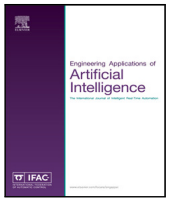
Engineering Applications of Artificial Intelligence 133 (2024) 108073

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Research paper

Airline dynamic pricing with patient customers using deep exploration-based reinforcement learning

Seongbae Jo [a], Gyu M. Lee [b], Ilkyeong Moon [a,c,*]

[a] Department of Industrial Engineering, Seoul National University, Seoul 08826, Republic of Korea
[b] Department of Industrial Engineering, Pusan National University, Busan 46241, Republic of Korea
[c] Institute of Engineering Research, Seoul National University, Seoul 08826, Republic of Korea

ARTICLE INFO

Keywords:
Airline revenue management
Dynamic pricing
Deep reinforcement learning
Patient customer

ABSTRACT

This paper addresses a crucial issue in the airline industry by tackling a dynamic pricing problem in the presence of patient customers, a scenario that has gained significance due to the revenue loss of airlines caused by customers' non-myopic decision-making. To effectively capture this non-myopic characteristic, we propose a Markov decision process (MDP) including a history of offered prices as a state variable. In contrast to previous studies, distributions of customers' properties are assumed to be unknown in advance for a more realistic representation of real-world scenarios. To deal with the new challenges of the problem, we propose utilizing a specific learning framework (i.e., deep exploration-based RL) that is unexplored in this domain. The numerical experiments demonstrate that its performance can be improved on the MDP we designed and show that it outperforms the benchmark algorithm. The structures of pricing policies generated from the bootstrapped deep Q-network algorithm imply that airlines should offer high and low prices alternately from the beginning of the sales period rather than increasing prices as time goes on. We also ascertain that more frequent consecutive high-priced periods can increase airlines' revenue in environments with higher customer patience levels.

1. Introduction

Dynamic pricing is a set of pricing strategies that adjust a price for the same product to get the optimized revenue from heterogeneity and intertemporal shift of customers' willingness to pay (Otero and Akhavan-Tabatabaei, 2015). The airline, fashion retailing, or e-commerce industry can be good examples of sectors where the dynamic pricing can be actively utilized (Ahmadi and Shavandi, 2014; Lei et al., 2018). In 2008, Spanish fashion retailer Zara increased its clearance revenue with the markdown pricing process proposed by Caro and Gallien (2012). Representative international e-commerce platform Amazon is also known to achieve a remarkable increase in revenue through real-time dynamic pricing policies derived from competitors' prices. Likewise, firms need to implement dynamic pricing strategies tailored to the characteristics of their industries in order to gain a competitive edge and pursue profitability.

Recently, airlines have been breaking away from static pricing mechanisms. Technical developments such as new distribution capability (NDC) enabled dynamic adjustment of prices for each fare class (Wittman and Belobaba, 2019). The extent of dynamic pricing strategies can be explained as follows. In basic dynamic pricing, airlines usually do not change pre-defined prices of fare products before the departure. A pre-defined price is selected at each decision point and offered to customers. As a result, the customers observe dynamically changing prices as time passes. In addition, adjustments of prices from the pre-defined prices can be allowed. Airlines can adjust each price of fare products according to the market information they are observing in real-time. Because this paper does not consider observations such as personal information or competitors' prices, the basic dynamic pricing framework is utilized. More specifically, a set of finite prices are defined before the departure, and an airline selects a price from the set for each period. Individual offers for customers are not allowed. Therefore, customers arriving at the same time observe the same prices. We assume that there is a single-fare product for the simplicity of the problem.

Customer behavior in the airline industry is getting more complex. Because many customers recognize that airlines implement dynamic pricing strategies, they do not behave myopically. Even if the ticket price is higher than their willingness to pay, they do not leave the market immediately and keep observing the ticket price. Furthermore, some search engines for flight tickets alert customers when the price of the flight ticket changes. They also provide their expectations of prices for each departure date to support the strategic behavior of customers.

* Corresponding author at: Department of Industrial Engineering, Seoul National University, Seoul 08826, Republic of Korea.
E-mail addresses: tjdqo7779@snu.ac.kr (S. Jo), glee@pnu.edu (G.M. Lee), ikmoon@snu.ac.kr (I. Moon).

https://doi.org/10.1016/j.engappai.2024.108073
Received 11 May 2023; Received in revised form 23 January 2024; Accepted 7 February 2024
Available online 21 February 2024

In these industrial situations, airlines might be able to increase their revenue by considering non-myopic customer behavior for constructing their pricing policies. However, the non-myopic customer behavior has rarely been focused on in the airline revenue management literature. This research gap encouraged us to investigate the impact of non-myopic the customer behavior on the airline dynamic pricing. In order to fill the research gap, we propose an MDP framework to formulate a dynamic pricing problem in the presence of customers with strategic behavior. Because of unpredictable and non-stationary characteristics of demand in the airline industry, we utilize model-free algorithms to solve it.

The rest of this paper is organized as follows. Section 2 presents a summary of the related works and our contributions within this context. Section 3 provides the dynamics of the seller and patient customers in our dynamic pricing problem. The problem is formulated as a Markov decision process (MDP), and its elements are defined in this section. Backgrounds of used reinforcement learning (RL) algorithms and modifications of the benchmark algorithm are presented in Section 4. The results of the numerical experiments are discussed in Section 5, and concluding remarks are presented in Section 6.

2. Literature review

There are two streams of literature related to our work: dynamic pricing with non-myopic customers and RL algorithms applied to complex sequential decision-making problems. Each body of existing literature will be accompanied by corresponding research gaps, and contributions of this study based on those research gaps will be presented in this section.

Many studies defined non-myopic customer models in two representative ways: strategic customers and patient customers. Strategic customers try to figure out sellers' pricing policies and delay their purchase up to their willingness to wait (Aviv and Pazgal, 2008; Den Boer, 2015). Patient customers just wait in the market for a specific number of time periods, which is designated as the willingness to wait. When the observed price is lower than customers' willingness to pay, they immediately purchase the product (Cao et al., 2015; Liu and Cooper, 2015; Lobel, 2020; Zhang and Jasin, 2022). Because airlines try to implement more complex dynamic pricing strategies, it gets difficult for customers to predict the price sequence of the flight ticket. Thus, we focus on the patient customer model. The relevant papers can be briefly introduced as follows. Liu and Cooper (2015) proved the existence of an optimal pricing policy with decreasing cycles under a fixed proportion of homogeneous patient customers. Lobel (2020) proposed a polynomial-time algorithm that can compute an optimal pricing policy when customers have heterogeneous patience levels. Most studies considering the patient customers, including studies by Liu and Cooper (2015) and Lobel (2020), focused on exploring the structure of the optimal pricing policies and constructing algorithms capable of deriving the optimal pricing policies based on customer valuation and patience level distributions that are known to the seller a priori. They can offer managerial insights to practitioners determining prices. However, as the assumption that customer-related distributions are perfectly known is unrealistic in the airline industry, further research is needed for advanced pricing algorithms that do not require such perfect information. In this perspective, Zhang and Jasin (2022) assumed that those distributions are not known to the seller in advance. They proposed an online learning and optimization algorithm to find the best decreasing cyclic or threshold-regulated policy. However, a limitation still lies in its inability to explore various structured pricing policies that can yield higher revenue under non-stationary demand and finite inventory.

To address these research gaps, we relax the three assumptions: Inventory is finite, demand is non-stationary, and distributions for customer valuation and patience level are not known in advance. We explain the details of the relaxations as follows. For a single flight, the number of seats is insufficient to serve all customers who want to buy a seat for the itinerary. Hence, we need to consider the number of remaining seats for constructing pricing policies. One of the most notable characteristics of customers considered in the airline revenue management literature can be explained by leisure and business customers (Bondoux et al., 2020; Wittman and Belobaba, 2018). Leisure customers are less willing to pay and arrive from the beginning of the selling horizon. In contrast, business customers are more willing to pay and arrive later than leisure customers. To incorporate this characteristic, we relax the assumption that the arriving pattern of customers is stationary. Because the demand of the airline industry is affected by many external factors that airlines cannot control, a pricing algorithm can result in inconsistent revenue when constructed based on known or estimated distributions. Therefore, we assumed the customer characteristic distributions are not known in advance. These relaxations introduce new challenges for the dynamic pricing problems. First of all, non-stationary demand and finite inventory settings increase the computational complexity of the problem. Owing to the structures of optimal pricing policies under stationary demand and infinite inventory, the previous studies could propose efficient pricing algorithms. Because those structures or even the existence of them cannot be guaranteed under the new problem that we proposed, the computational complexity of calculating the optimal pricing policies would increase in the new problem. Furthermore, the unknown demand assumption requires learning-based algorithms. To deal with these challenges, we propose utilizing model-free algorithms (i.e., reinforcement learning) that do not need any assumptions for dynamics of environments such as demand Rana and Oliveira (2014), Mao and Shen (2018), Krasheninnikova et al. (2019), Seo et al. (2021), Yang et al. (2022) and Dixit and ElSheikh (2022). Note that the goal of this study is to assess the viability of a novel algorithm that has remained unexplored in this particular domain. Comparative studies with state-of-the-art algorithms are beyond the scope of this study.

RL is a solution method for sequential decision-making problems. Because many revenue management problems correspond to sequential decision-making problems, RL has been widely utilized in the revenue management literature (Rana and Oliveira, 2014; Pandey et al., 2020; Yang et al., 2022). Gosavii et al. (2002) might be the first study that used RL to solve airline revenue management problems. They adopted RL to accommodate complex environments with random customer arrivals and cancellations dependent on multiple fare classes. Lawhead and Gosavi (2019) proposed a bounded actor–critic algorithm for the seat allocation problem. The proposed algorithm improved the computational overflow of the classical actor–critic algorithm and outperformed the EMSR-b heuristic in large-scale problems. Bondoux et al. (2020) applied deep Q-network (DQN), one of the most well-known RL algorithms, to solve the airline dynamic pricing problem. However, the previous studies that utilized RL in the airline revenue management literature did not focus on specific characteristics such as non-stationary arrival processes or strategic behavior of customers. Incorporating these characteristics requires RL algorithms with efficient exploration, and the reason for this will be discussed later. Extensive literature exists that considers the exploration issue in RL for complex problems. Osband et al. (2016) utilized multiple networks to address the exploration issue that arises in environments with rarely observed rewards at the beginning of episodes. Selim et al. (2022) and Yu et al. (2022) proposed methods based on reachability analysis to efficiently train RL agents by minimizing unnecessary and unsafe exploration. Lopes et al. (2012) proposed using empirical estimates of learning progress for driving exploration in model-based RL. Hong et al. (2018) and Parker-Holder et al. (2020) also utilized behavioral diversity to improve exploration of RL agents. Sekar et al. (2020) proposed a method that leverages planning explore via latent disagreement. Hafez et al. (2019, 2020) also proposed novel approaches to enhance the sample efficiency of RL algorithms by employing intrinsic rewards based on learning progress. Moreover, Tutsoy (2021a,b) introduced

innovative methodologies to produce the optimal policies under real-world constraints and uncertainty. Their approach involves improving the current policy with the instantaneous rewards, which is calculated through the adaptive parametric model. These frameworks can be effective in situations where extrinsic rewards fail to provide good directions for policy improvement, as in the airline dynamic pricing problem with patient customers. Because we focus on showcasing fundamental RL frameworks to provide baselines for problems similar to the dynamic pricing problem we are addressing, we utilized the RL algorithm proposed by Osband et al. (2016). Additional methods, such as reward shaping, will be proposed as future research directions in Section 6.

Based on the research gaps introduced in this section, the contributions of this study can be presented as follows. First, we propose a new MDP framework for the dynamic pricing problem with patient customers, which incorporates realistic assumptions in the airline industry. We show that a history of prices offered in the past has to be considered, and we propose a new sequential decision-making problem based on the history of prices. Second, we test some RL algorithms and present the most appropriate one for the airline industry among them. To the best of our knowledge, the dynamic pricing problem with patient customers under the aforementioned relaxations has never been studied. Therefore, it is valuable to find the approach that can deal with the new challenges and investigate its applicability for the real-world airline industry. Third, we analyze pricing policy structures suitable for the airline industry. Differences between the policy structures when airlines consider patient customers or not are presented, and these can give some managerial insights to airlines that want to increase their revenue by considering patient customers.

3. Problem description

We consider a finite-horizon dynamic pricing problem where a seller is a monopolist. To maximize the revenue, the seller selects the prices of a product for each period from the finite set of pre-defined prices. As mentioned in the previous section, all customers observe the same price at the same time. Without loss of generality, we assume that the product's marginal cost is zero. There is no replenishment, and the seller terminates the system without any penalty if the inventory is out of stock. This problem statement is assumed to represent a situation in which an airline implements the dynamic pricing strategy for a single fare product.

3.1. Dynamics of patient customers

As explained before, customers in our system are patient and have three characteristics that determine how they make decisions in the system: time of arrival, reservation price and patience level. The reservation price indicates customers' maximum willingness to pay. If a sale price presented by the seller is lower than or equal to a customer's reservation price, the customer buys the product. The patience level means the customers' maximum willingness to wait. Customers with patience level k have up to $k+1$ opportunities for purchasing. Specifically, when a customer visits the system in period t, one compares a sale price with one's reservation price from period t to period $t+k$. Within those periods, if one observes that the sale price is lower than or equal to one's reservation price, one immediately makes a purchase and leaves the system. Otherwise, one's patience level decreases by one every period one does not make a purchase. If one's patience level becomes negative, one leaves the system without purchasing. We present these dynamics of patient customers as a flow chart shown in Fig. 1(a).

For a customer who visited in period t, an initial patience level is a random variable defined in G_t, which is a finite set of non-negative integers with the maximum value W. The reservation price for a customer who arrives in period t with patience level k follows a probability distribution with the cumulative density function $F_k^t(\cdot)$. And the probability density function of the number of customers arriving in period t with patience level k is $d_k^t(\cdot)$. We assume that G_t, $F_k^t(\cdot)$ and $d_k^t(\cdot)$ are independent of the inventory and the price offered by the seller in each period. All uncertainties in this study are parametric and lie on the customer side. To be specific, the sources of uncertainties include the number of customers arriving in each period, as well as the reservation prices and patience levels of customers.

3.2. Markov decision process

MDP is one of the frameworks for discrete sequential decision-making problems (Sutton and Barto, 2018). Many sequential decision-making problems with stochastic demand, such as dynamic pricing or inventory management, can be formulated as MDPs (Lawhead and Gosavi, 2019; Ma et al., 2021; Ahiska and King, 2015; He et al., 2023). The key elements of MDPs are states, actions, rewards, and transition probabilities. States capture situations where the formalized system is in. In contrast to previous studies that separate observations of the decision-maker from the states (Pandey et al., 2020; Shou et al., 2022; Bautista-Montesano et al., 2022), we define the states as fully observable information of the system for the decision-maker. When the decision-maker takes an action and gets a reward, the system transits to the next state depending on transition probabilities. If each transition probability does not depend on the immediately preceding state and action only, the decision process does not satisfy the Markov property. Because most methods of finding optimal policies for sequential decision making problems start from solving the Bellman optimality equation and assume the Markov property (Sutton and Barto, 2018), the violation of the Markov property can degrade the quality of solutions. We present how state variables in our dynamic pricing problem can satisfy the Markov property in the following subsection.

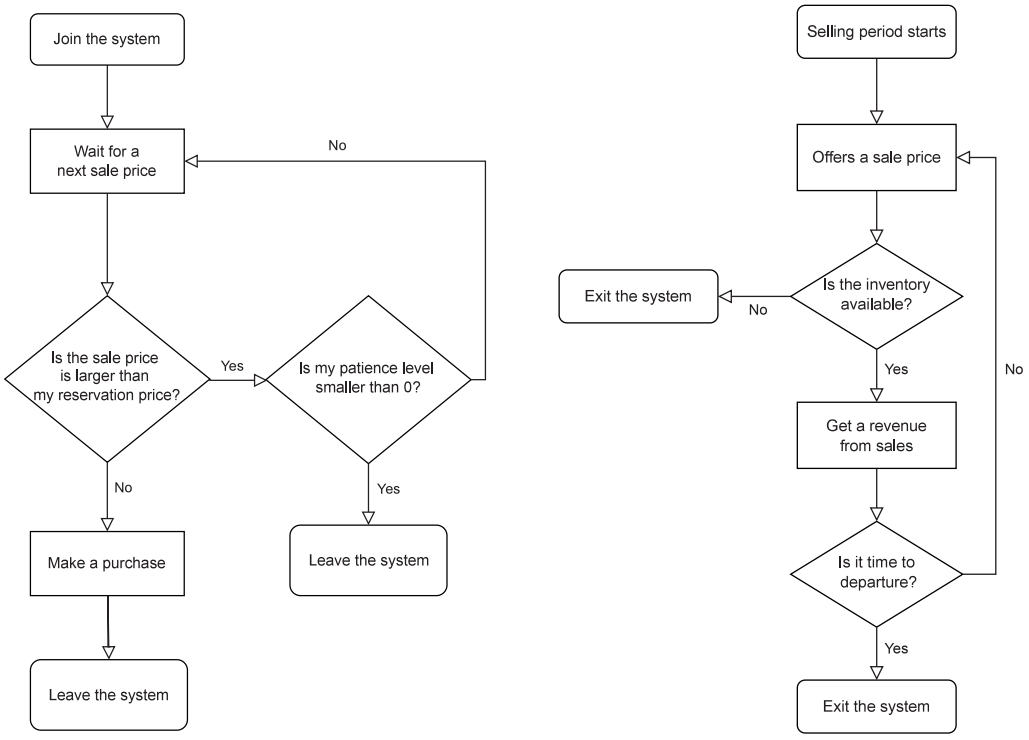
3.3. Airline dynamic pricing

To formulate the airline dynamic pricing problem described at the beginning of Section 3 as an MDP with finite states and actions, we consider an airline as a decision-maker. The behavior of the airline can be presented as in Fig. 1(b). The action space contains possible prices that the airline can offer to customers. Because the objective of the airline in this paper is to maximize the total revenue over the finite selling periods, we set the revenue gained in each period as a reward. In this study, we define the discount factor of the MDP to be one. Before defining our state variables, we define some notations:

- q_t: number of remaining seats in period t
- l_t: remaining time to departure in period t
- W: maximum patience level that customers can have
- s_t: state of the system in period t
- a_t: price offered by the decision-maker in period t
- $\mathcal{S}$: finite set of states
- $\mathcal{A}$: finite set of actions
- $\mathcal{G}_k^t$: group of customers who arrive in period t with patience level k

To show that the last W actions have to be contained in s_t to satisfy the Markov property, we present the procedure of calculating the state transition probability from s_t to s_{t+1} when the state variables are q_t and l_t only, which are commonly used state variables in the airline revenue management literature (Gosavii et al., 2002; Lawhead and Gosavi, 2019; Bondoux et al., 2020; Otero and Akhavan-Tabatabaei, 2015).

We first calculate the probability that the number of seats sold in period t is i. For $t' \le t \le t'+k$, a customer in $\mathcal{G}_k^{t'}$ does not make a purchase until period t if the customer's reservation price is lower than every price in $\{a_{t'}, a_{t'+1}, \dots, a_{t-1}\}$. And if the reservation price is larger than or equal to the offered price in period t which is the lowest price



(a) Customer (b) Seller

Fig. 1. Flow chart of decision-making for a customer and a seller.

in $\{a_{t'}, a_{t'+1}, \ldots, a_t\}$, the customer makes a purchase in period t. From these facts, we can derive the probability that a customer in $\mathcal{G}_k^{t'}$ makes a purchase in period t where $t' \le t \le t' + k$ as follows:

$$p_k^{t',t} = \left(F_k^{t'}(\min\{a_{t'}, \ldots, a_{t-1}\}) - F_k^{t'}(a_t) \right)^+, \tag{1}$$

where $x^+ = \max\{x, 0\}$. This is consistent with the result of Lobel (2020). If the number of customers arriving in period t' with patience level k is $n_k^{t'}$, the probability that $i_k^{t'}$ seats are sold in period t by customers in $\mathcal{G}_k^{t'}$ is $\binom{n_k^{t'}}{i_k^{t'}} \left(p_k^{t',t}\right)^{i_k^{t'}} \left(1 - p_k^{t',t}\right)^{n_k^{t'} - i_k^{t'}}$. In order to calculate the probability that totally i seats are sold in period t for the given $H_t = \{a_{t-W}, \ldots, a_{t-1}\}$ with these results, we introduce additional mathematical notation:

- $\bar{P}_i^t$: probability that i seats are sold in period t for the given $H_t = \{a_{t-W}, \ldots, a_{t-1}\}$
- $X_k^{t',t}$: binomial random variable with parameters $n_k^{t'}$ and $p_k^{t',t}$
- n^t: vector containing the numbers of customers in $\mathcal{G}_k^{t'}$ for $t - W \le t' \le t$ and $t - t' \le k \le W$
- i^t: vector containing the numbers of seats sold to customers in $\mathcal{G}_k^{t'}$ for $t - W \le t' \le t$ and $t - t' \le k \le W$
- N_i^w: set of vectors which have w non-negative integer elements and the sum of the elements is i

Because customers who arrive in periods $t - W, \ldots, t$ can make a purchase in period t, the probability that i seats are sold in period t is calculated as follows:

$$\bar{P}_i^t = \sum_{n^t \ge i^t} \left[\left(\prod_{l=0}^{W} \prod_{u=0}^{l} d_k^t \left(n_{W-u}^{t-W+l} \right) \right) \left(\sum_{i^t \in N_i^{\widehat{W}}} \prod_{l=0}^{W} \prod_{u=0}^{l} P\left(X_{W-u}^{t-W+l,t} = i_{W-u}^{t-W+l} \right) \right) \right], \tag{2}$$

where $\widehat{W} = \frac{(W+1)(W+2)}{2}$ and the first summation is the summation over n^t.

As a result of Eqs. (1) and (2), $\bar{P}_i^t$ depends on $\{a_{t-W}, \ldots, a_{t-1}\}$ which denote the prices the decision-maker selected in periods $t-W, \ldots, t-1$. It implies that if the decision-maker does not consider the actions selected in the last W periods, the expectation for how many seats will be sold in the next period can be completely wrong when customers in the system are patient. Note that we defined s_t with q_t and l_t only. In this case, the inconsistency of $\bar{P}_i^t$ for the same s_t and a_t makes the system not satisfy the Markov property because the transition probability $P(s_{t+1} \mid s_t, a_t)$ cannot be calculated without $\bar{P}_i^t$ where $s_{t+1} = (q_t - i, l_t - 1)$. One simple way to solve this problem is by adding $\{a_{t-W}, \ldots, a_{t-1}\}$ in s_t. The transition probability is still calculated with $\bar{P}_i^t$, but invariability of $\bar{P}_i^t$ for the same s_t and a_t is guaranteed by fixed $\{a_{t-W}, \ldots, a_{t-1}\}$ in s_t. Note that Eqs. (1) and (2) are ordinary equations that represent the stochastic characteristics of the dynamics for the problem considered in this study. However, deriving state variables from them can offer a new perspective to the literature that formulates the airline revenue management problem as MDPs and employs model-free algorithms.

Now, all the components of the MDP formulation are defined. Before the departure, the airline determines a set of prices that can be offered to customers. We call it a pre-defined price set in this study. An action of the agent is selecting a price from the pre-defined price set and offering it to customers. If we denote the pre-defined price set as $\{p_1, \ldots, p_A\}$, the action space $\mathcal{A}$ can be defined as $(p_1, \ldots, p_A)$, and it is the set of positive real numbers. The state space is $S = (q, l, H) \in [0, Q] \times [0, T] \times \mathcal{A}^W$ where H is a vector of size W that contains previously offered prices, as explained above. Q and T are positive integers, which mean the number of total seats and the selling horizon, respectively. The immediate reward from taking action a in state s is denoted by $r(s, a)$. In this study, $r(s, a)$ is defined as immediate revenue when the agent offers price a based on state s. The transition probabilities can be calculated with Eq. (2). The purpose of our problem is to find the optimal pricing policy $\pi^*(s, a)$ that maximizes the expected total revenue over a finite selling horizon T:

$$\max_{\pi} \mathbb{E}^{\pi} \left[\sum_{t=0}^{T-1} r(s_t, a_t) \right] \tag{3}$$

Because the agent can observe the reward immediately after taking an action in a specific state, it can learn a pricing policy close to the

optimal policy with solution methods that are presented in the next section.

4. Solution methods

In this section, we explain why we use model-free deep reinforcement learning (DRL) to solve the airline dynamic pricing problem in the presence of patient customers. As mentioned in Section 1, the previous studies considering patient customers assume that customer arrival rates and distributions of reservation prices and patience levels are known to the decision maker or are unknown but stationary. We relax them because pricing algorithms based on those assumptions can result in serious revenue loss under unpredictable and non-stationary demand (Sutton and Barto, 2018; Rana and Oliveira, 2014; Yang et al., 2022). Model-free RL is a simulation-based approach that can give reasonable pricing policies without any of those assumptions. An agent in model-free RL can learn how to select actions in each state from interactions with an unknown environment. Moreover, as we can see in Eq. (2), transition probabilities are affected by the properties of customers arriving in multiple periods. In a non-stationary environment, heterogeneity of probability distributions for demand properties in each period can make the calculation and storage of the transition probabilities more complex. Because model-free RL does not need the calculation or storage of the transition probabilities, it can be an appropriate method in this case.

An advantage of DRL is its adaptability to problems with high dimensional state space (Gosavii et al., 2002). Because state s_t consists of q_t, l_t, and $(a_{t-W}, \dots, a_{t-1})$, the dimension of the state space is $W+2$. As many real-world customers in the airline industry know that prices of tickets can fluctuate, the maximum patience level of the customers can be large, which increases the dimension of the state space. In a high-dimensional state space, RL without an approximation of value functions or policy functions can be intractable. Therefore, approximators such as neural networks can be utilized to enable the agent to learn policies for challenging real-world problems (Gosavii et al., 2002; Yang et al., 2022).

4.1. Deep Q-network

To solve large-scale problems, parameterized Q-function has been utilized in the value-based RL literature. Instead of tracking Q-values for all state–action pairs, Q-function is estimated by fewer parameters than state–action pairs. Then they are repeatedly updated to approximate the optimal Q-function. The most popular value-based algorithm using parameterized Q-function is deep Q-network (DQN) proposed by Mnih et al. (2013). They used convolutional neural networks to approximate Q-functions for learning optimistic control policies for some Atari 2600 environments. We utilize the DQN algorithm of Mnih et al. (2015) for numerical experiments. The detailed procedure of the algorithm is presented in Appendix A.

4.2. Bootstrapped DQN

Some advanced RL algorithms based on DQN have been studied to achieve practical improvements for more challenging environments (Van Hasselt et al., 2016; Schaul et al., 2015; Wang et al., 2016). One of the difficulties that those challenging problems have is the exploration issue. When comparatively small rewards are given at the states close to the initial state in an episodic environment, RL agents can easily get stuck to those states even if larger rewards can be given at the other states. In this situation, many episodes are needed to learn the optimal policy when RL agents use ineffective exploration methods such as the ϵ-greedy policy. The airline dynamic pricing problem in this study has the same exploration issue. Because the fraction of customers with higher reservation prices increases as time passes and the number of seats is limited, an RL agent learns sub-optimally if it concentrates on maximizing rewards from customers who arrive earlier with lower reservation prices. Indeed, when we conducted numerical experiments using DQN, we observed that the agent learned pricing policies that concentrated on the early periods of the selling horizon, resulting in early episode termination. Therefore, RL algorithms with more effective exploration methods would be required.

Bootstrapped DQN (BDQN) proposed by Osband et al. (2016) is one of the RL algorithms known to be effective for episodic exploration to solve challenging problems. The detailed procedure of the algorithm is presented in Appendix B. In the next section, we provide results of numerical experiments that imply that BDQN is also more effective for the airline dynamic pricing environment. Osband et al. (2016) proposed a shared network architecture where multiple Q-networks share a network directly connected to input data. It learns a feature representation of the input data to reduce computational costs. However, we do not adopt the architecture because our input data is not a two-dimensional image frame. In this case, the shared network can excessively abbreviate the state variables so that the BDQN agent gets insufficient information of the environment. Therefore, we design the BDQN framework with Q-networks that receive state variables directly as input. Each of the Q-networks is individually constructed in the same way as the DQN architecture we used. Fig. 2(a) shows the network architecture of DQN, and Fig. 2(b) shows the architecture of BDQN which consists of multiple networks whose structures are the same as those shown in Fig. 2(a).

4.3. Optimistic learning for decreasing cyclic policies

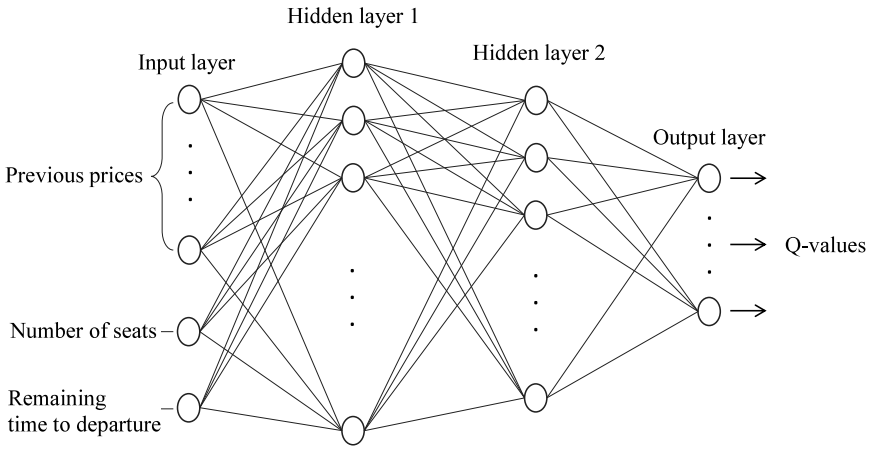
We use a variant of the optimistic learning for decreasing cyclic policies (OLD) algorithm proposed by Zhang and Jasin (2022) to evaluate RL algorithms for our problem. To the best of our knowledge, Zhang and Jasin (2022) is a unique study that does not assume that the joint distribution of reservation price and patience level is known to the seller in advance. However, the original OLD algorithm is designed based on the idea that a decreasing cyclic policy is near-optimal when the selling horizon is very large and demand is stationary. Therefore, we modified the original OLD algorithm for a comparatively short selling horizon and non-stationary demand in the airline dynamic pricing environment.

First, we reconstructed a recursive equation that calculates the optimal decreasing cyclic policy with given expected revenue under specific conditions. Those specific conditions can be satisfied on stationary demand only. Therefore, we must relax them for non-stationary demand even though it increases the computational cost. The second part of the modification is on the condition for the recalculation of decreasing cyclic policy. We found that the original recalculation condition could not be frequently satisfied in our environment. As a result, the performance of decreasing cyclic policy is not improved as the number of training episodes increases. To deal with this problem, the modified OLD algorithm reconstructs the policy once it is used.

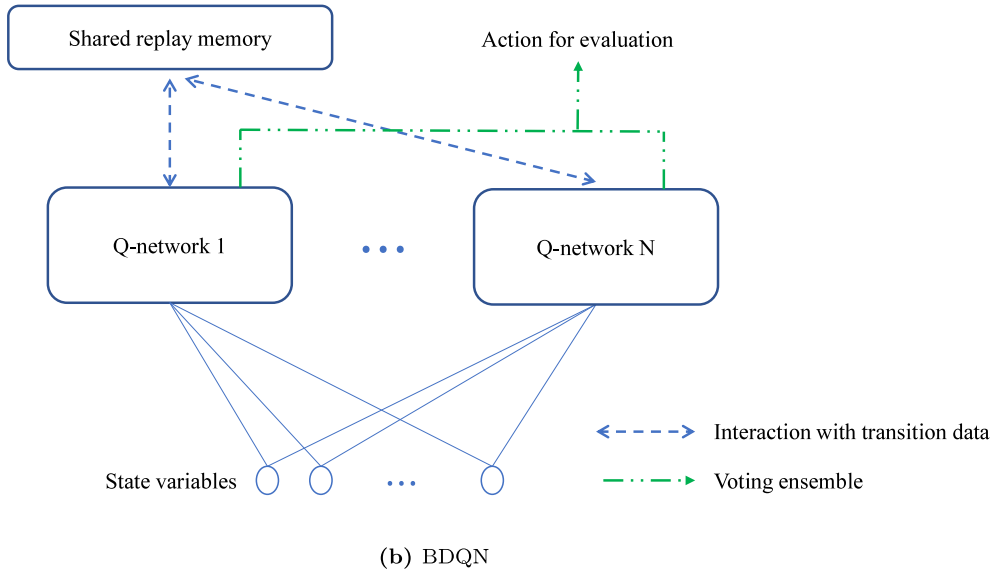
5. Numerical experiments

In this section, we present the results of numerical experiments. We constructed an artificial environment for simulations of the dynamic pricing. The seller and the customers make their decisions in the simulations as shown in Fig. 1. The first subsection demonstrates the effectiveness of the proposed MDP formulation in the presence of patient customers. In the second subsection, we compare the performance of algorithms for the environments where demand is non-stationary and inventory is insufficient. In the last subsection, we analyze the structures of pricing policies.

In implementing RL algorithms, we fix the value of ϵ as one and do not update network parameters for the initial 1000 episodes, which is common practice in the RL community. After the 1000 initialization episodes, we linearly decrease ϵ to 0.1. The terminal state mentioned in



(a) DQN



(b) BDQN

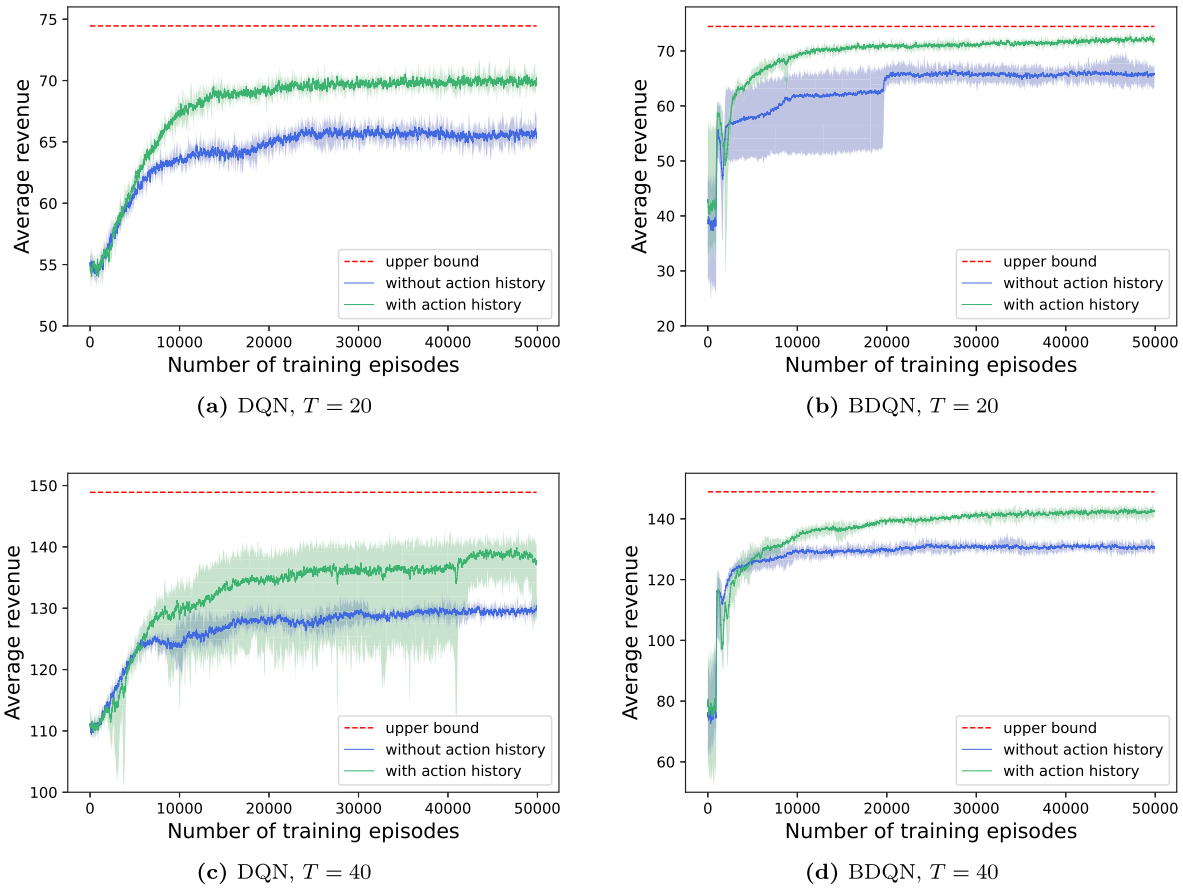
Fig. 2. Architecture of DQN and BDQN for airline dynamic pricing.

Algorithms 1 and 2 means the situation in which there are no remaining seats, or in which time until departure is zero in our problem. The size of the replay memory and each minibatch is set to be 300,000 and 512, respectively. For the SGD method, the learning rate is 0.001, and the error between current Q-values and target Q-values is calculated by Huber loss. Target networks are updated by copying the corresponding policy networks every 15 episodes, and the parameters of policy networks are updated every five episodes for computational efficiency. The remaining time to departure in the state is one-hot encoded for RL algorithms. The structure of the Q-network in DQN is as follows. The number of input nodes is equal to the dimension of the state space, and the number of output nodes equals the number of actions. We use two hidden layers with 100 and 50 nodes each. In BDQN, we use five networks with the same structure as in DQN. The values of hyperparameters explained above are set so that the RL algorithms give adequate results as presented in the next subsection. All experiments are conducted using a Python 3 with an Intel Core i5-9400F and 16 GB of RAM.

5.1. Comparison between MDP formulations in the presence of patient customers

We construct two different MDP formulations in this subsection. The only difference between them is the state s_t. We call an MDP that contains a history of actions $(a_{t-W}, \dots, a_{t-1})$ in its state s_t as MDP with action history. Otherwise, if a state of an MDP contains only q_t and l_t, we define it as MDP without action history. Therefore, MDP without action history does not satisfy the Markov property when patient customers exist in the system. The convergence or performance improvement of DQN and BDQN cannot be theoretically guaranteed even when the Markov property is satisfied. However, we identified revenue improvement in the MDP with action history compared to the one without action history from the results of numerical experiments.

We set $\mathcal{A} = \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $T \in \{20, 40\}$ and $W = 11$. As mentioned in Section 3.3, T and W mean the selling horizon and the maximum patience level, respectively. These values are set based on numerical experiments of the previous studies that considered airline



(a) DQN, $T = 20$ (b) BDQN, $T = 20$

(c) DQN, $T = 40$ (d) BDQN, $T = 40$

Fig. 3. Average revenue of recent 100 episodes over five random seeds in training episodes.

revenue management problems or dynamic pricing problems with patient customers (Lawhead and Gosavi, 2019; Lobel, 2020). Note that we are now concentrating on evaluating the effectiveness of using action history in the presence of patient customers, regardless of the characteristics of the airline industry. Therefore, we assume a deterministic and stationary arrival process of customers and sufficient inventory to serve all customers in this subsection. To be more specific, the simulator is constructed as follows. Initial patience levels for customers arriving in period t are uniformly distributed over $G_t = \{0, 1, \dots, W\}$. For every $k \in G_t$ and $t \in \{1, \dots, T\}$, reservation prices for customers in the group $\mathcal{G}_k^t$ are generated from the standard uniform distribution, and the number of arriving customers in group $\mathcal{G}_k^t$ is set to be one. The total inventory is 300 when $T = 20$ and 500 when $T = 40$. We conduct 5 runs of 50,000 training episodes for every experimental instance.

Fig. 3 represents the average revenue over the recent 100 episodes for each training episode. This means that every point in the curve indicates an average over the last 100 training episodes. For example, the point on episode 20,000 indicates the average revenue over episodes 19,901 to 20,000. The bold line is the mean of the average revenue over five random seeds, and the shaded area shows the interval between the minimum and maximum value of average revenue over the same five seeds. The dotted red line represents an upper bound calculated by the same algorithm as Lobel (2020) proposed. As mentioned in Section 1, he constructed a dynamic program when the arrival of patient customers is stationary and inventory is infinite. If the distributions related to customers are known in advance, the algorithm can derive the optimal policy and corresponding revenue in a polynomial time. Therefore, it can provide the upper bound for the expected total revenue per episode in our problem.

Table 1
Average revenue of evaluation episodes over five random seeds.

Case	Customer	T	Algorithm	Without action history[a]	With action history[b]
1	Patient	20	DQN	66.31	71.24
2	Patient	20	BDQN	65.66	72.68
3	Patient	40	DQN	129.82	141.32
4	Patient	40	BDQN	131.07	144.45
5	Myopic	40	DQN	119.69	119.13
6	Myopic	40	BDQN	119.94	120.04

[a] Mean of average revenue under the MDP without action history.
[b] Mean of average revenue under the MDP with action history.

The results in Fig. 3 imply that using action history can generate revenue closer to the upper bound regardless of which RL algorithms are used. In other words, the seller can improve the revenue by constructing the pricing policies depending on the prices offered recently. In Figs. 3(a) and 3(c), it is likely that DQN fails to construct policies close to the optimal. However, DQN encourages the agent to explore non-optimal policies in the training episodes. If the agent excludes the exploration and selects the optimal actions with probability 1 in the evaluation episodes, the gap between the revenue of DQN and the upper bound can decrease.

Table 1 provides the results of DQN and BDQN in evaluation episodes. The first value in the last column indicates the mean of the average revenue with action history over five random seeds, and the second value is for entities without action history. Each of Cases 1, 2, 3, and 4 correspond to each experiment in Fig. 3. As in the training episodes, when the action history is included in the state,

Table 2
Computational time for each algorithm.

Figure	T	W	Algorithm	Time[a]
Fig. 4(a)	50	11	DQN	128.76
			BDQN	263.20
			Modified OLD	147.49
Fig. 4(b)	50	29	DQN	156.74
			BDQN	314.11
			Modified OLD	166.46
Fig. 4(c)	100	11	DQN	273.86
			BDQN	568.80
			Modified OLD	289.19
Fig. 4(d)	100	29	DQN	318.13
			BDQN	723.36
			Modified OLD	200.65

[a] Time: average computational time to complete 50,000 training episodes (in minutes).

the average revenue in the evaluation episodes increases regardless of RL algorithms. In addition, policies calculated from DQN and BDQN generate revenue close to the upper bounds (the optimal expected revenues with infinite inventory in Cases 2 and 4 are 74.45 and 149.8, respectively). These numerical results imply that without perfect information for dynamic pricing environments, the DRL algorithms can find policies close to the optimal policy.

To verify that the improvement of performance on the RL algorithms truly comes from using action history, we additionally examine its effectiveness when there are no patient customers at all. Cases 5 and 6 represent the system where all customers are myopic (i.e., the patience levels of all customers are zero). When there are no patient customers, both MDP formulations with and without the action history satisfy the Markov property. Therefore, if the average revenue increases from using action history in Cases 5 and 6, we cannot say that the satisfaction of the Markov property can improve the performances of the RL algorithms in our dynamic pricing problem. However, as Table 1 shows, there is no difference in the average revenue between the two MDP formulations. From this result, we can more obviously conclude that using action history improve the performance of RL algorithms when patient customers exist in the system.

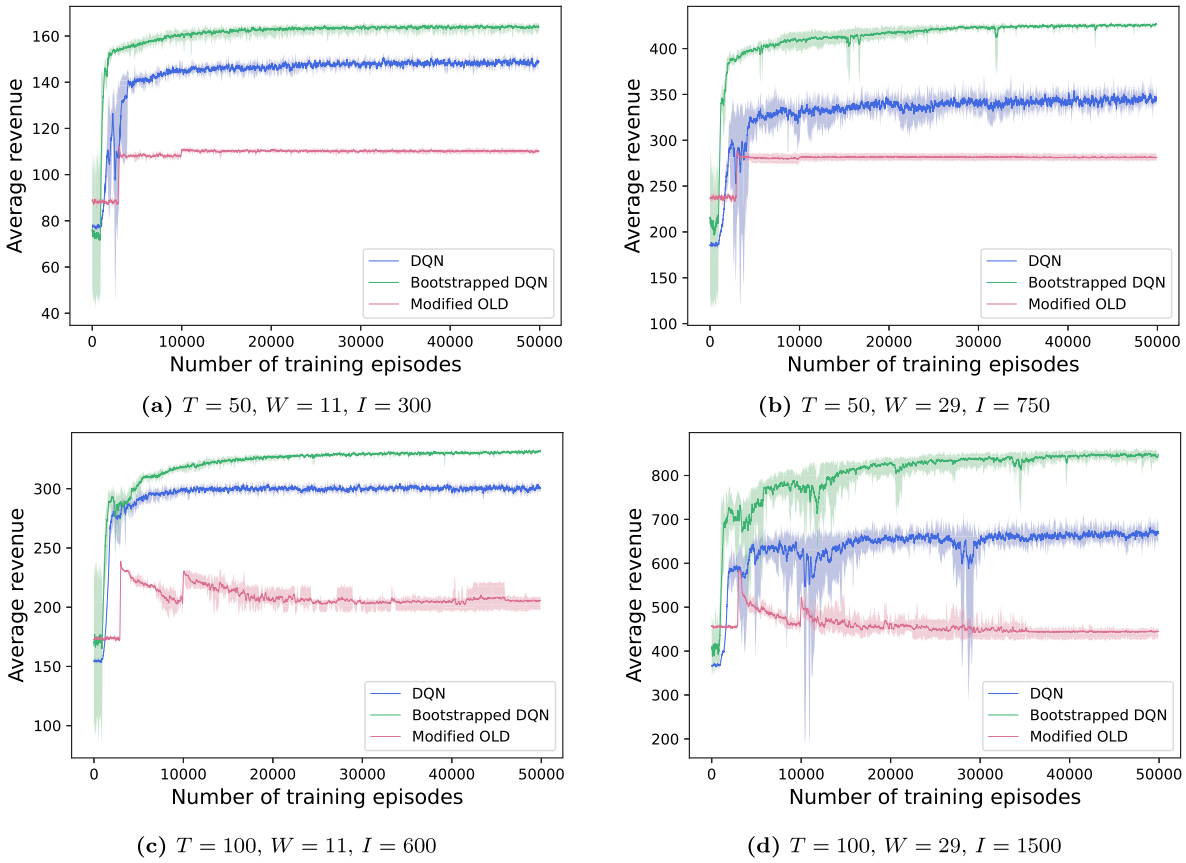
5.2. Comparison between pricing algorithms for non-stationary demand and insufficient inventory

In this subsection, we demonstrate the pricing algorithms with the MDP with action history when demand is non-stationary and the inventory is insufficient. We divide customers into leisure and business customers, one of the most commonly used customer segmentation methods in the airline industry (Li et al., 2014; Bondoux et al., 2020; Varella et al., 2017). All elements that make up the MDP formulation are defined identically in the previous section and 5 runs of 50,000 training episodes are also conducted. Both leisure and business customers are assumed to arrive in the Poisson process, as previous studies generally assumed (Flapper et al., 2012; Yousuk and Luong, 2013; Yu et al., 2019). The arrival rate of leisure customers decreases linearly as time passes, and vice versa for business customers. Reservation prices for both types of customers are generated from uniform distributions, but business customers have higher average reservation prices than leisure customers. The expected number of customers arriving in the system is set to be twice the number of seats. The number of total seats is denoted by I in this subsection. To verify which algorithm is appropriate for this situation, we compare the performances of each algorithm for some instances. Fig. 4 shows the learning curves of each algorithm where all elements in the figure are the same as in Fig. 3. The corresponding average computational times to complete 50,000 training episodes for the DQN, BDQN, and modified OLD algorithms are shown in Table 2.

Figs. 4(a) and 4(b) show that both RL algorithms outperform the modified OLD algorithm. Structures of the policies generated from the modified OLD algorithm no longer change after about 20,000 training episodes. Furthermore, the number of decision points in each episode must be less than those in RL algorithms because the modified OLD algorithm cannot make a new decision before the generated pricing policy ends. This might make the algorithm incapable of responding to stochastic and non-stationary demand in our problem. As a result, the revenue from the modified OLD algorithm is stable over episodes but remarkably lower than in RL algorithms. In contrast to the modified OLD algorithm, the performance of RL algorithms rapidly improved at the initial learning interval, where exploration for the optimal pricing policy is encouraged. After that, they kept updating their Q-networks to get more stable and improved revenue. We can ascertain this from Figs. 4(b) and 4(d) where the shaded area between the best and worst learning curves decreases as the number of training episodes increases. In Figs. 4(c) and 4(d), the modified OLD algorithm exhibits declining learning curves, indicating that it fails to learn suitable policies under large-scale problems. In contrast to the RL algorithms, the modified OLD algorithm generated policies that offer the lowest price in several periods. When the maximum patience level increases, such pricing policies result in selling an excessive number of seats early on and ultimately cause the premature termination of the selling period with lower revenue. This pattern can also be observed from the fact that the learning curves of the modified OLD algorithm in Fig. 4(d) were generated in a shorter time compared to the learning curves in Fig. 4(c), as in Table 2.

As explained in the previous subsection, even though the average revenue of BDQN is higher than DQN in training episodes, we cannot conclude that BDQN performs better than DQN owing to the difference between the exploration methods of the two RL algorithms. The DQN algorithm selects non-optimal actions with probability ϵ, and BDQN does not in training episodes. Therefore, we ran 1000 evaluation episodes with five trained agents to compare the performance of the RL algorithms. In the evaluation episodes, the DQN agent selects the optimal actions generated from the trained Q-network with probability 1. Note that the BDQN algorithm uses multiple Q-networks and randomly selects one of them to construct a training episode. In contrast, it selects actions based on ensemble policy with every trained Q-network in evaluation episodes. The results are presented in Table 3 and Fig. 5. Unlike in the training episodes, there is no difference between DQN and BDQN when $W = 11$. But when $W = 29$, the gap between the mean of average revenue from two RL algorithms becomes larger. This implies that BDQN might give higher revenue than DQN when the size of the problem (i.e., the maximum patience level, the number of total customers and the initial inventory) becomes larger.

Furthermore, we can figure out that BDQN gives more consistent revenue over multiple learning for the same problem compared to DQN from Fig. 6. It shows distributions of total revenue per evaluation episode over five random seeds when $W = 29$. Each smoothing curve indicates a continuous density estimate obtained through a kernel density estimation with a Gaussian kernel. These curves were provided to enhance the clarity in distinguishing various distributions. In Fig. 6(a), total revenue from seeds 1,2 and 4 have similar shapes of distributions but seed 3 and 5 show different shapes. In Fig. 6(c), all the distributions are significantly different from each other. This implies that airlines could be suffering from unexpected revenue loss for the same itinerary product when they construct pricing policies based on the DQN algorithm even though they considered the purchase behavior of patient customers. In contrast to DQN, BDQN can give consistent revenue for the same environments as shown in Figs. 6(b) and 6(d). When $T = 50$, all distributions have almost the same mean and variance. They become slightly different when T increases, but have sufficiently consistent structures compared to DQN. Therefore, we can conclude that the BDQN algorithm might give the highest and



(a) $T = 50$, $W = 11$, $I = 300$

(b) $T = 50$, $W = 29$, $I = 750$

(c) $T = 100$, $W = 11$, $I = 600$

(d) $T = 100$, $W = 29$, $I = 1500$

Fig. 4. Average revenue when the demand is non-stationary and the number of seats is limited.

Table 3
Average revenue of evaluation episodes over five random seeds.

T	W	Algorithm	Mean	Maximum	Minimum	Variance
50	11	DQN	164.60	164.95	164.12	0.09
		BDQN	164.38	165.54	162.30	1.30
		Modified OLD	110.69	110.81	109.10	0.40
50	29	DQN	409.39	424.38	387.88	177.45
		BDQN	427.36	428.52	425.72	0.87
		Modified OLD	281.38	285.55	277.78	8.70
100	11	DQN	332.17	333.86	330.29	2.45
		BDQN	332.50	335.17	330.23	2.78
		Modified OLD	205.44	208.50	197.81	15.11
100	29	DQN	753.17	812.97	690.15	1755.44
		BDQN	856.09	863.52	847.53	40.35
		Modified OLD	444.60	451.78	429.03	64.11

stable revenue among the three pricing algorithms in the presence of patient and non-stationary customers with limited inventory.

5.3. Structure of pricing policies from the BDQN algorithm

In this subsection, we analyze the structures of pricing policies calculated using the BDQN algorithm. Because the BDQN algorithm gives consistent policies as shown in Fig. 6, we randomly select a trained BDQN agent and run 10,000 evaluation episodes with the agent. We set $T = 50$ and $W \in \{11, 29\}$. The number of seats is 100, and the expected number of customers arriving to buy tickets is set to be 200. As in Section 5.2, customers are segmented by leisure and business groups. Fig. 7 indicates trajectories of prices under non-stationary demand. It shows prices offered by the airline over the selling horizon. In Figs. 7(a) and 7(b), 10,000 evaluation episodes are simulated to calculate average actions in each period, and from Figs. 7(c) to 7(f) are examples randomly selected between those evaluation episodes. The first MDP formulation in Section 5.1 is used for the situation in which airlines construct pricing policies considering the effectiveness of patient customers, and the second MDP formulation in Section 5.1 is used to capture characteristics of policies when the airlines do not consider patient customers at all. The green and red lines correspond to pricing policies learned under the first and second MDP formulations, respectively. In addition, Fig. 8 shows how many seats remain and when selling periods are terminated according to each policy structure in Figs. 7(a) and 7(b). The values next to the horizontal bars indicate how many episodes satisfy the constraints in the vertical axis. For example, the first value next to the horizontal bar colored in red means that 1368 episodes out of 10,000 evaluation episodes are terminated with a positive number of remaining seats.

From Figs. 7(a), 7(b), 7(e) and 7(f), we can figure out that if airlines do not consider the effectiveness of patient customers in relation to their revenue, low prices are offered in initial periods, and prices

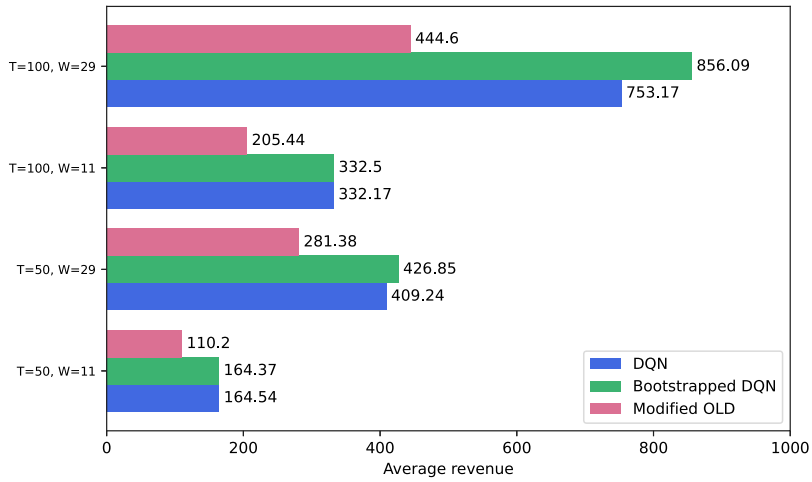
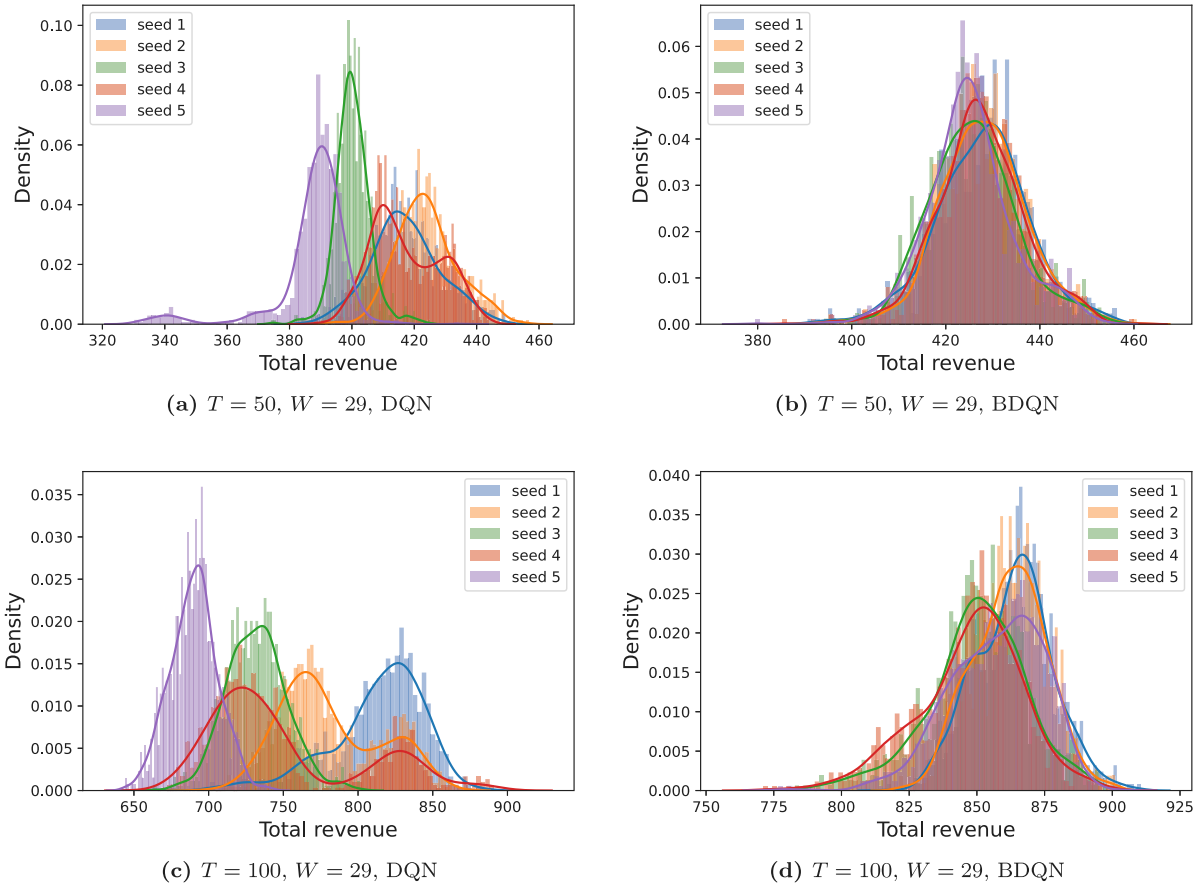


Fig. 5. Mean of the average revenue in evaluation episodes over five random seeds.



(a) $T = 50$, $W = 29$, DQN

(b) $T = 50$, $W = 29$, BDQN

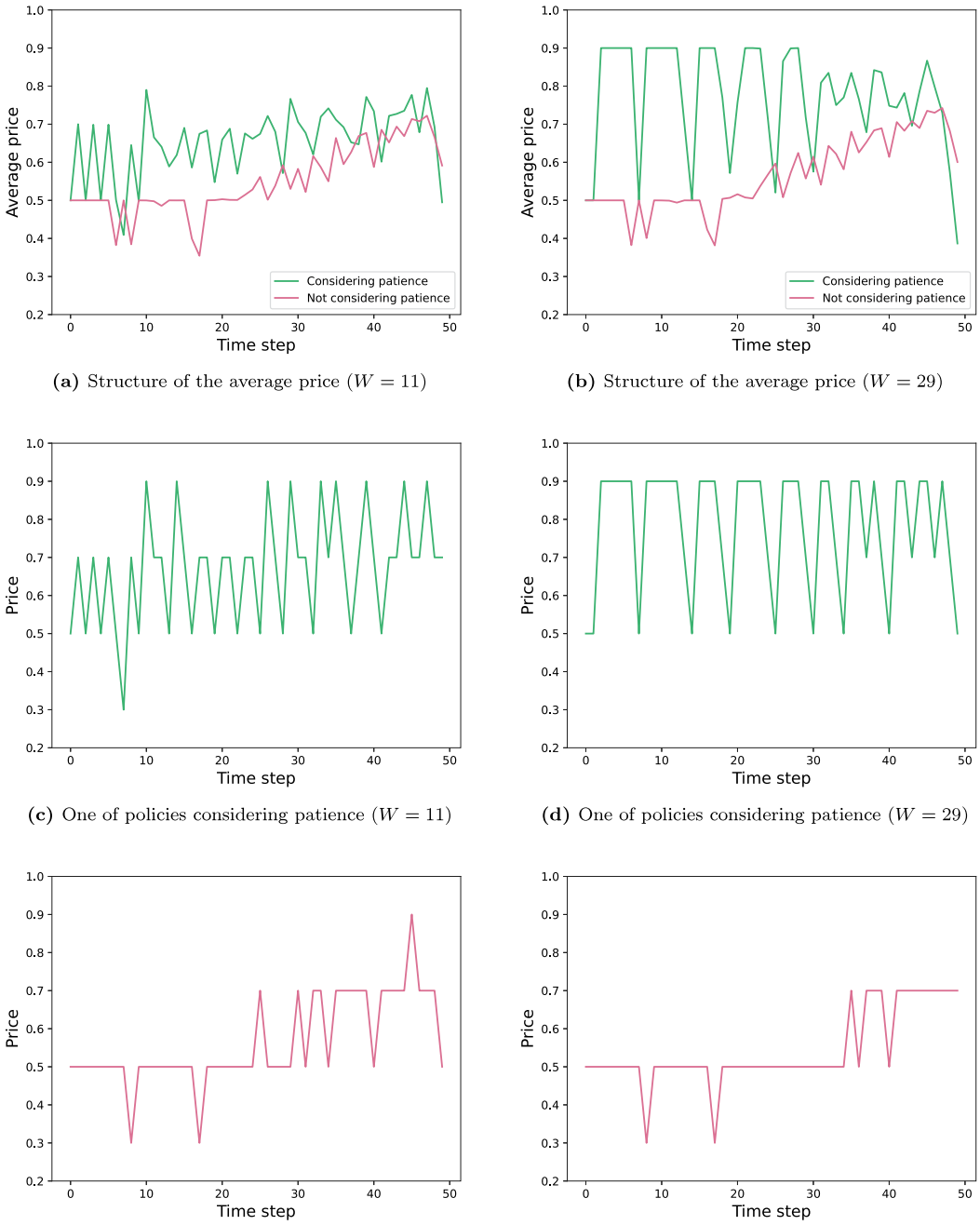
(c) $T = 100$, $W = 29$, DQN

(d) $T = 100$, $W = 29$, BDQN

Fig. 6. Distributions of the total revenue in evaluation episodes over five random seeds.

become higher as the departure date approaches. However, if airlines are aware of patient customers, high and low prices are alternately offered throughout the entire selling horizon as presented in Figs. 7(a) and 7(b). In the end, comparatively lower prices are offered to maximize revenue by minimizing unsold seats at the departure time. Under this pattern of pricing policies, fractions of episodes where they are terminated at the time of departure and terminated with non-zero remaining seats are larger than the increasing price structures as shown in Fig. 8. However, because higher average revenue is observed under the structure of green lines in Fig. 7, we can conclude that it is a more appropriate structure for the non-stationary system with patient customers.

From comparing the green lines in Figs. 7(a) and 7(b), we know that the average gap between higher and lower prices increases as the maximum patience level of customers increases. And the number of consecutive periods in which the highest price is offered also increases. These trends can be interpreted as follows. Airlines are trying to generate more revenue from customers with extremely high reservation prices by offering the highest price more frequently. Because the fraction of customers with higher patience levels becomes larger, revenue



(a) Structure of the average price ($W = 11$)

(b) Structure of the average price ($W = 29$)

(c) One of policies considering patience ($W = 11$)

(d) One of policies considering patience ($W = 29$)

(e) One of policies that do not consider patience ($W = 11$)

(f) One of policies that do not consider patience ($W = 29$)

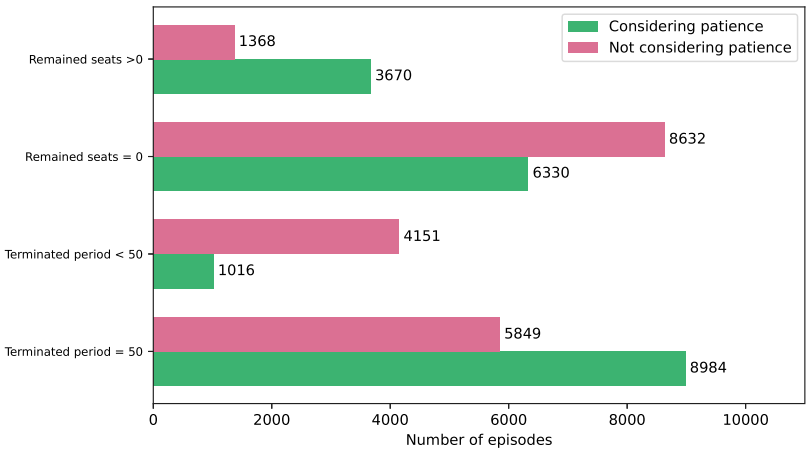
Fig. 7. Structures of pricing policies.

loss from selling their seats at lower prices than customers' willingness to pay can grow significantly. Furthermore, additional revenue in a low-priced period might be larger than when $W = 11$. As a result, they provide low prices infrequently and increase the fraction of periods in which higher prices are offered.

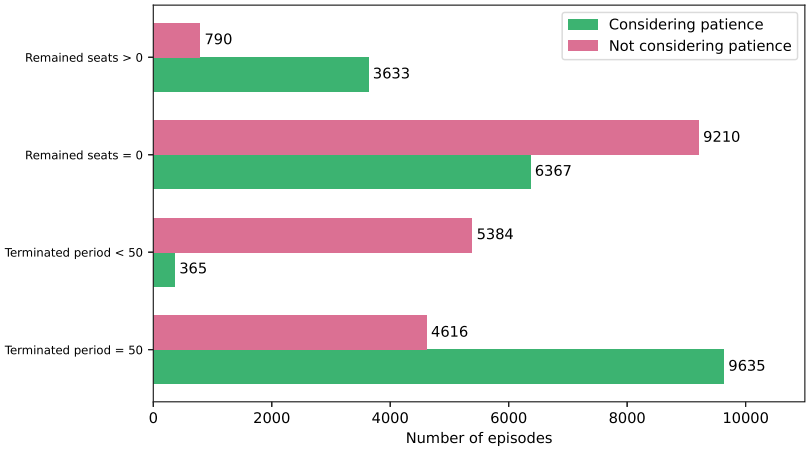
6. Conclusions

Patterns of purchasing behavior in the airline industry are becoming more complex. Customers do not leave the market immediately, keep observing price changes, and try to buy tickets at an acceptable price. Airlines can get additional revenue if they consider these patterns in their pricing policies. To capture this strategic characteristic, we adopted the patient customer model. To the best of our knowledge, this paper is the first to examine the dynamic pricing framework with limited inventory under non-stationary demand and unknown distributions for properties of patient customers. The MDP framework was designed to formulate this new dynamic pricing problem. We included the history of prices in state variables to satisfy the Markov property when customers are patient. Due to the high dimensional state space and complexity of transition probabilities, we used DQN and BDQN algorithms.

Comparisons between the MDP formulations with and without the action history were conducted, and we showed that using action history can improve the performance of the DRL algorithms. This implies that airlines should construct their pricing policies based on the prices offered in the past when the existence of patient customers is expected. Furthermore, we identified a small gap between revenue from the



(a) $W = 11$



(b) $W = 29$

Fig. 8. Number of episodes grouped by terminated period and the number of remaining seats.

model-free DRL algorithms and the upper bound calculated with perfect information.

This study also compared two DRL algorithms with the modified OLD algorithm. Both DRL algorithms outperformed the modified OLD algorithm in terms of average revenue. However, the variance of DQN significantly increased in large-scale problems, whereas BDQN did not. This difference came from the approach of exploration, which means that the RL agent without efficient exploration failed to learn appropriate policies when most customers with a higher willingness to pay come at the end of the sales period.

The structural characteristics of pricing policies were analyzed in the last part of the numerical experiments. Airlines constructed increasing pricing policies when they did not consider patient customers. In contrast, they offered high and low prices alternately if they considered the effectiveness of patient customers to their revenue. We identified that this price structure results in higher revenue than the increasing structure. Therefore, airlines should accommodate the alternating price structure when the existence of patient customers is expected. Furthermore, we found that airlines should increase the number of consecutive high-priced periods and infrequently offer lower prices when the patience levels of customers are expected to be large.

In this study, we focused on providing the deep exploration-based RL framework that is unexplored in dynamic pricing problems with non-myopic customers. While it can be employed as a baseline framework in this domain, it has some limitations being utilized directly in the real world. First, we did not consider a situation in which the size of the possible price set becomes large. In that situation, DQN or BDQN, value-based DRL algorithms, would not make good pricing policies. To handle this problem, policy-based DRL algorithms that are developed for continuous action spaces can be utilized. Second, many factors that airlines consider in the real-world were not considered. Factors such as holidays, weather, or competitors' prices must be considered in the real-time applications of airline pricing algorithms. In the MDP framework, by adding those factors to state variables, their effectiveness and patient customers can be simultaneously reflected in dynamic pricing environments. Third, even though the RL agent does not require any information for probability distributions included in the environment, additional numerical experiments under various probability distributions of patience levels, reservation prices, or the number of arriving customers should be required. Because we conducted the numerical experiments under the limited type of probability distributions, an overfitting issue may arise. Therefore, the performance of the considered algorithms should be investigated under various probability distributions in the dynamic pricing environment. Furthermore, future research directions regarding solution methods can be provided as follows. While BDQN demonstrated stable and effective learning, reward shaping methods might be able to help the BDQN

agent achieve higher sample efficiency. Defining an intrinsic reward function, as in Hafez et al. (2020), could be one example. Alternatively, applying the network structure of BDQN to advanced solution methods proposed in Tutsoy (2021b) and in Kulkarni et al. (2016) could be a promising future research direction.

CRediT authorship contribution statement

Seongbae Jo: Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Gyu M. Lee:** Conceptualization, Methodology, Writing – review & editing. **Ilkyeong Moon:** Conceptualization, Supervision, Validation, Writing – review & editing.

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

Data will be made available on request.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning [Grant Number NRF-2021R1A4A1029124]. The authors are grateful for the valuable comments from the associate editor and seven anonymous reviewers.

Appendix A. DQN algorithm

The procedure of the DQN algorithm can be explained as follows. Q-function is approximated by Q-network, which is parameterized with ϕ. Updating ϕ is done by the stochastic gradient descent (SGD) method, whose goal is to find ϕ minimizing the mean squared error between parameterized Q-function Q_ϕ and optimal Q-function Q^*. Because Q^* is not known unlike in situations of supervised learning, gradients of the mean squared error are calculated with the estimated target value, which is $r+\gamma \max_{a' \in \mathcal{A}} Q_{\phi_k}(s', a')$ in the kth update. Therefore, updating the network parameters in batch can be done by Eq. (A.1):

$$\phi_{k+1} = \phi_k - \alpha \sum_{j \in \mathcal{M}} \left[\left(Q_{\phi_k}(s_j, a_j) - \left(r_j + \gamma \max_{a' \in \mathcal{A}} Q_{\phi_k}(s'_j, a') \right) \right) \nabla_{\phi_k} Q_{\phi_k}(s_j, a_j) \right] \tag{A.1}$$

$\mathcal{M}$ denotes a minibatch selected from the replay memory D which stores samples that the agent gained from previous interactions with the environment. Due to the moving target in Eq. (A.1), the stability of the SGD method can be degraded. To alleviate this problem, Mnih et al. (2015) used an additional target Q-network denoted by $Q_{\phi_k^-}$. Q_{ϕ_k} of the target value in Eq. (A.1) is substituted with the target network $Q_{\phi_k^-}$, and it copies the original Q-network every C updates.

Appendix B. BDQN algorithm

The BDQN algorithm utilizes multiple Q-networks for exploration. One of the networks is randomly selected to make steps in a training episode, and the transition tuples are stored in the shared replay memory. Then, a minibatch of transitions is randomly selected from the replay memory as in DQN. Some Q-networks use it to update their weights, and some do not. The randomness in selecting and updating multiple Q-networks leads an agent to try consecutive sub-optimal actions for multiple time steps. Numerical experiments by Osband et al. (2016) showed that the BDQN agent could get out from sub-optimal policies in significantly fewer training episodes than the DQN agent.

Algorithm 1 DQN algorithm

Initialize replay memory D
Initialize parameters ϕ for Q-network
Set $\phi^- = \phi$
for *episode* $= 1, \ldots, C$ **do**
  Initialize state s_0
  Set $t = 0$
  while s_t *is not the terminal state* **do**
    Select a random action a_t with probability ϵ, otherwise $a_t = \text{argmax}_a Q_\phi(s_t, a)$
    Implement action a_t and get reward r_t and next state s_{t+1}
    Store transition data (s_t, a_t, r_t, s_{t+1}) in replay memory D
    Randomly select a minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from D
    Set target $y_j = \begin{cases} r_j & \text{if } s_{j+1} \text{ is the terminal state} \\ r_j + \max_{a'} Q_{\phi^-}(s_{j+1}, a') & \text{otherwise} \end{cases}$
    Update ϕ using the SGD method according to Eq. (A.1) on y_j
    Set $t = t + 1$
  end
  Set $\phi^- = \phi$ every E episodes
end

Algorithm 2 BDQN algorithm

Initialize replay memory D
for $i = 1$ *to* N **do**
  Initialize parameters ϕ_i for Q-network i
  Set $\phi_i^- = \phi_i$
end
for *episode*$= 1, \ldots, C$ **do**
  Select $k \sim$ Uniform{1, …, N}
  Initialize state s_0
  Set $t = 0$
  while s_t *is not the terminal state* **do**
    Set $a_t = \text{argmax}_a Q_{\phi_k}(s_t, a)$
    Implement action a_t and get reward r_t and next state s_{t+1}
    for $i = 1$ *to* N **do**
      Generate a bootstrap mask $u_i \sim$ Bernoulli($\frac{1}{2}$) for Q-network i
    end
    Set vector of bootstrap masks $\hat{u_t} = (u_1, \ldots, u_N)$
    Store transition data $(s_t, a_t, r_t, s_{t+1}, \hat{u_t})$ in replay memory D
    Randomly select a minibatch of transitions $(s_j, a_j, r_j, s_{j+1}, \hat{u_j})$ from D
    for $i = 1$ *to* N **do**
      Set target $y_j = \begin{cases} r_j & \text{if } s_{j+1} \text{ is the terminal state} \\ r_j + \max_{a'} Q_{\phi_i^-}(s_{j+1}, a') & \text{otherwise} \end{cases}$
      Update ϕ_i using the SGD method if ith element of $\hat{u_j}$ is equal to 1, otherwise do not update
    end
    Set $t = t + 1$
  end
  Update target networks every E episodes
end

References

Ahiska, S.S., King, R.E., 2015. Inventory policy characterisation methodologies for a single–product recoverable manufacturing system. Eur. J. Ind. Eng. 9 (2), 222–243.
Ahmadi, M., Shavandi, H., 2014. Joint pricing and rationing in a production system with two demand classes. Eur. J. Ind. Eng. 8 (6), 836–860.
Aviv, Y., Pazgal, A., 2008. Optimal pricing of seasonal products in the presence of forward-looking consumers. Manuf. Serv. Oper. Manag. 10 (3), 339–359.
Bautista-Montesano, R., Galluzzi, R., Ruan, K., Fu, Y., Di, X., 2022. Autonomous navigation at unsignalized intersections: A coupled reinforcement learning and model predictive control approach. Transp. Res. C 139, 103662.

Bondoux, N., Nguyen, A.Q., Fiig, T., Acuna-Agost, R., 2020. Reinforcement learning applied to airline revenue management. J. Revenue Pricing Manag. 19 (5), 332–348.

Cao, P., Fan, M., Liu, K., 2015. Optimal dynamic pricing problem considering patient and impatient customers' purchasing behaviour. Int. J. Prod. Res. 53 (22), 6719–6735.

Caro, F., Gallien, J., 2012. Clearance pricing optimization for a fast-fashion retailer. Oper. Res. 60 (6), 1404–1422.

Den Boer, A.V., 2015. Dynamic pricing and learning: historical origins, current research, and new directions. Surv. Oper. Res. Manag. Sci. 20 (1), 1–18.

Dixit, A., ElSheikh, A.H., 2022. Stochastic optimal well control in subsurface reservoirs using reinforcement learning. Eng. Appl. Artif. Intell. 114, 105106.

Flapper, S., Gayon, J.-P., Vercraene, S., 2012. Control of a production–inventory system with returns under imperfect advance return information. European J. Oper. Res. 218 (2), 392–400.

Gosavii, A., Bandla, N., Das, T.K., 2002. A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking. IIE Trans. 34 (9), 729–742.

Hafez, M.B., Weber, C., Kerzel, M., Wermter, S., 2019. Efficient intrinsically motivated robotic grasping with learning-adaptive imagination in latent space. In: 2019 Joint Ieee 9th International Conference on Development and Learning and Epigenetic Robotics. Icdl-Epirob, IEEE, pp. 1–7.

Hafez, M.B., Weber, C., Kerzel, M., Wermter, S., 2020. Improving robot dual-system motor learning with intrinsically motivated meta-control and latent-space experience imagination. Robot. Auton. Syst. 133, 103630.

He, J., Liu, X., Duan, Q., Chan, W.K.V., Qi, M., 2023. Reinforcement learning for multi-item retrieval in the puzzle-based storage system. European J. Oper. Res. 305 (2), 820–837.

Hong, Z.-W., Shann, T.-Y., Su, S.-Y., Chang, Y.-H., Fu, T.-J., Lee, C.-Y., 2018. Diversity-driven exploration strategy for deep reinforcement learning. Adv. Neural Inf. Process. Syst. 31.

Krasheninnikova, E., García, J., Maestre, R., Fernández, F., 2019. Reinforcement learning for pricing strategy optimization in the insurance industry. Eng. Appl. Artif. Intell. 80, 8–19.

Kulkarni, T.D., Narasimhan, K., Saeedi, A., Tenenbaum, J., 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. Adv. Neural Inf. Process. Syst. 29.

Lawhead, R.J., Gosavi, A., 2019. A bounded actor–critic reinforcement learning algorithm applied to airline revenue management. Eng. Appl. Artif. Intell. 82, 252–262.

Lei, Y., Jasin, S., Sinha, A., 2018. Joint dynamic pricing and order fulfillment for e-commerce retailers. Manuf. Serv. Oper. Manag. 20 (2), 269–284.

Li, J., Granados, N., Netessine, S., 2014. Are consumers strategic? Structural estimation from the air-travel industry. Manage. Sci. 60 (9), 2114–2137.

Liu, Y., Cooper, W.L., 2015. Optimal dynamic pricing with patient customers. Oper. Res. 63 (6), 1307–1319.

Lobel, I., 2020. Dynamic pricing with heterogeneous patience levels. Oper. Res. 68 (4), 1038–1046.

Lopes, M., Lang, T., Toussaint, M., Oudeyer, P.-Y., 2012. Exploration in model-based reinforcement learning by empirically estimating learning progress. Adv. Neural Inf. Process. Syst. 25.

Ma, Y., Hao, X., Hao, J., Lu, J., Liu, X., Xialiang, T., Yuan, M., Li, Z., Tang, J., Meng, Z., 2021. A hierarchical reinforcement learning based optimization framework for large-scale dynamic pickup and delivery problems. Adv. Neural Inf. Process. Syst. 34, 23609–23620.

Mao, C., Shen, Z., 2018. A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network. Transp. Res. C 93, 179–197.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

Osband, I., Blundell, C., Pritzel, A., Van Roy, B., 2016. Deep exploration via bootstrapped DQN. Adv. Neural Inf. Process. Syst. 29.

Otero, D.F., Akhavan-Tabatabaei, R., 2015. A stochastic dynamic pricing model for the multiclass problems in the airline industry. European J. Oper. Res. 242 (1), 188–200.

Pandey, V., Wang, E., Boyles, S.D., 2020. Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations. Transp. Res. C 119, 102715.

Parker-Holder, J., Pacchiano, A., Choromanski, K.M., Roberts, S.J., 2020. Effective diversity in population based reinforcement learning. Adv. Neural Inf. Process. Syst. 33, 18050–18062.

Rana, R., Oliveira, F.S., 2014. Real-time dynamic pricing in a non-stationary environment using model-free reinforcement learning. Omega 47, 116–126.

Schaul, T., Quan, J., Antonoglou, I., Silver, D., 2015. Prioritized experience replay. arXiv preprint arXiv:1511.05952.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., Pathak, D., 2020. Planning to explore via self-supervised world models. In: International Conference on Machine Learning. PMLR, pp. 8583–8592.

Selim, M., Alanwar, A., Kousik, S., Gao, G., Pavone, M., Johansson, K.H., 2022. Safe reinforcement learning using black-box reachability analysis. IEEE Robot. Autom. Lett. 7 (4), 10665–10672.

Seo, D.-W., Chang, K., Cheong, T., Baek, J.-G., 2021. A reinforcement learning approach to distribution-free capacity allocation for sea cargo revenue management. Inform. Sci. 571, 623–648.

Shou, Z., Chen, X., Fu, Y., Di, X., 2022. Multi-agent reinforcement learning for Markov routing games: A new modeling paradigm for dynamic traffic assignment. Transp. Res. C 137, 103560.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.

Tutsoy, O., 2021a. COVID-19 epidemic and opening of the schools: Artificial intelligence-based long-term adaptive policy making to control the pandemic diseases. IEEE Access 9, 68461–68471.

Tutsoy, O., 2021b. Pharmacological, non-pharmacological policies and mutation: an artificial intelligence based multi-dimensional policy making algorithm for controlling the casualties of the pandemic diseases. IEEE Trans. Pattern Anal. Mach. Intell. 44 (12), 9477–9488.

Van Hasselt, H., Guez, A., Silver, D., 2016. Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence.

Varella, R.R., Frazão, J., Oliveira, A.V., 2017. Dynamic pricing and market segmentation responses to low-cost carrier entry. Transp. Res. E 98, 151–170.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N., 2016. Dueling network architectures for deep reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 1995–2003.

Wittman, M.D., Belobaba, P.P., 2018. Customized dynamic pricing of airline fare products. J. Revenue Pricing Manag. 17, 78–90.

Wittman, M.D., Belobaba, P.P., 2019. Dynamic pricing mechanisms for the airline industry: A definitional framework. J. Revenue Pricing Manag. 18, 100–106.

Yang, C., Feng, Y., Whinston, A., 2022. Dynamic pricing and information disclosure for fresh produce: An artificial intelligence approach. Prod. Oper. Manage. 31 (1), 155–171.

Yousuk, R., Luong, H.T., 2013. Modelling a two-retailer inventory system with preventive lateral transshipment using expected path approach. Eur. J. Ind. Eng. 7 (2), 248–274.

Yu, X., Gao, S., Hu, X., Park, H., 2019. A Markov decision process approach to vacant taxi routing with e-hailing. Transp. Res. B 121, 114–134.

Yu, D., Ma, H., Li, S., Chen, J., 2022. Reachability constrained reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 25636–25655.

Zhang, H., Jasin, S., 2022. Online learning and optimization of (some) cyclic pricing policies in the presence of patient customers. Manuf. Serv. Oper. Manag. 24 (2), 1165–1182.