y., & Glock, C. H. (2019). Pricing and inventory decisions in a dual-channel supply chain with learning and forgetting. *Computers & Industrial Engineering, 136*, 397–420. https://doi.org/10.1016/j.cie.2019.07.034

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *OpenAI Gym*. https://doi.org/10.48550/arXiv.1606.01540

Cao, L., & Li, L. (2015). The impact of cross-channel integration on retailers' sales growth. *Journal of Retailing, 91*(2), 198–216. https://doi.org/10.1016/j.jretai.2014.12.005

Cárdenas-Barrón, L. E., Shaikh, A. A., Tiwari, S., & Treviño-Garza, G. (2020). An EOQ inventory model with nonlinear stock dependent holding cost, nonlinear stock dependent demand and trade credit. *Computers & Industrial Engineering, 139*, Article 105557. https://doi.org/10.1016/j.cie.2018.12.004

Chen, B., Chao, X., & Wang, Y. (2020). Data-based dynamic pricing and inventory control with censored demand and limited price changes. *Operations Research, 68*(5), 1445–1456. https://doi.org/10.1287/opre.2020.1993

Chen, X., & Hu, P. (2012). Joint pricing and inventory management with deterministic demand and costly price adjustment. *Operations Research Letters, 40*(5), 385–389. https://doi.org/10.1016/j.orl.2012.05.011

De Moor, B. J., Gijsbrechts, J., & Boute, R. N. (2022). Reward shaping to improve the performance of deep reinforcement learning in perishable inventory management. *European Journal of Operational Research, 301*(2), 535–545. https://doi.org/10.1016/j.ejor.2021.10.045

Ding, Y., Feng, M., Liu, G., Jiang, W., Zhang, C., Zhao, L., … Bian, J. (2022). *Multi-Agent Reinforcement Learning with Shared Resources for Inventory Management*. https://doi.org/10.48550/arXiv.2212.07684.

Elmaghraby, W., & Keskinocak, P. (2003). Dynamic pricing in the presence of inventory considerations: research overview, current practices, and future directions. *Management Science, 49*(10), 1287–1309. https://doi.org/10.1287/mnsc.49.10.1287.17315

Fang, F., Nguyen, T.-D., & Currie, C. S. (2021). Joint pricing and inventory decisions for substitutable and perishable products under demand uncertainty. *European Journal of Operational Research, 293*(2), 594–602. https://doi.org/10.1016/j.ejor.2020.08.002

Feng, Q., Luo, S., & Shanthikumar, J. G. (2020). Integrating Dynamic Pricing with Inventory Decisions Under Lost Sales. *Management Science, 66*(5), 2232–2247. https://doi.org/10.1287/mnsc.2019.3299

Goedhart, J., Haijema, R., & Akkerman, R. (2022a). Inventory rationing and replenishment for an omni-channel retailer. *Computers & Operations Research, 140*, Article 105647. https://doi.org/10.1016/j.cor.2021.105647

Goedhart, J., Haijema, R., & Akkerman, R. (2022b). Modelling the influence of returns for an omni-channel retailer. S0377221722006646 *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2022.08.021.

Gupta, V. K., Ting, Q. U., & Tiwari, M. K. (2019). Multi-period price optimization problem for omnichannel retailers accounting for customer heterogeneity. *International Journal of Production Economics, 212*, 155–167. https://doi.org/10.1016/j.ijpe.2019.02.016

He, Y., Huang, H., & Li, D. (2020). Inventory and pricing decisions for a dual-channel supply chain with deteriorating products. *Operational Research, 20*(3), 1461–1503. https://doi.org/10.1007/s12351-018-0393-2

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, S., & Ontañón, S. (2022). *A Closer Look at Invalid Action Masking in Policy Gradient Algorithms*. https://doi.org/10.32473/flairs.v35i.130584.

Jalilipour Alishah, E., Moinzadeh, K., & Zhou, Y.-P. (2015). Inventory Fulfillment Strategies for an Omni-Channel Retailer. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2659671

Keskin, N. B., Li, Y., & Song, J.-S. (2022). Data-driven dynamic pricing and ordering with perishable inventory in a changing environment. *Management Science, 68*(3), 1938–1958. https://doi.org/10.1287/mnsc.2021.4011

Lei, Y., Jasin, S., & Sinha, A. (2018). Joint dynamic pricing and order fulfillment for e-commerce retailers. *Manufacturing & Service Operations Management, 20*(2), 269–284. https://doi.org/10.1287/msom.2017.0641

Li, M., & Mizuno, S. (2022). Dynamic pricing and inventory management of a dual-channel supply chain under different power structures. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2022.02.049

Liu, J., & Xu, Q. (2020). Joint decision on pricing and ordering for omnichannel BOPS retailers: considering online returns. *Sustainability, 12*(4), 1539. https://doi.org/10.3390/su12041539

Mou, S., Robb, D. J., & DeHoratius, N. (2018). Retail store operations: Literature review and research directions. *European Journal of Operational Research, 265*(2), 399–422. https://doi.org/10.1016/j.ejor.2017.07.003

Neghab, D. P., Khayyati, S., & Karaesmen, F. (2022). An integrated data-driven method using deep learning for a newsvendor problem with unobservable features. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2021.12.047

Oroojlooyjadid, A., Nazari, M., Snyder, L. V., & Takáč, M. (2022). A Deep Q-Network for the Beer Game: Deep reinforcement learning for inventory optimization. *Manufacturing & Service Operations Management, 24*(1), 285–304. https://doi.org/10.1287/msom.2020.0939

Qiao, W., Huang, M., Gao, Z., & Wang, X. (2024). Distributed dynamic pricing of multiple perishable products using multi-agent reinforcement learning. *Expert Systems with Applications, 237*, Article 121252. https://doi.org/10.1016/j.eswa.2023.121252

Qiu, R., Ma, L., & Sun, M. (2023). A robust omnichannel pricing and ordering optimization approach with return policies based on data-driven support vector clustering. *European Journal of Operational Research, 305*(3), 1337–1354. https://doi.org/10.1016/j.ejor.2022.07.029

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. https://doi.org/10.48550/arXiv.1707.06347.

Sepehri, A., Mishra, U., Tseng, M.-L., & Sarkar, B. (2021). Joint pricing and inventory model for deteriorating items with maximum lifetime and controllable carbon emissions under permissible delay in payments. *Mathematics, 9*(5), 470. https://doi.org/10.3390/math9050470

Simchi-Levi, D., & Agrawal, N. (Eds.). (2004). *Handbook of quantitative supply chain analysis: Modeling in the e-business era*. Boston: Kluwer Acad. Publ.

Teunter, R. H., & Klein Haneveld, W. K. (2008). Dynamic inventory rationing strategies for inventory systems with two demand classes, Poisson demand and backordering. *European Journal of Operational Research, 190*(1), 156–178. https://doi.org/10.1016/j.ejor.2007.06.009

Topkis, D. M. (1968). Optimal ordering and rationing policies in a nonstationary dynamic inventory model with n demand classes. *Management Science, 15*(3), 160–176. https://doi.org/10.1287/mnsc.15.3.160

Turgay, Z., Karaesmen, F., & Örmeci, E. L. (2015). A dynamic inventory rationing problem with uncertain demand and production rates. *Annals of Operations Research, 231*(1), 207–228. https://doi.org/10.1007/s10479-014-1573-y

Wang, H., Tao, J., Peng, T., Brintrup, A., Kosasih, E. E., Lu, Y., & Hu, L. (2022). Dynamic inventory replenishment strategy for aerospace manufacturing supply chain: Combining reinforcement learning and multi-agent simulation. *International Journal of Production Research, 1–20*. https://doi.org/10.1080/00207543.2021.2020927

Wang, R., Gan, X., Li, Q., & Yan, X. (2021). Solving a Joint Pricing and Inventory Control Problem for Perishables via Deep Reinforcement Learning. *Complexity, 2021*, 1–17. https://doi.org/10.1155/2021/6643131

Wu, C., Bi, W., & Liu, H. (2023). Proximal policy optimization algorithm for dynamic pricing with online reviews. *Expert Systems with Applications, 213*, Article 119191. https://doi.org/10.1016/j.eswa.2022.119191

Wu, J., Zhao, C., Yan, X., & Wang, L. (2020). An Integrated Randomized Pricing Strategy for Omni-Channel Retailing. *International Journal of Electronic Commerce, 24*(3), 391–418. https://doi.org/10.1080/10864415.2020.1767434

Yang, C., Feng, Y., & Whinston, A. (2022). Dynamic Pricing and Information Disclosure for Fresh Produce: An Artificial Intelligence Approach. *Production and Operations Management, 31*(1), 155–171. https://doi.org/10.1111/poms.13525

Zhou, Q., Yang, Y., & Fu, S. (2022). Deep reinforcement learning approach for solving joint pricing and inventory problem with reference price effects. *Expert Systems with Applications, 195*, Article 116564. https://doi.org/10.1016/j.eswa.2022.116564