International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 29-44

https://doi.org/10.70517/ijhsa46403

Reinforcement Learning Optimization Strategies for Dynamic Pricing and Inventory Control in E-commerce Retail

Sijie Huang^{1} and Yan Yang^{2,*}
^{1} School of Management, Zhanjiang University of Science and Technology, Zhanjiang, Guangdong, 524086, China
^{2} School of Economics and Finance, Zhanjiang University of Science and Technology, Zhanjiang, Guangdong, 524086, China
Corresponding authors: (e-mail: yangniu30@outlook.com).

Abstract With the continuous development of Internet e-commerce, reasonable inventory arrangement for different warehouses or retailers and dynamic pricing of goods have gradually become key factors affecting the profitability of each company. The study proposes a reinforcement learning-based approach for dynamic pricing and inventory control in e-commerce retailing. The problem is modeled and converted into a Markov decision process by incorporating e-commerce retailing characteristics, and a joint inventory control and dynamic pricing algorithm for e-commerce retailing is designed based on the Deep Deterministic Policy Gradient (DDPG) method. The results of numerical experiments show that the joint inventory control and dynamic pricing strategies based on deep reinforcement learning have the best performance in terms of gains, with gains of 0.197 and 0.035, respectively. The numerical experiments validate the performance effectiveness of the algorithms proposed in this paper, and the DDPG algorithm significantly outperforms the traditional methods. This research can improve enterprise revenue and effectively promote the landing of reinforcement learning in the field of revenue management, which has practical application value.

Index Terms markov decision making, DDPG, inventory control, dynamic pricing, e-commerce retailing

I. Introduction

The rise of the Internet and the continuous development of e-commerce, network payment technology as well as logistics and distribution services continue to improve, the rapid development of online retail channels, changing people's lives and ways of consumption, online consumption is more and more favored by consumers, online marketing along the way to become today's hottest marketing promotion [1]-[3]. Compared with traditional retailing, e-retail stores have become an emerging marketing channel for retailers with unique advantages [4]. Under the thriving trend of e-commerce industry, major retailers' management decisions consider more about how to set the price of fresh produce and reduce the cost to maximize the revenue [5], [6]. However, due to the particularity that the value of the product decreases with the passage of time, it requires specialized management, which is time-consuming and labor-intensive, and the actual management process is difficult, with a large proportion of inventory costs [7]-[9].

Dynamic pricing is the process of adjusting the price of goods sold according to changes in channels, quality, customers, and time [10]. Many researchers have paid much attention to the problem of dynamic pricing of products and have conducted numerous studies. And in this field, demand is an important influence in analyzing the acquisition of optimal pricing mechanisms [11], [12]. Some scholars have used the willingness to pay (WTP) model to measure the impact of pricing strategies on consumers' purchasing decisions, in which product scarcity has also been the focus of many scholars [13]-[15]. Zhao, Y et al. examined the impact of online and offline sales channels on strategic customer behavior and pricing strategies for horizontally differentiated products under two dual-channel models [16]. Dai, L et al. considered a manufacturer-retailer CSR remanufacturing supply chain (CSR) with differences in WTP for new and remanufactured products, and analyzed the optimal pricing, recycling, and profitability decisions in both centralized and decentralized models [17]. In dynamic pricing research for multiple products, scholars have considered the correlation between diverse products.Kayikci, Y et al. constructed a dynamic pricing model based on data-driven techniques that uses Internet of Things (IoT) sensor data to determine the optimal pricing strategy for perishable food products at the retail stage of the food supply chain [18]. Chen, X et al. explored the optimal pricing strategy for a two-stage supply chain with suppliers and retailers and introduced game theory to derive equilibrium for a single-stage pricing strategy and a two-stage pricing strategy [19]. Li, H et al. used ARIMA forecasting and dynamic pricing strategy optimization, an innovative approach to pricing and inventory decision making for retailing of perishable vegetable products in supermarkets in order to optimize product freshness and profitability [20]. Li, D and Wang, X proposed a dynamic pricing model for refrigerated food retail chain based on sensor data-driven, which aims to improve supply chain performance by utilizing sensor networks to provide real-time product quality information [21]. Yang, S et al. explored the joint decision-making problem of pricing strategy, shelf space allocation and replenishment strategy in single-item and multi-item perishable food supply chains by taking the retailer's total expected profit maximization as an objective [22].

Inventory control runs through the whole process of enterprise production and operation, including the control and management of ordering, warehousing, inventory information, storage and other aspects, in order to realize the purpose of controlling inventory according to the needs of the enterprise [23]. Inventory control is an important part of the daily operation of enterprises, so the research on inventory control is also a very important research hotspot [24]. Li, N and Wang, Z investigated the multi-cycle, two-tiered inventory control problem in omni-channel retailing, including online, offline and online pickup channels, and proposed specific approaches to address inventory control in stores and warehouses [25]. Gökbayrak, E and Kayış, E investigated the optimal inventory control strategy of a retailer in the face of product returns that are randomly dependent on the previous sale with the optimization objectives of maximizing expected profit and minimizing environmental impact [26]. Tao, S et al. proposed an integrated inventory management strategy for retail supply chain using Parallel Chicken Swarm Optimization (PCSO) algorithm to dynamically adjust inventory and optimize cost effectiveness [27]. Xu, G et al. examined the impact of BOPS (Buy Online, Pick Up Physical) model on inventory structure, pricing strategy and sales channel from the perspective of customer experience, and compared the optimal inventory control strategies under different sales models through numerical experiments [28].

Traditional inventory and pricing strategies need to make assumptions about market demand to simplify the problem, which is somewhat different from the complex and changing real environment [29]. Many scholars have applied artificial intelligence techniques to solve the revenue management problem, and proposed research on inventory control based on reinforcement learning methods. Sui, Z et al. proposed a reinforcement learning approach based on replenishment strategy in consignment inventory VMI system considering the lack of optimization model for replenishment strategy in Vendor Managed Inventory (VMI) system [30]. Mo, D. Y et al. proposed a non-smooth demand e-commerce inventory management method based on reinforcement learning theory, which aims to optimize cost and service level, and the proposed method achieves cost savings and higher service level [31]. Selukar, M et al. implemented inventory control of perishable goods using deep reinforcement learning techniques to reduce inventory costs and retail merchandise wastage rate considering real world factors such as delivery time, product life cycle and demand distribution [32]. Cuartas, C and Aguilar, J proposed a hybrid reinforcement learning and DDMRP based algorithm for determining the optimal time and quantity of products to be purchased and searched for the optimal inventory management strategy with three different reward functions [33]. Piao, M et al. proposed a multi-intelligence reinforcement learning approach and applied it to the civil aircraft manufacturing supply chain inventory management, achieving a 45% efficiency improvement [34].

Reinforcement learning theory applied to the field of dynamic pricing has also been the subject of considerable research. For example, Dogan et al. used reinforcement learning algorithms to analyze the retailer indefinite joint ordering and pricing problem considering multi-retailer competition [35]. Maestre, R et al. used reinforcement learning techniques to solve the dynamic pricing problem to maximize the revenue of the firm while maintaining fairness among different customer groups [36]. Chen, S et al. proposed a reinforcement learning approach using policy gradient and deep neural networks to dynamically price mobile edge computing services to maximize revenue for corporate profits [37]. Lu, R et al. proposed a dynamic pricing algorithm for energy management in a tiered electricity market based on a reinforcement learning algorithm that takes into account both the profit of the service provider and the cost of the customer [38]. The study by Wang, R et al. further used deep reinforcement learning to study the joint inventory and pricing problem focusing on more difficult perishables study and using neural networks to avoid dimensional catastrophe, and the results showed that the deep reinforcement learning model outperformed the traditional reinforcement learning model without the use of neural networks [39]. Most of the above studies use reinforcement learning algorithms to study the library dynamic pricing and inventory control problems, but reinforcement learning methods tend to perform poorly when the problems have very large state and action spaces and unknown state transfer probabilities.

In this paper, we study the joint inventory control and dynamic pricing problem in e-commerce retailing through deep reinforcement learning methods. A multi-product joint inventory control and dynamic pricing revenue model for e-commerce retail considering out-of-stock substitution and the corresponding Markov decision process are constructed, and the mathematical model is transformed into a quaternion element for reinforcement learning. Then the algorithms for solving the joint inventory control and dynamic pricing models are designed based on the DDPG framework by combining the ideas of empirical replay mechanism, independent objective function, stochastic noise, and soft update. Finally, comparative experiments are designed to illustrate the superiority of DDPG algorithms in solving the mentioned problems, and a series of simulation experiments are conducted to obtain meaningful research conclusions for managers and decision makers.

II. Method

II. A.Enhanced learning

Reinforcement learning has its roots in interactive learning processes in which an intelligent body performs actions based on the state of the environment in which it finds itself and receives appropriate rewards as a result [40]. The central goal of this learning mechanism is to maximize the accumulated rewards, and by learning through feedback on reward signals, the intelligent body optimizes its behavior to achieve the best adaptation to the environment.

II. A. 1) Markov properties

The Markov property originates from the field of probability theory and refers to the fact that when a stochastic process is given a present state and all past states, the conditional probability distribution of its future states depends only on the current state and is independent of the past states. This property specifies the dynamic system behavior through a specific probability distribution formula (1):

$$Pr\{r_{t+1}=r,s_{t+1}=s^{'}\mid s_0,a_0,r_1,\cdots,s_{t-1},a_{t-1},r_t,s_t,a_t\} \tag{1}$$

For all $r,s^{'}$ and all possible values of past events $s_0,a_0,r_1,\cdots,s_{t-1},a_{t-1},r_t,s_t,a_t,$ if the environment response at time t+1 is related only to the state and action at time t, independent of the previous history of states and actions, then the state signal can be said to possess the Markov property, in which case the behavioral dynamics of the environment can be represented by equation (2) for all $r,s^{'},s_i$ and a_t:

$$p(s',r\mid s,a)=Pr\{r_{t+1}=r,s_{t+1}=s^{'}\mid s_t,a_t\} \tag{2}$$

When the full probability distribution of a stochastic process corresponds directly to the dynamical equations of the environment, the state of the process is then recognized as possessing Markovian properties, and in this framework, the environment and its associated tasks as a whole are also considered to satisfy Markovian properties.

II. A. 2) Markov decision-making process

Reinforcement learning tasks are usually represented as decision processes possessing Markovian properties, where a finite Markovian decision process is formulated by defining the immediate dynamics of the state space, the action space, and the environment, and the probability of the subsequent state $s^{'}$ and the corresponding reward r can be expressed by the following equation for any given state s and action a, and these parameters together determine the dynamic properties of the process.

$$p(s^{'},r\mid s,a)=\Pr\{s_{t+1}=s^{'},r_{t+1}=r\mid s_t=s,a_t=a\} \tag{3}$$

The expected reward corresponding to the state-action:

$$r(s,a)=E[r_{t+1}\mid s_t=s,a_t=a]=\sum_{r\in R}r\sum_{s^{'}\in S}p(s^{'},r\mid s,a) \tag{4}$$

State transition probabilities:

$$p(s^{'}\mid s,a)=\Pr\{s_{t+1}=s^{'}\mid s_t=s,a_t=a\}=\sum_{r\in R}p(s^{'},r\mid s,a) \tag{5}$$

The expected reward for the next state:

$$r(s,a,s^{'})=E\left[r_{t+1}\mid s_t=s,a_t=a,s_{t+1}=s^{'}\right]=\frac{\sum_{r\in R}rp(s^{'},r\mid s,a)}{p(s^{'}\mid s,a)} \tag{6}$$

Markov decision tasks for reinforcement learning typically cover:

1) A set of executable actions A.

2) A set of possible states S, and a distribution of starting states $p(s_0)$.

3) A timely feedback reward function $r(s_t,a_t,s_{t+1})$.

4) Transition function $T(s_{t+1}\mid s_t,a_t)$, after executing an action, the environment will be transferred to another state according to some probabilities, and these probabilities constitute the state transfer function.

5) Discount factor $\gamma \in [0,1]$, higher values of γ indicate a greater emphasis on future payoffs, and smaller values indicate a preference for immediate payoffs.

In each training cycle, the output of the strategy π consists of the state in which it is located, the actions taken, and the rewards it generates. The $\pi : S \to p(A = a \mid S)$ represents a mapping from the state to a probability distribution that accumulates rewards from the environment each time an action is taken according to the policy, returning the result $R = \sum_{t=0}^{T-1} \gamma^t r_{t+1}$. The goal of reinforcement learning is to find an optimal policy π^* that maximizes the expected total reward across all states through continuous training:

$$\pi^* = \arg\max_{\pi} E[R \mid \pi] \tag{7}$$

II. A. 3) Basic framework for enhanced learning

The process of reinforcement learning can be viewed as a cycle of exploration and evaluation, which involves the core components of the environment state, actions, rewards, and intelligences. In the process of reinforcement learning, an intelligent body selects and executes an action a in state s, and the environment changes to state s' after accepting the action, and at the same time feeds back a reward signal r to the intelligent body, which selects the next action based on this reward signal. The main goal of the feedback of state and reward is to enable the intelligent body to obtain the maximum cumulative reward value, so as to realize the optimal response to the external environment. The basic framework of reinforcement learning is shown in Figure 1.

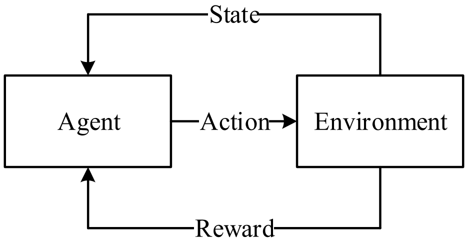


Figure 1: Basic framework of reinforcement learning

II. B.Dynamic Pricing and Inventory Control Model for E-commerce Retailing

II. B. 1) Mathematical modeling

The following assumptions are introduced in this chapter:

(1) In order to simplify the problem, the study in this chapter considers only one series of products, the product set is $Products = \{1, 2, ..., M\}$, and there is a shortage substitution relationship between both products m and products k in the product set, $m, k \in Products, m \neq k$.

(2) Assuming that Product m is out of stock in a certain period and consumers seek its substitution, when some consumers turn to Product k and Product k is facing a shortage of inventory at the same time, the part of consumers will not make any further consumption and give up purchasing substitutes of Product m.

For the research problem in this chapter, the total profit of the platform consists of two components: the total revenue obtained by multiplying the sales of each finished product with the corresponding price, and the total cost of each product. Considering the substitutability of products, part of the sales of a product is the sales of the product itself, and the other part is the sales resulting from the switching of customers of its substitute product to that product. Whereas, the ordering cost, shortage cost, inventory cost and fixed cost of a product constitute the total cost of that product.

The mathematical notation is defined as shown in Table 1, considering that there are multiple products, the total sales of the platform is the sum of the sales of the multiple products:

$$\sum_{t=0}^{T} Sales = \sum_{t=0}^{T} \sum_{m=1}^{M} p_m^t \times n_m^t \tag{8}$$

Where the price of each product varies from period to period, the current price of product m, p_m^t, is given by the base price and the current discount, i.e., $p_m^t = p_{base} \times p_{discount}$. And the sales volume of each product is determined by the current product demand and available inventory.

Also, considering the substitution relationship between products, the current demand for product m is the sum of the initial demand for that product and the substitute demand for product m that customers turn to when other products are out of stock as they seek their substitutes.

$$d_m^t = w_m^t + \sum_{k=1,k\neq m}^{M} h_{km}^t \tag{9}$$

Similar to the previous chapter, the total cost of the platform is divided into ordering costs, inventory costs, out-of-stock costs and fixed costs for each product:

$$\sum_{t=0}^{T} Costs = \sum_{t=0}^{T}\sum_{m=1}^{M}(c_o^m \times q_m^t + c_h^m \times (OI_m^t - n_m^t)^+ + c_p^m \times (OI_m^t - n_m^t)^- + c_f^m) \tag{10}$$

The merchandise inventory control and pricing model for e-commerce retailing consists of Equation (8) and Equation (10) as shown below:

$$\begin{aligned}\max \sum_{t=0}^{T} r^t &= \sum_{t=0}^{T} Sales - \mathrm{Cos}\,ts \\ &= \sum_{t=0}^{T}\sum_{m=1}^{M}(p_m^t \times n_m^t - c_o^m \times q_m^t - c_h^m \times (OI_m^t - n_m^t)^+ - c_p^m \times (OI_m^t - n_m^t)^- - c_f^m)\end{aligned} \tag{11}$$

$$s.t. OI_m^t = \sum_{i=L}^{l-1} s_i^t \tag{12}$$

$$d_m^t = w_m^t + \sum_{k=1,k\neq m}^{M} h_{km}^t \tag{13}$$

$$n_m^t = \min(OI_m^t, d_m^t) \tag{14}$$

Table 1: Model mathematical symbol meaning

Symbol	Meaning
r^t	Total revenue of t
p_m^t	The price of the t phase product m
n_m^t	The t period is the sales of product m after pricing p_m^t
d_m^t	The customer reaches the product m after the t period
c_o^m	The unit order cost of the product m
c_h^m	The unit inventory cost of the product m
c_p^m	The unit of the product is out of goods
c_f^m	Fixed cost of product m
OI_m^t	The available inventory of the t phase product m
$s_{m,i}^t$	The number of stocks of m in t life
w_m^t	Customer's initial demand for product m
h_{km}^t	The replacement quantity of the product m to the production of k when the product k is out

II. B. 2) Markov decision process design

This section will construct a reinforcement learning quaternion by defining a Markov decision process as the basis for applying reinforcement learning algorithms to solve the research problems in this chapter. This chapter defines the Markov decision process as follows:

(1) State space: the state variable in the t th period represents the number of products in stock corresponding to each shelf-life of product m, which is represented by the $M \times (l-1)$ dimensional vector matrix $s^t = [s_1^t, s_2^t, \ldots, s_i^t, \ldots, s_M^t]$. where each item is a $(l-1)$ -dimensional vector $S^t = [s_0^t, s_1^t, \cdots, s_k^t, \cdots, s_{l-1}^t]$, denoting the inventory information of product m, and s_k^t denoting the remaining validity of product in the t th period for $l-k$ of the product inventory quantity.

(2) Action space: similar to the previous chapter, the decision variables in period t are defined as $a^t = (q_m^t, p_m^t)$, q_m^t is the order quantity of product m, and p_m^t is the price discount of product m. The order quantity also follows the $d+x$ rule, where the retailer observes the demand for product m in period $t-1$, d_m^{t-1}, and decides in period t that the order quantity for product m is $q_m^t = d_m^{t-1} + x_m^t$; and for the price discount there is $p_m^t \in [p_{\min}, p_{\max}]$.

(3) State transfer: the deterioration rate $\theta_m(t)$ represents the deterioration characteristics of product m in period t. The model does not consider the case of order stacking, when the consumer demand for a product m cannot be satisfied, the customer will move to an alternative product or give up the purchase, i.e., the order for product m will disappear. For product m, when $i < L$ there is $s_i^t = s_{i-1}^{t-1}$, and when $L \le i < l$ there is $s_i^t = \theta_m(t)(s_{i-1}^{t-1} - (n_m^t - \sum_{j=i}^{l-1} s_j^t)^+)^+$.

(4) Reward function: in a certain state s^t, the intelligent body makes a decision with the strategy π to decide the order quantity and price discount of all products in the current period, from which it gets the corresponding reward r^t.

II. C.DDPG-based inventory control and dynamic pricing algorithm design

II. C. 1) Strategic Gradient Algorithm

The strategy gradient algorithm maximizes the cumulative reward by optimizing the strategy parameters by directly ramping up or down the gradient, and by virtue of this direct learning of the strategy in continuous or discrete action space has led to good results in tasks in continuous action space [41]. The cleverness of this algorithm is to adjust the probability of the actions that can be selected individually based on the feedback of the rewards, with actions considered good having an increased probability of being selected and actions considered bad having a lower probability of being selected.

In the strategy gradient approach. The policy is usually represented as a parameterized probability distribution $\pi_\theta(a \mid s)$, which means the probability of taking an action a given the state s, but the drawn action is not necessarily the most probable one, but rather one that obeys the $\pi_\theta(a \mid s)$ policy probability density with stochasticity. where denotes θ the strategy parameters, which can be the weights of the neural network. The strategy can be either deterministic or stochastic, but in strategy gradient methods, stochastic strategies are usually used to facilitate exploration.

The goal of a strategy gradient algorithm is to maximize the expected reward, which can be achieved by optimizing the objective function $J(\theta)$, which is defined as the expected value of the cumulative reward under the strategy $\pi\theta$:

$$J(\theta) = \mathrm{E}_{s_0}[V^{\pi_\theta}(s_0)] \tag{15}$$

where $R(\tau)$ is the total reward of a trajectory τ (consisting of a series of state and action pairs). The objective function reflects the performance of a given strategy, and the strategy gradient method finds the strategy that maximizes $J(\theta)$ by adjusting θ.

The strategy gradient theorem provides a way to compute the gradient of the objective function, i.e., how to adjust the strategy to increase the total reward based on changes in the strategy parameters. According to the strategy gradient theorem, the gradient of the objective function $J(\theta)$ with respect to the parameter θ is:

$$\begin{aligned} \nabla_\theta J(\theta) &\propto \sum_{s\in S} v^{\pi_\theta}(s) \sum_{a\in A} Q^{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(a \mid s) \\ &= \sum_{s\in S} v^{\pi_\theta}(s) \sum_{s\in S} \pi_\theta(a \mid s) Q^{\pi_\theta}(s,a) \frac{\nabla_\theta \pi_\theta(a \mid s)}{\pi_\theta(a \mid s)} \\ &= E_{\pi_\theta}\left[Q^{\pi_\theta}(s,a) \nabla_\theta \log g \pi_\theta(a \mid s) \right] \end{aligned} \tag{16}$$

The $\nabla_\theta J(\theta)$ in this equation denotes the gradient of the objective function $J(\theta)$ with respect to the strategy parameter θ, i.e., the optimization direction to be found. $\sum_{s\in S} v^{\pi_\theta}(s)$ denote the summation over all states s and the probability of visiting the state s under the strategy π_θ, respectively. $\Sigma_a \in A$ are denoted as the summation over all actions a, respectively, and $Q^{\pi_\theta}(s,a)$ denote the value function of an action when it is taken in the state s under action a and following the strategy π_θ. $\nabla_\theta \pi_\theta(a \mid s)$ denotes the gradient of the strategy π_θ with respect to the parameter θ, i.e., how to adjust the parameter θ in order to increase the probability of choosing the action a under the state s.

This formula is the core of the strategy gradient approach and shows how the parameter θ can be tuned by estimating the gradient of the strategy as a way to maximize the cumulative reward. It shows that optimal strategies can be efficiently learned by tuning the strategy parameters in a direction that maximizes the expected reward.

II. C. 2) Actor-Critic Algorithm

The Actor-Critic algorithm is a reinforcement learning method that combines the advantages of the value function approach and the policy gradient approach for solving problems in continuous action spaces [42]. This algorithm consists of two parts: the Actor and the Critic. The Actor is responsible for learning the strategy, i.e., the action that should be taken in a given state; the Critic evaluates the current strategy, i.e., it calculates the value of the state or action. With this structure, the Actor-Critic algorithm aims to learn a parameterized strategy directly, while reducing the variance of the strategy evaluation and improving the learning efficiency.

The specific process of the Actor-Critic algorithm is as follows:

(1) Initialize the strategy network parameters, the value network parameters.

(2) For sequence do.

(3) Sample the trajectory $\{s_1, a_1, r_1, s_2, a_2, r_2, r_2, ...\}$.

(4) Calculate for each data step: $\delta_t = r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t)$.

(5) Update the value parameter $w + \alpha_\omega \sum_t \delta_t \nabla_\omega V_\omega(s_t)$.

(6) Update the strategy parameters $\theta + \alpha_\theta \sum_t \delta_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$.

(7) end for.

The core of the Actor-Critic algorithm is that it combines policy parameterization and value function estimation to implement a mechanism that balances exploration and exploitation. Actor updates the policy based on the feedback of the value function provided by Critic, and Critic updates the estimate of the value function based on the behavior of Actor.

II. C. 3) Deep deterministic strategy gradient

The Deep Deterministic Policy Gradient (DDPG) algorithm is a reinforcement learning algorithm that combines the Actor-Critic framework and deep learning techniques, and is especially designed for solving problems in the space of successive actions. DDPG guides the updating of a policy by learning a deterministic policy (i.e., the optimal action in a given state) and a value function to maximize the cumulative reward. The DDPG algorithm is proposed based on the ideas of Actor-Critic method and DQN algorithm, aiming to deal with high-dimensional problems in continuous action space while maintaining the stability and efficiency of the learning process.

Where the deterministic policy gradient theorem is formulated as:

$$\nabla_\theta J(\pi_\theta) = \mathrm{E}_{s\sim v}^{\pi_\beta}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q_\omega^\mu(s,a) r \mid_{a=\mu_\theta(s)}\right] \quad (17)$$

$\nabla_\theta J(\pi_\theta)$ denotes the gradient of the objective function J with respect to the strategy parameter θ. This gradient indicates the direction in which to adjust θ to increase the expected value of the long-term cumulative reward of the strategy π_θ. The $\mathrm{E}_{s\sim v^{\pi_\beta}}$ denotes the expectation operation, where the expectation is with respect to a distribution of states s that are based on another strategy π_β (usually in practice, where π_β and π_θ are the same, i.e., the strategy π_θ itself) is sampled. The $\nabla_\theta \mu_\theta s$) denote the gradient of the deterministic strategy $\mu_\theta(s)$ with respect to its parameter θ. A deterministic strategy means that for each state s, the strategy produces a deterministic action a. $\nabla_\theta \mu_\theta(s) \nabla_a Q_\omega^\mu(s,a)\mid_{a=\mu_\theta(s)}$ denotes the action value function $Q_\omega^\mu(s,a)$ gradient with respect to action a at the point where action a is equal to the value of the action generated by policy μ_θ in state s. Here, $Q_\omega^\mu(s,a)$ is usually parameterized by another neural network (called the Critic network) with parameters ω.

II. C. 4) E-commerce retail joint pricing and inventory control algorithm design

Now for single-product studies, the inventory control and dynamic pricing problem for multiple products has a particularly large decision space and state space, and solving it with a tabular reinforcement learning method will lead to a huge and hard-to-maintain Q-value table, whereas a reinforcement learning method based on the estimation of the value function needs to maximize the Q-value of the problem by $a_t = arg\max_a q(S_t, a)$ to find the action that maximizes the value of Q. It is more applicable to finite discrete action spaces, and solving high-dimensional decision problems will lead to a drastic increase in the number of neurons in the output layer, and thus is difficult to be applied to larger decision spaces and state spaces. Based on this, this chapter solves the inventory control and dynamic pricing problems of e-commerce retail goods through a deep deterministic policy gradient approach applicable to larger state spaces and continuous decision spaces.

According to the strategy gradient theorem, the algorithm is solved by iterating the change of strategy parameters along the gradient of revenue to strategy as a way to maximize the expected revenue. The parameter θ is used in the above equation to define the deterministic strategy with parameters, and in this paper, we mainly use the neural network for fitting, and here θ is expressed as the weight of the neural network.

Specifically, the study uses two evaluation networks, a Critic evaluation network, which is responsible for updating the parameters w of the evaluation network iteratively in order to accurately evaluate the corresponding Q-value $q(s,a,w)$ and the target Q-value $q^{target} = r_t + \gamma q^{w'}(s_{t+1}, \pi_\theta(s_{t+1}))$; and the other is the Critic target network, i.e., the target evaluation network, under which $q^{w'}(s_{t+1}, \pi_{theta}(s_{t+1}))$ part of the computation and update the parameter w' according to the parameter w. The update is soft: $w' \leftarrow \tau w + (1-\tau)w'$. The loss function of the Critic evaluation network is similar to that of the previous chapter, and takes the form of a mean-square error, defined as:

$$J(w) = \frac{1}{m}\sum_{j=1}^{m}(q^{target} - q(s,a,w))^2 \tag{18}$$

According to the loss function in the above equation, the gradient on the Critic network for parameter w is:

$$\frac{\partial J(w)}{\partial w} = E\left[(q^{t\arg et} - q(s,a,w))\frac{\partial q(s,a,w)}{\partial w}\right] \tag{19}$$

III. Results and Discussion

III. A. Experimental design

For the dynamic pricing and inventory control problem in this paper, the following numerical experiments are designed to obtain the simulation results in order to test the performance of the proposed DDPG algorithm.

Various hyper-parameter settings are tested according to the randomized grid search method and the hyper-parameter values with good performance are recorded, all the hyper-parameter values are shown in Table 2, once all the hyper-parameters are selected, the two intelligences will start learning.

Table 2: The basic parameter values for numerical experiments

Parameter symbol	value	Parametric interpretation
M	6000	Game rounds
T	200	Decision cycle length
ε_0	1	Initial standard deviation
ε_{decay}	0.99998	Attenuation rate
S	9	Batch size
lr_{actor}	0.0002	The learning rate of actors
lr_{critic}	0.002	The learning rate of the critic
v	0.002	Target network weight
Θ	5000	Experience replay pool size

III. B. Learning Objectives and Approximate Optimal Strategies

Convergence ensures that the decision making subject is able to select a stable strategy matching the adversary and the average return value will plateau at each round, while rationality ensures that the decision maker selects the optimal response strategy of its adversary provided that the adversary's strategy is stable. The algorithm is executed for 5000 rounds with the parameter settings of the previous section and its average round return value is recorded as shown in Figure 2. The figure shows that after about 1300 rounds, the average return values of the retailer and the strategic consumer reach a steady state, which indicates that the strategies of the two decision-makers are essentially unchanged and the convergence of the environment has been achieved. Due to the unique payoff function design mechanism in this chapter, once the average payoff value of the strategic consumer converges to 0, it can be indicated that it realizes the optimal response strategy; for the retailer, assuming that the strategic consumer has learned the optimal response strategy $(v_{lt} = v_{lt}^*, v_{2t} = v_{2t}^*)$, and the stochastic terms occurring in the cycle are obtained in advance, then the average payoff value of the retailer and strategic consumer will reach steady state after about 1,300 rounds. of the random term is obtained in advance, then the original problem can be transformed into a nonlinear programming with equation constraints, which can be solved with the help of optimization algorithms such as the Lagrange multiplier method or commercial solvers

The figure shows that in the last 1000 rounds of the game, the average round gain reaches 0.197 for the retailer and 0.035 for the strategic consumer.Thus, to some extent, both the retailer and the strategic consumer can learn the best response strategy and thus reach the rationality objective.

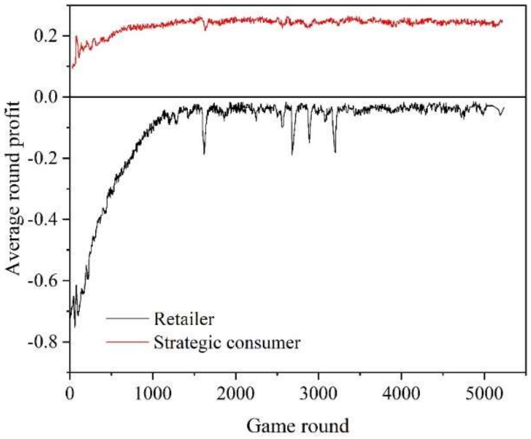


Figure 2: Average profits of the retailer and strategic consumers

In order to obtain the approximate optimal strategies, this paper saves and records the values of the network parameters in the smooth state, and then calculates the approximate optimal strategies of the retailer and the strategic consumers with different reference prices down, and the results are shown in Table 3 (q: order quantity, p_1: price in the full-price period, p_2: discounted price in the price-reduction period, v_1 :Minimum willingness to pay in the first stage, v_2 :Minimum willingness to pay in the second stage, v_1^* :Minimum willingness to pay in the optimal first stage, v_2^*:Minimum willingness to pay in the optimal second stage) A lot of decision making conclusions can be obtained from the table.

Analyze the relationship between reference price and consumer behavior. Negative reference price effect occurs when the consumer's reference price is located between 0 and 0.5, and as the reference price increases, v_1 gradually increases and moves away from p_1, while v_2 starts from p_1, and moves gradually closer to p_2. A positive reference price effect occurs when the reference price level lies between 0.8 and 1. As the reference price increases, v_1 gradually moves closer to 1 and v_2 gradually moves closer to 0. When the reference price level lies around 0.6 and 0.7, no reference price effect occurs, v_2 equals p_2, which leads to the conjecture that the reference price of the steady state should stay at the level of 0.6 to 0.7.

Table 3: Near-optimal policies of the retailer and strategic consumers with reference prices

Reference price	q	p_1	p_2	v_1	v_2	v_1^*	v_2^*	Retailer profit
0	0.3706	0.6508	0.6492	0.6449	0.6348	0.6508	0.6508	0.1964
0.1	0.3766	0.6452	0.6443	0.6445	0.6333	0.6452	0.6452	0.1969
0.2	0.3795	0.6406	0.6395	0.6455	0.6341	0.6406	0.6406	0.1967
0.3	0.3818	0.6405	0.6392	0.6445	0.6332	0.6405	0.6405	0.1973
0.4	0.3798	0.6420	0.6405	0.6471	0.6347	0.6420	0.6420	0.1973
0.5	0.3805	0.6425	0.6411	0.6471	0.6341	0.6435	0.6424	0.1982
0.6	0.3789	0.6426	0.6409	0.6437	0.6303	0.6580	0.6409	0.1993
0.7	0.3740	0.6355	0.6341	0.6620	0.6408	0.6630	0.6325	0.1941
0.8	0.3856	0.6421	0.6408	0.6651	0.6125	0.9178	0.6095	0.2016
0.9	0.4010	0.6427	0.6418	0.6895	0.5834	1	0.5610	0.2092
1	0.4076	0.6397	0.6384	0.7128	0.5398	1	0.5059	0.2159

The experiment is redesigned to test the reference price level during the smoothing period, and the final results are obtained under the condition of different initial reference prices, as shown in Fig. 3. The results show that given different initial reference prices, the final system state basically stays at 0.6324, indicating that the previous guess is accurate.

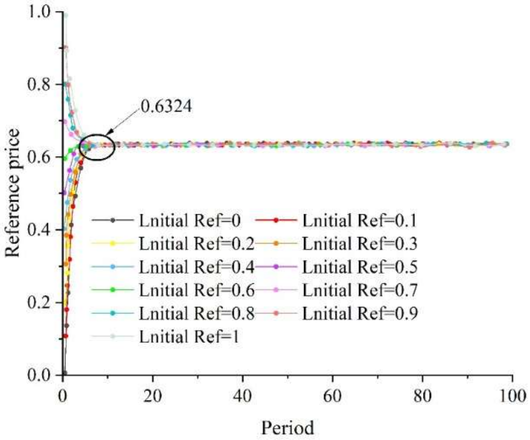


Figure 3: The convergence of reference prices over the whole period

III. C. Dynamic Pricing and Inventory Management Analysis

The assumption that e-commerce retailing has the same selling price in the same period is removed in this paper in this section of the dynamic inventory pricing management problem, which allows a richer selection space for the actions of the intelligent body. The problem is trained by the Near DDPG algorithm and the Action Advantage Review algorithm, respectively, and the reward curves of the two algorithms training are shown in Fig. 4 and Fig. 5. As can be seen from Fig. 4, when the proximal decision optimization algorithm is used, the fluctuation of the reward decreases when the training reaches about 20,000 episodes, and the training reward no longer has an obvious upward trend, and the reward curve begins to flatten out. As can be seen from Figure 5, when the dominant action review algorithm is used, the fluctuation amplitude of the reward decreases when the training reaches about 21,000 episodes, the training reward no longer has an obvious upward trend, and the reward curve begins to stabilize.

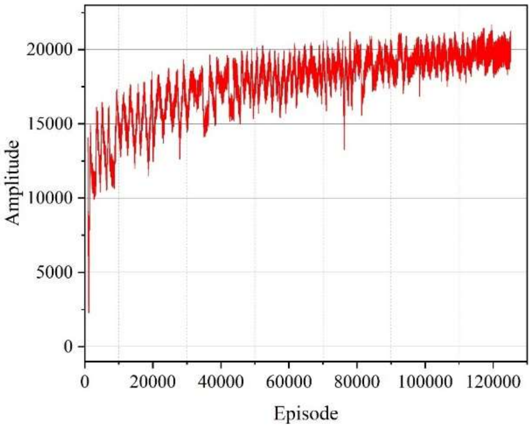


Figure 4: The reward curve of the DDPG algorithm in dynamic inventory pricing

Combining the two reward curve graphs, it can be seen that for the dynamic inventory pricing management problem, both the DDPG algorithm and the PAQ-A2C algorithm can obtain positive gains and the reward curve gradually fluctuates from a large fluctuation and grows rapidly and tends to be stable with the training process, due to the fact that even if the reward curve tends to be stable during the training process of the two reinforcement learning algorithms, there still exists a certain magnitude of fluctuation, so in this figure can only be Roughly judged by the coordinates of the DDPG algorithm compared to the PAQ-A2C algorithm can ultimately obtain more benefits, but still need to use the average reward curve to make further judgments.

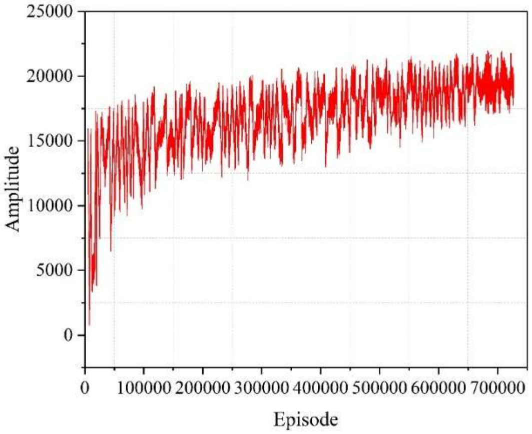
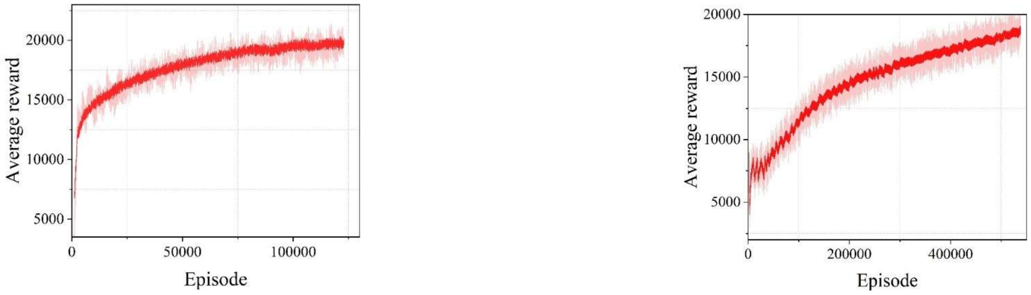


Figure 5: The reward curve of the PAQ-A2C algorithm at the dynamic inventory pricing

The average reward curves of the two algorithms trained are shown in Fig. 6 (with a sliding average of rewards, except for the first 99 times, the number of rewards averaged in each sliding average is 100):

Combining the two average reward curves, it can be seen that the final average reward of the DDPG algorithm is higher than that of the PAQ-A2C algorithm. In this paper, we give the average rewards of the last 100 episodes in the training data of the two algorithms, the average reward of the DDPG algorithm is: 20218, while the average reward of the PAQ-A2C algorithm is 19216, compared with the PAQ-A2C algorithm, the average reward of the DDPG algorithm is improved by 8.42%, which can show that the DDPG algorithm is better than the PAQ-A2C algorithm in both cases.



(a) DDPG algorithm

(b) PAQ-A2C algorithm

Figure 6: Average reward for the price uniform algorithm

The changes in selling prices over time for the five e-commerce retailers are shown in Figure 7, where it can be observed that for the different retailers, there is a clear difference in the interval of price changes over time for each retailer due to their varying demand functions, but at the same time, for each retailer there is a relatively more stable price for most of the time.

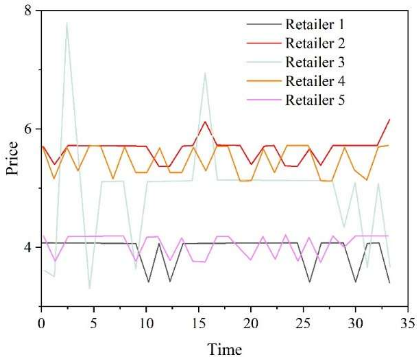


Figure 7: The change in the sales price of the e-commerce retail

Figure 5 Number of items obtained by e-commerce retailers ordering from suppliers is shown in Figure 8, and again it can be seen that the number of items obtained by each retailer varies over time.

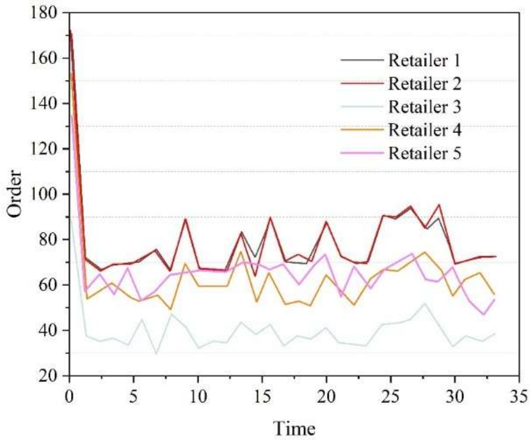


Figure 8: The number of goods that e-commerce retail stores are changing at any time

Meanwhile, it can be noticed from the two graphs that the pricing and ordering volume curves of Retailer 1 and Retailer 2 are extremely similar and overlap in many places, and combining the parameters of the demand functions of both of them, it can be found that the high similarity between the two stores in terms of their demand functions is a possible reason for this phenomenon.

III. D. Numerical experiments under the loss principle

III. D. 1) Dynamic Pricing Experiments with Positive Delivery Periods

The results of the comparison of dynamic pricing under positive lead time are shown in Table 4, which shows that it is preferable to dynamically adjust the price under the loss principle so that the price can be adjusted according to the availability of inventory and the remaining life of the product to maximize the profit.

Table 4: The results of the dynamic pricing of the orthogonal cargo period are compared

Algorithm	Quality time	MEP	MEP_{FP}	MDC	MDC_{FP}
DDPG	2	2838.187	2724.926	448.287	448.025
	3	4909.047	4701.827	128.057	150.027
	4	4994.85	4864.539	105.627	119.555
PAQ-A2C	2	2816.802	2569.242	407.529	406.429
	3	4724.207	4567.19	170.896	179.508
	4	4883.969	4715.392	161.767	161.159

The results on average profit as well as average processing cost under different algorithms are shown in Table 5. It can be seen that the proposed DDPG algorithm has better performance than PAQ-A2 algorithm in case of positive lead time and most importantly the proposed DDPG algorithm is significantly better than Q-learning algorithm. The reason for this is also explained under the backlog principle, in short traditional table based reinforcement learning methods are difficult to achieve effective learning as well as application in today's context of large data dimensions and complex environments.

Table 5: The average profit and the average cost result of different algorithms

Algorithm	Quality time	Delivery date	MEP	MDC
DDPG	2	0	5451.625	78.409
		1	12818.187	428.287
PAQ-A2C	2	0	5423.917	73.029
		1	2796.802	387.529
Q-learning	2	0	4494.11	144.801
		1	<0	0

III. D. 2) DDPG Algorithm Performance Experiments

The average revenue MEP performance of the two reinforcement learning algorithms DDPG and PAQ-A2C and the optimal policy after 20,000 simulations is shown in Table 6. From the table, it can be seen that the proposed algorithms achieve good performance for three different shelf times, and the difference between the average revenue performance of the proposed algorithms and the average optimal profit is almost always less than the maximum possible profit per unit.

Table 6: The average yield of the two algorithms and the optimal strategy is MEP

Method	l	MEP	MEP_{OP}	MEP_{AVE}	$\frac{MEP}{MEP_{OP}}*100\%$
DDPG	2	5406.625	5727.845	12.597	94.392
	3	5631.172	5749.987	5.85	97.935
	4	5491.958	5742.401	10.238	95.639
PAQ-A2C	2	5423.917	5752.001	12.826	94.296
	3	5684.285	5743.227	3.854	98.974
	4	5561.377	5751.292	8.22	96.698

The variation of the difference between the algorithm and the optimal upper bound as a function of the number of simulations for three different shelf-life times is shown in Figures 9, 10, and 11. To better show the speed of convergence, the graphs are plotted in log-log form. From the three plots, it can be seen that the average profit MEP starts to decrease rapidly after ninety simulations, which indicates that the proposed deep deterministic policy gradient algorithm learns effective ordering and pricing strategies. With a better understanding of the changes in the environment, the agent starts to know what actions to make in what state will maximize the reward. In this paper, the speed of convergence is also fitted by taking the fitted curves, and the following equations are fitted for shelf life times 2, 3, and 4, respectively.

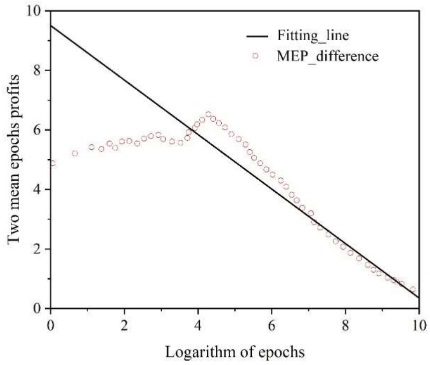


Figure 9: Average profit MEP difference in period 2

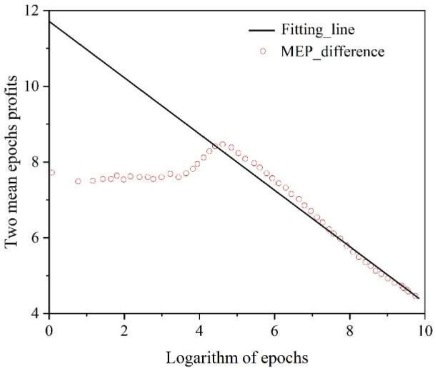


Figure 10: Average profit MEP difference in three periods

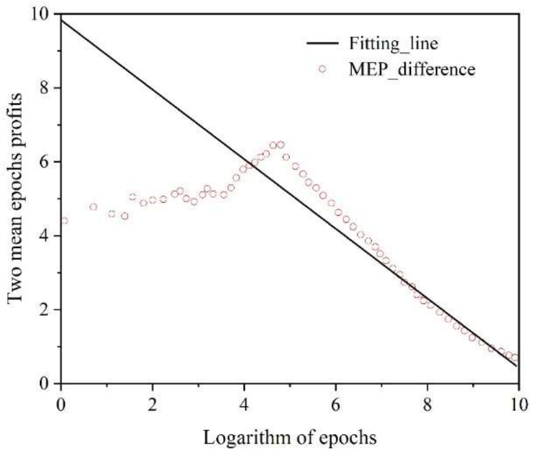


Figure 11: Average profit mep difference in four hours

III. D. 3) Study of strategies with and without fixed ordering costs

For the whereas case without fixed ordering costs, this part of the experiment is the same as under the same setup as under the backlogging principle, and this paper obtains the same results, i.e., learned order quantities do not increase for both in-transit and existing in-stock inventory levels. When lead time L = 0, the learned price strategy is always equal to the price at which the highest expected profit can be achieved; when lead time L > 0, the learned price is most sensitive to the oldest inventory. For the case with fixed ordering costs, we obtain a value for the critical inventory level similar to the one under the backlogging principle, and we have done related experiments for the critical value as well, and the results are consistent in that the learned convergent strategies with different lead times can achieve near-optimal standard performance, which is not shown here in order not to duplicate the presentation. The average profit performance of the algorithm with different fixed order costs and cost ratios under the shelf life of 4 and delivery dates of 0 and 1 is compared with the optimal criterion as shown in Table 7, from which it can also be seen that the proposed algorithm has good performance.

Table 7: The average yield of the algorithm under different parameters is MEP

Quality time	Delivery date	K	$\frac{u}{h+u}$	MEP	MEP_{Ben}	$\frac{MEP}{MEP_{Ben}}*100\%$
4	0	50	97%	15443.275	16192.417	95.374
	1	50	97%	14549.43	15642.24	93.014
4	0	25	97%	15819.19	16573.04	95.451
	1	25	97%	14909.926	15688.029	95.040

IV. Conclusion

In this paper, we construct a multi-product joint inventory control and dynamic pricing revenue model for e-commerce retail considering out-of-stock substitution, and design a solution algorithm based on DDPG algorithm for multi-product joint inventory control and dynamic pricing model according to the model. In the framework of this algorithm, the approximate optimal decision and revenue functions of retailers and strategic consumers are obtained, and the revenue reaches 0.197 and 0.035 to achieve a better revenue effect, respectively. Secondly, when the training reward reaches stability, the average gain obtained by the DDPG algorithm is larger compared to the simplified scenario. Finally the importance of dynamic pricing for inventory control of e-commerce retail goods is experimentally verified.

Funding

Project of Undergraduate University Teaching Quality and Teaching Reform Improvement in Guangdong Province in 2023: Research on the Reform Path of Logistics Management Course Based on the OBE Educational Concept in the Context of Special Creation Integration (Yuejiao Gao Han [2023]9-1200).

References

[1] Wang, Y., & Chang, J. (2021, April). Future development trend of “new retail” and e-commerce based on big data. In Journal of physics: Conference series (Vol. 1852, No. 3, p. 032029). IOP Publishing.

[2] Kumar, V., & Ayodeji, O. G. (2021). E-retail factors for customer activation and retention: An empirical study from Indian e-commerce customers. Journal of Retailing and Consumer Services, 59, 102399.

[3] Kleisiari, C., Duquenne, M. N., & Vlontzos, G. (2021). E-commerce in the retail chain store market: An alternative or a main trend?. Sustainability, 13(8), 4392.

[4] Risberg, A. (2023). A systematic literature review on e-commerce logistics: towards an e-commerce and omni-channel decision framework. The International Review of Retail, Distribution and Consumer Research, 33(1), 67-91.

[5] Chava, S., Oettl, A., Singh, M., & Zeng, L. (2024). Creative destruction? Impact of e-commerce on the retail sector. Management Science, 70(4), 2168-2187.

[6] 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.

[7] Tolstoy, D., Nordman, E. R., Hånell, S. M., & Özbek, N. (2021). The development of international e-commerce in retail SMEs: An effectuation perspective. Journal of World Business, 56(3), 101165.

[8] Wang, T. (2023). Research on the Impact of E-commerce on Offline Retail Industry. Frontiers in Business, Economics and Management, 10(1), 169-173.

[9] Weidinger, F., Boysen, N., & Schneider, M. (2019). Picker routing in the mixed-shelves warehouses of e-commerce retailers. European Journal of Operational Research, 274(2), 501-515.

[10] Chung, J. (2019). Effective pricing of perishables for a more sustainable retail food market. Sustainability, 11(17), 4762.

[11] Liu, K., Qiu, X., Chen, W., Chen, X., & Zheng, Z. (2019). Optimal pricing mechanism for data market in blockchain-enhanced Internet of Things. IEEE Internet of Things Journal, 6(6), 9748-9761.

[12] Kung, L. C., & Zhong, G. Y. (2017). The optimal pricing strategy for two-sided platform delivery in the sharing economy. Transportation Research Part E: Logistics and Transportation Review, 101, 1-12.

[13] Chatterjee, P., & Kumar, A. (2017). Consumer willingness to pay across retail channels. Journal of Retailing and Consumer Services, 34, 264-270.

[14] Szakály, Z., Kovács, S., Pető, K., Huszka, P., & Kiss, M. (2019). A modified model of the willingness to pay for functional foods. Appetite, 138, 94-101.

[15] Carson, R. T., & Czajkowski, M. (2019). A new baseline model for estimating willingness to pay from discrete choice models. Journal of Environmental Economics and Management, 95, 57-61.

[16] Zhao, Y., Ji, G., Jiang, Y., & Dai, X. (2020). Strategic customer behavior and pricing strategy based on the horizontal differentiation of products. Mathematical Problems in Engineering, 2020(1), 9475150.

[17] Dai, L., Shu, T., Chen, S., Wang, S., & Lai, K. K. (2020). CSR remanufacturing supply chains under WTP differentiation. Sustainability, 12(6), 2197.

[18] Kayikci, Y., Demir, S., Mangla, S. K., Subramanian, N., & Koc, B. (2022). Data-driven optimal dynamic pricing strategy for reducing perishable food waste at retailers. Journal of cleaner production, 344, 131068.

[19] Chen, X., Wu, S., Wang, X., & Li, D. (2019). Optimal pricing strategy for the perishable food supply chain. International Journal of Production Research, 57(9), 2755-2768.

[20] Li, H., Liu, J., Qiu, J., Zhou, Y., Zhang, X., Wang, Y., & Guo, W. (2024). ARIMA-driven vegetable pricing and restocking strategy for dual optimization of freshness and profitability in supermarket perishables. Sustainability, 16(10), 4071.

[21] Li, D., & Wang, X. (2017). Dynamic supply chain decisions based on networked sensor data: an application in the chilled food retail chain. International Journal of Production Research, 55(17), 5127-5141.

[22] Yang, S., Xiao, Y., & Kuo, Y. H. (2017). The supply chain design for perishable food with stochastic demand. Sustainability, 9(7), 1195.

[23] Shu, T., Wu, Q., Chen, S., Wang, S., Lai, K. K., & Yang, H. (2017). Manufacturers’/remanufacturers’ inventory control strategies with cap-and-trade regulation. Journal of cleaner production, 159, 11-25.

[24] Tundura, L., & Wanyoike, D. (2016). Effect of inventory control strategies on inventory record accuracy in Kenya power company, Nakuru. Journal of investment and Management, 5(5), 82-92.

[25] Li, N., & Wang, Z. (2023). Inventory control for omnichannel retailing between one warehouse and multiple stores. IEEE Transactions on Engineering Management, 71, 7395-7412.

[26] Gökbayrak, E., & Kayış, E. (2023). Single item periodic review inventory control with sales dependent stochastic return flows. International Journal of Production Economics, 255, 108699.

[27] Tao, S., Liu, S., Zhou, H., & Mao, X. (2024). Research on Inventory Sustainable Development Strategy for Maximizing Cost-Effectiveness in Supply Chain. Sustainability, 16(11), 4442.

[28] Xu, G., Kang, K., & Lu, M. (2023). An omnichannel retailing operation for solving joint inventory replenishment control and dynamic pricing problems from the perspective of customer experience. IEEE Access, 11, 14859-14875.

[29] Yan, R. (2008). Pricing strategy for companies with mixed online and traditional retailing distribution markets. Journal of Product & Brand Management, 17(1), 48-56.

[30] Sui, Z., Gosavi, A., & Lin, L. (2010). A reinforcement learning approach for inventory replenishment in vendor-managed inventory systems with consignment inventory. Engineering Management Journal, 22(4), 44-53.

[31] Mo, D. Y., Tsang, Y. P., Wang, Y., & Xu, W. (2024). Online reinforcement learning-based inventory control for intelligent E-Fulfilment dealing with nonstationary demand. Enterprise Information Systems, 18(2), 2284427.

[32] Selukar, M., Jain, P., & Kumar, T. (2022). Inventory control of multiple perishable goods using deep reinforcement learning for sustainable environment. Sustainable Energy Technologies and Assessments, 52, 102038.

[33] Cuartas, C., & Aguilar, J. (2023). Hybrid algorithm based on reinforcement learning for smart inventory management. Journal of intelligent manufacturing, 34(1), 123-149.

[34] Piao, M., Zhang, D., Lu, H., & Li, R. (2023). A supply chain inventory management method for civil aircraft manufacturing based on multi-agent reinforcement learning. Applied Sciences, 13(13), 7510.

[35] Dogan, I., & Güner, A. R. (2015). A reinforcement learning approach to competitive ordering and pricing problem. Expert Systems, 32(1), 39-48.

[36] Maestre, R., Duque, J., Rubio, A., & Arévalo, J. (2019). Reinforcement learning for fair dynamic pricing. In Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys) Volume 1 (pp. 120-135). Springer International Publishing.

[37] Chen, S., Li, L., Chen, Z., & Li, S. (2020). Dynamic pricing for smart mobile edge computing: A reinforcement learning approach. IEEE Wireless Communications Letters, 10(4), 700-704.

[38] Lu, R., Hong, S. H., & Zhang, X. (2018). A dynamic pricing demand response algorithm for smart grid: Reinforcement learning approach. Applied energy, 220, 220-230.

[39] 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), 6643131.

[40] Xiaoke Deng,Pengcheng Hu,Zhaoyu Li,Wenze Zhang,Dong He & Yuanzhi Chen. (2025). Reinforcement Learning-based five-axis continuous inspection method for complex freeform surface. Robotics and Computer-Integrated Manufacturing,94,102990-102990.

[41] Junsong Lu,Zongsheng Wang,Kang Pan & Hanshuo Zhang. (2024). Research on the influence of multi-agent deep deterministic policy gradient algorithm key parameters in typical scenarios. Journal of Physics: Conference Series,2858(1),012037-012037.

[42] Hua Xu,Juntai Tao,Lingxiang Huang,Chenjie Zhang & Jianlu Zheng. (2025). A Deep Reinforcement Advantage Actor-Critic-Based Co-Evolution Algorithm for Energy-Aware Distributed Heterogeneous Flexible Job Shop Scheduling. Processes,13(1),95-95.