공학석사학위논문

Airline Dynamic Pricing with Patient Customers Using Deep Exploration-Based Reinforcement Learning

전략적 고객 행동을 고려한 심층 강화학습 기반 항공사 동적 가격 결정 연구

2023 년 2 월

서울대학교 대학원

산업공학과

조 성 배

Airline Dynamic Pricing with Patient Customers Using Deep Exploration-Based Reinforcement Learning

전략적 고객 행동을 고려한 심층 강화학습 기반 항공사 동적 가격 결정 연구

지도교수 문 일 경

이 논문을 공학석사 학위논문으로 제출함

2022 년 12 월

서울대학교 대학원

산업공학과

조 성 배

조성배의 공학석사 학위논문을 인준함

2023 년 1 월

위 원 장 이 재 욱 (인)

부위원장 문 일 경 (인)

위 원 박 건 수 (인)

Abstract

Airline Dynamic Pricing with Patient Customers Using Deep Exploration-Based Reinforcement Learning

Sungbae Jo
Department of Industrial Engineering
The Graduate School
Seoul National University

This thesis considers an airline dynamic pricing problem in the presence of patient customers. Nowadays, customers behave strategically to pay lower than their willingness to pay because they know airlines are implementing dynamic pricing strategies. To capture the 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. Deep reinforcement learning (DRL) algorithms are utilized to solve it, and the results of numerical experiments are presented to show that their performance can be improved with the proposed formulation. Comparisons between algorithms are also made to determine which can construct appropriate pricing structures for the patient and non-stationary demand. 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.

Keywords: Airline revenue management, Dynamic pricing, Reinforcement learning, Deep Q-network, Bootstrapped deep Q-network, Patient customer

Student Number: 2021-23948

Contents

List of Tables

List of Figures

Chapter 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 (WTP). It can be utilized in industries with two characteristics as follows: perishability of goods and low cost for changing a price. The airline or fashion retailing industry can be good examples of the former because tickets for planes that have already left or obsolete fashion items will not generate additional revenue after a specific sales horizon ends. The latter case includes the e-commerce industry, where a large amount of cost is not frequently incurred from changing a price. In 2008, Spanish fashion retailer Zara increased its clearance revenue with the markdown pricing process proposed by [9]. 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. Therefore, many firms are encouraged to implement dynamic pricing strategies.

The airline industry is where revenue management has been implemented systematically since its deregulation in the 1970s. The standard mechanism of traditional airline revenue management can be decomposed into two sequential processes: pric-

ing and seat allocation [34]. In the pricing phase, once prices for each fare class are determined, they do not change before the flight departs. In the seat allocation phase, seats for each fare class are opened for customers according to the pre-defined schedule. 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. Some new pricing mechanisms even allow airlines to eliminate the fare classes and select a price of an itinerary product from an unrestricted set of prices.

The International Air Transport Association (IATA) classifies concepts of those dynamic revenue management mechanisms into two aspects: the way of determining airline products and the way of adjusting their prices. In product determination, most products of current airlines are static, meaning pre-defined fare products do not change over time. A mechanism with a dynamic determination of product bundles can be proposed, but we do not consider it in this thesis. The extent of dynamic pricing strategies under static fare products 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 thesis 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 in the hope that it will fall. Some customers even try to predict pricing patterns and purchase at the lowest price they expect. Furthermore, some search engines for flight tickets, such as Skyscanner, alert customers when the price of the flight ticket changes or provide their expectations of prices for each departure date to support the strategic behavior of customers. In these industrial situations, airlines might be able to increase their revenue by considering non-myopic customer behavior for constructing their pricing policies.

To capture the characteristics of customers described above, many studies defined customer models in two representative ways: strategic customers and patient customers. Strategic customers try to figure out sellers' pricing policies [2, 11]. They are ready to delay their purchase up to their willingness to wait. They anticipate the trajectory of price changes and purchase at the time when their utilization is expected to be maximized. The strategic customer model is appropriate for the markets where customers can learn the pricing policy from the frequent experience of purchasing the same product. [10] assumed that all customers are strategic, arrive at the beginning of the selling horizon, and have a willingness to wait that is greater than the length of the selling horizon. Under these assumptions, they investigated

how problem factors such as the number of price changing by the seller or scarcity of the inventory affect the seller's revenue. [6] relaxed the extreme assumptions of [10] and considered more generalized properties of customers. They showed that finding an optimal pricing policy with few assumptions for customers is tractable when arrivals of customers are stationary and how the computational cost for finding an optimal policy increases under non-stationary environments. Branching off from constructing optimal pricing policies, [15] estimated the proportion of strategic customers across leisure and business markets in the airline industry. They observed that the fraction of strategic customers increases at the beginning and end of the booking period except for popular leisure destinations.

In contrast to strategic customers, patient customers do not try to learn the pricing policies of sellers. They 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 their willingness to pay, they immediately purchase the product. [8], [17], [18], and [36] considered the patient customer model under various problem settings. [17] proved the existence of an optimal pricing policy with decreasing cycles under an infinite selling horizon and a fixed proportion of homogeneous patient customers. From numerical experiments, they showed that finding optimal decreasing cyclic policies can be a good heuristic approach for problems with heterogeneous patience levels. Relaxing the assumption that patient customers have the same patience levels, [18] proposed a polynomial-time algorithm that can compute an optimal pricing policy for a finite selling horizon. Although [17] and [18] did not restrict customer valuation and patience level distributions to specific probability distributions, both assumed that those distributions are known to the seller a priori. In contrast, [36]

considered the situation where customer valuation and patience level distributions are not known to the seller in advance. Instead of calculating an optimal policy, [36] proposed an online learning and optimization algorithm to find the best decreasing cyclic or threshold-regulated policy for finite-horizon dynamic pricing problems in the presence of patient customers. They found that as the selling horizon increase, the revenue gap between the policy calculated from the proposed algorithm and the optimal policy decreases. Compared with the naive upper confidence bound algorithm [1], they also recalled the importance of considering the existence of patient customers when constructing pricing policies.

Because airlines try to implement more complex dynamic pricing strategies, it gets difficult for customers to predict the price sequence of the flight ticket, and the strategic customer model is inappropriate for the airline industry. Therefore, we focus on the patient customer model. Previous studies considering patient customers assumed that inventory is infinite, demand is stationary, and distributions for customer valuation and patience level are known in advance. To propose a suitable pricing algorithm for the airline industry, we relax those assumptions: inventory is finite, demand is non-stationary, and distributions for customer valuation and patience level are not known in advance. 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 construct pricing policies considering the number of remaining seats. It is known that demand is non-stationary in the airline industry. One of the most notable characteristics of the airline industry can be explained by leisure and business customers. 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. Considering the different behaviors of leisure and business customers, 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 can not control, parametric estimation of distributions may not be correct. This implies that a pricing algorithm can result in inconsistent revenue when it is constructed based on known or estimated distributions. For this reason, we use model-free algorithms (i.e., reinforcement learning) that do not need any assumptions for demand [25, 19, 27, 35].

Many revenue management problems can be formulated as sequential decision-making problems (e.g., dynamic pricing, capacity allocation). Reinforcement learning (RL) is one of the solution methods for revenue management problems. It can be applied to various industries because it does not assume a specific model structure [16]. [25] tested the Q-learning algorithm and Q(λ) algorithm for dynamic pricing problems with finite inventory, finite selling horizon, and non-stationary demand. Compared to the parametric learning algorithm, both RL algorithms gave consistent revenue whether demand distributions were estimated correctly or not. [24] formulated dynamic pricing of express lanes as a partially observable Markov decision process. They utilized policy-based and actor-critic methods due to their continuous action space. Compared to other heuristic algorithms, RL algorithms gave higher revenue even though the heuristic algorithms assumed full observability and RL algorithms did not. [35] simulated three scenarios of pricing fresh products using RL: naive pricing, quality-based pricing without information disclosure, and quality-based pricing with information disclosure. They adopted RL because it does not need any properties of demand to construct pricing policies. Furthermore, their

three pricing scenarios could be solved by the same RL algorithm with a simple redefinition of state variables.

RL can be widely utilized for revenue management problems in the airline industry. [13] might be the first study that used RL to solve airline revenue management problems. They considered seat allocation and overbooking simultaneously. RL is adopted to accommodate complex environments with random customer arrival and cancellations dependent on multiple fare classes. In every sample problem, the revenue generated from RL was higher than the nested version of expected marginal seat revenue (EMSR), one of the widely used heuristic algorithms in the airline industry. [14] 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. In contrast to the studies considering seat allocation, [7] considered the dynamic pricing problem in the airline industry. Because current airlines estimate the parametric demand model first and optimize their revenue based on the model, dealing with the defects of the parametric estimation can increase their revenue, as mentioned in [25]. In this perspective, [7] applied deep Q-network (DQN), one of the most well-known RL algorithms. It does not need an estimated demand model to construct pricing policies but too many interactions with the environment are required. To overcome this complexity, they combined the parametric estimation of the demand model and DQN. They showed that initialization of the Q-network from the estimated demand model enabled DQN to learn pricing policy close to the optimal using less data.

Contributions of this study compared to previous studies can be presented as follows. First, we determine which information should be considered to construct

pricing policies under non-stationary and patient demand. As mentioned above, we assume the customers' characteristics can not be known or estimated in advance. In reality, the available information can be obtained only by the airlines' observations. We show that a history of prices offered in the past can be the information and 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. Furthermore, we show that the selected RL algorithm outperforms the benchmark algorithm proposed by [36]. 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.

The rest of this thesis is organized as follows. Chapter 2 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 chapter. Backgrounds of used RL algorithms and modifications of the benchmark algorithm are presented in Chapter 3. The results of the numerical experiments are discussed in Chapter 4, and concluding remarks are presented in Chapter 5.

Chapter 2

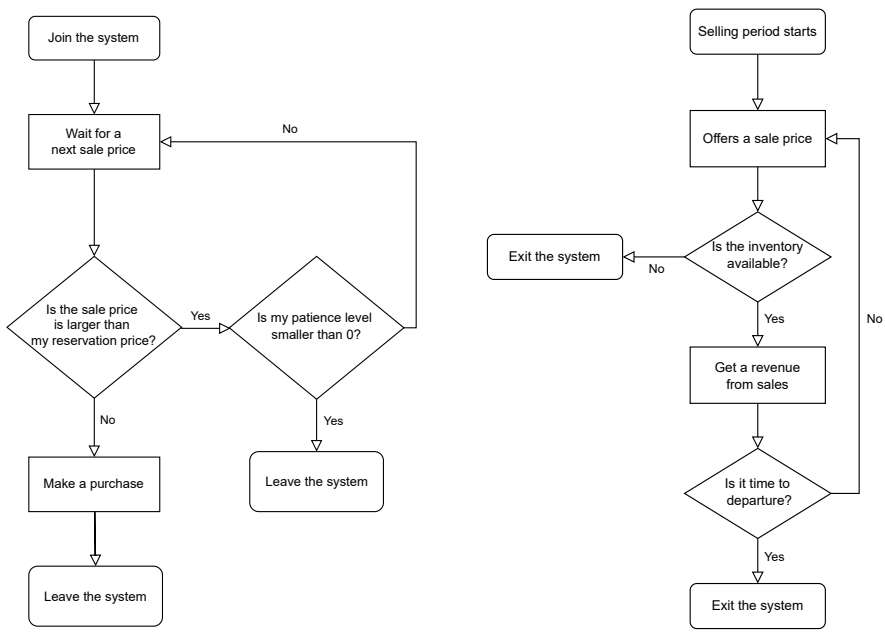
Problem description

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. 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.

2.1 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 Figure 2.1a.



(a) Customer (b) Seller

Figure 2.1: Flow chart of decision-making for a customer and a seller

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.

2.2 Markov decision process

MDP is one of the frameworks for discrete sequential decision-making problems [29]. 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 [24, 28, 4], 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 [29], 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 section.

2.3 Airline dynamic pricing

To formulate the airline dynamic pricing problem described at the beginning of Chapter 2 as an MDP with finite states and actions, we consider an airline as a decision-maker. The action space contains possible prices that the airline can offer to customers. Because the objective of the airline in this thesis 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 [13, 14, 7, 23].

We first calculate the probability that the number of seats sold in period t is i. For $t' \leq t \leq t' + k$, 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 $\{a_{t'}, a_{t'+1}, \ldots, a_{t-1}\}$. And if the reservation price is larger than or equal to the offered price in period t which is the lowest price 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 $t' \leq t \leq t' + k$ as follows:

$$p_k^{t',t} = \left(F_k^{t'}(\min\{a_{t'}, \ldots, a_{t-1}\}) - F_k^{t'}(a_t)\right)^+, \tag{2.1}$$

where $x^+ = \max\{x, 0\}$. This is consistent with the result of [18]. 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 $H_t = \{a_{t-W}, \cdots, a_{t-1}\}$ with these results, we introduce additional mathematical notation:

- $\bar{P}_i^t$: probability that i seats are sold in period t for the given $H_t = \{a_{t-W}, \cdots, a_{t-1}\}$
- $X_k^{t',t}$: 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 $t - W \leq t' \leq t$ and $t - t' \leq k \leq W$
- i^t: vector containing the numbers of seats sold by customers in $\mathcal{G}_k^{t'}$ for $t - W \leq t' \leq t$ and $t - t' \leq k \leq W$
- 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:

$$\bar{P}_i^t = \sum_{n^t \geq i^t} \left[\left(\prod_{l=0}^{W} \prod_{u=0}^{l} d_k^t \left(n_{W-u}^{t-W+l} \right) \right) \left(\sum_{i^t \in N_i^{\widehat{W}}} \prod_{l=0}^{W} \prod_{u=0}^{l} P\left(X_{W-u}^{t-W+l,t} = i_{W-u}^{t-W+l} \right) \right) \right], \tag{2.2}$$

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

As a result of Equations (2.1) and (2.2), $\bar{P}_i^t$ depends on $\{a_{t-W}, \cdots, a_{t-1}\}$ which denote the prices the decision-maker selected in periods $t - W, \cdots, t - 1$. 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 $P(s_{t+1} \mid s_t, a_t)$ can not 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 $\{a_{t-W}, \cdots, a_{t-1}\}$ 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 $\{a_{t-W}, \cdots, a_{t-1}\}$ in s_t.

Now, all the components in the MDP formulation for the dynamic pricing problems with patient customers are defined. The action space $\mathcal{A}$ contains all pre-defined prices the decision-maker can offer customers, and the reward $r(s, a)$ is the revenue the decision-maker gains when one takes action a in state s. The state space $\mathcal{S}$ is the set of states containing the amount of inventory, the number of remaining periods to the departure, and the last W prices that the decision-maker offered before. Hence, the transition probabilities can be calculated with Equation (2.2). The purpose of our problem is to find the optimal pricing policy that maximizes the expected total revenue over a finite selling horizon T:

$$\max_{\pi} \mathbb{E}^{\pi} \left[\sum_{t=0}^{T-1} r(s_t, a_t) \right] \tag{2.3}$$

Chapter 3

Solution methods

In this Chapter, we explain why we use model-free RL to solve the airline dynamic pricing problem in the presence of patient customers and describe model-free RL algorithms we used in numerical experiments. As mentioned in Chapter1, the 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 [29, 25, 35]. 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.

Another advantage of model-free RL is its adaptability to problems with high dimensional state space and complex transition probabilities [13]. Because state s_t consists of q_t, l_t, and $(a_{t-W}, \cdots, a_{t-1})$, 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. Moreover, as we can see in Equation (2.2), tran-

sition 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. Model-free RL can be an effective methodology in this situation because it does not require calculating and storing any transition probabilities. In addition, approximation of the value functions or policy functions with non-linear approximators such as neural networks enables the agent to learn policies for challenging real-world problems [13, 35].

Model-free RL can be divided into value-based and policy-based algorithms. Because the Q-function that satisfies the Bellman optimality equation presented in Equation (3.1) is the optimal Q-function, the value-based RL algorithms try to solve Equation (3.1) with iterative updates of the Q-function.

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s',a) \tag{3.1}$$

One of the most well-known value-based algorithms is the Q-learning proposed by [33]. In order to update the Q-function, it uses transition pairs gained from interactions with the environment. Equation (3.2) represents the k th update of the Q-function with a transition pair (s,a,r,s') and the learning rate α in Q-learning:

$$Q_{k+1}(s,a) = (1-\alpha)Q_k(s,a) + \alpha \left[r(s,a) + \gamma \max_{a' \in \mathcal{A}} Q_k(s',a') \right] \tag{3.2}$$

Under certain conditions, $Q_k(s,a)$ for $s \in \mathcal{S}$ and $a \in \mathcal{A}$ solve the Equation (3.1) when $k \to \infty$. However, the convergence to the optimal Q-function requires too high number of iterations in large-scale environments for real-world problems.

3.1 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 [20]. They used convolutional neural networks to approximate Q-functions for learning optimistic control policies for some Atari 2600 environments. They also utilized an experience replay mechanism, which is effective for breaking the correlation between consecutive samples.

More specifically, 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 Equation (3.3):

$$\phi_{k+1} = \phi_k - \alpha \sum_{j \in \mathcal{M}} \left[\left(Q_{\phi_k}(s_j, a_j) - \left(r_j + \gamma \max_{a' \in \mathcal{A}} Q_{\phi_k}(s'_j, a') \right) \right) \nabla_{\phi_k} Q_{\phi_k}(s_j, a_j) \right] \tag{3.3}$$

$\mathcal{M}$ denotes a minibatch selected from the replay memory $\mathcal{D}$ which stores samples that the agent gained from previous interactions with the environment. Due to the moving target in Equation (3.3), the stability of the SGD method can be degraded.

To alleviate this problem, [21] used an additional target Q-network denoted by $Q_{\phi_k^-}$. Q_{ϕ_k} of the target value in Equation (3.3) is substituted with the target network $Q_{\phi_k^-}$, and it copies the original Q-network every C updates. We utilized the DQN algorithm of [21] for numerical experiments and presented the procedure of the algorithm for our dynamic pricing problem as follows:

Algorithm 1 DQN algorithm for airline dynamic pricing

Initialize replay memory $\mathcal{D}$
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 $a_t =_a Q_\phi(s_t, a)$
    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 $\mathcal{D}$
    Randomly select a minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from $\mathcal{D}$
    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 Equation (3.3) on y_j
    Set $t = t + 1$
  end
  Set $\phi^- = \phi$ every E episodes
end

3.2 Bootstrapped DQN

Some advanced RL algorithms based on DQN have been studied to achieve practical improvements for more challenging environments [30, 26, 32]. 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. Therefore, RL algorithms with more effective exploration methods would be required to solve the airline dynamic pricing problem.

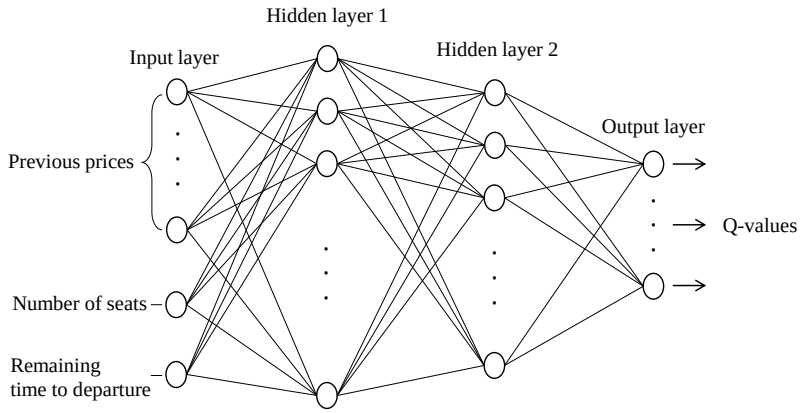
Bootstrapped DQN (BDQN) proposed by [22] is one of the RL algorithms known to be effective for episodic exploration to solve challenging problems. Rather than using ϵ-greedy policy, it utilizes multiple Q-networks for exploration. One of the networks is randomly selected to make steps in a training episode, and the transition pairs 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 [22] showed that the BDQN agent could get out from sub-optimal policies in significantly fewer training episodes than the DQN agent. In the next Chapter, we provide results of numerical experiments that imply that BDQN is also more effective for the airline dynamic pricing environment.

[22] 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 archi-

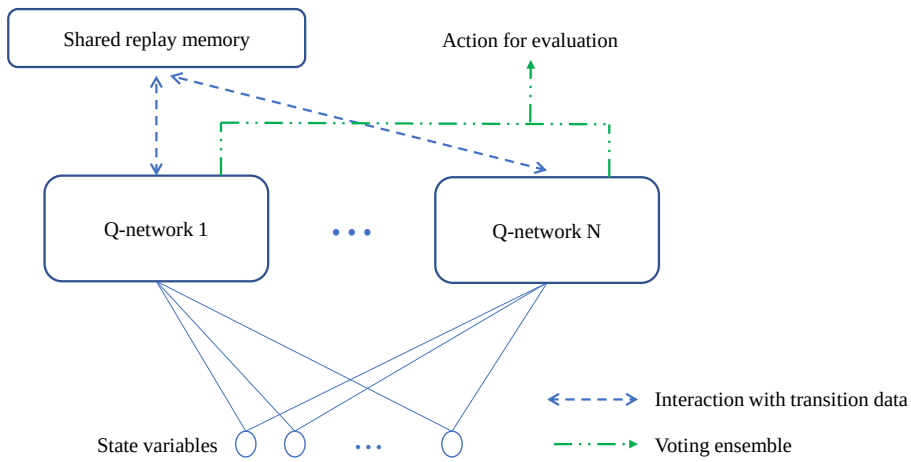
tecture because our input data is not a two-dimensional image frame. Each of the Q-networks is individually constructed in the same way as the DQN architecture we used. Therefore, none of them shares hidden layers in this thesis. Figure 3.1a shows the network architecture of DQN, and Figure 3.1b shows the architecture of BDQN which consists of multiple networks whose structures are the same as those shown in Figure 3.1a. We also present the procedure of BDQN in Algorithm 2.

Algorithm 2 BDQN algorithm for airline dynamic pricing

Initialize replay memory $\mathcal{D}$
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 $a_t =_a Q_{\phi_k}(s_t, a)$
    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 $\widehat{u_t} = (u_1, \ldots, u_N)$
    Store transition data $(s_t, a_t, r_t, s_{t+1}, \widehat{u_t})$ in replay memory $\mathcal{D}$
    Randomly select a minibatch of transitions $(s_j, a_j, r_j, s_{j+1}, \widehat{u_j})$ from $\mathcal{D}$
    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 $\widehat{u_j}$ is equal to 1, otherwise do not update
    end
    Set $t = t + 1$
  end
  Update target networks every E episodes
end



(a) DQN



(b) BDQN

Figure 3.1: Architecture of DQN and BDQN for airline dynamic pricing

3.3 Optimistic learning for decreasing cyclic policies

We use a variant of the optimistic learning for decreasing cyclic policies (OLD) algorithm proposed by [36] to evaluate RL algorithms for our problem. To the best of our knowledge, [36] 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.

Chapter 4

Numerical experiments

In this chapter, we present the results of numerical experiments. The first section demonstrates the effectiveness of the proposed MDP formulation in the presence of patient customers. In the second section, we compare the performance of algorithms for the environments where demand is non-stationary and inventory is insufficient. In the last section, we analyze the structures of pricing policies.

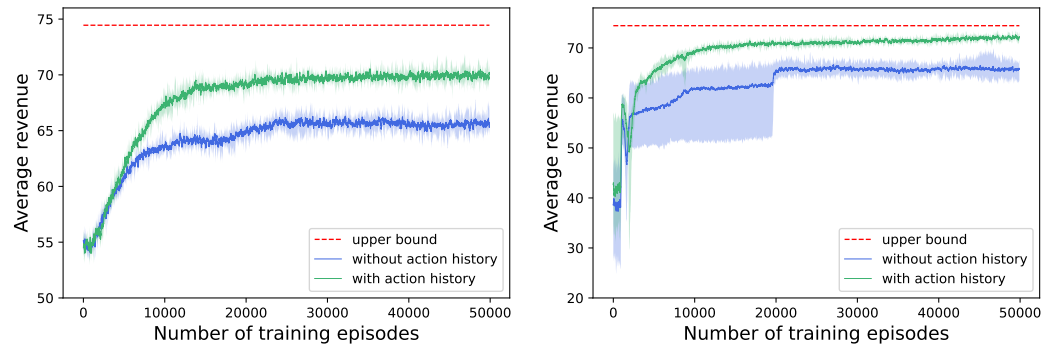
In implementing RL algorithms, we fix the value of ϵ as one and do not update network parameters for the initial 1,000 episodes, which is common practice in the RL community. After the 1,000 initialization episodes, we linearly decrease ϵ to 0.1. The terminal state mentioned in Algorithms 1 and 2 means 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. All experiments are conducted using a Python 3 with an Intel Core

i5-9400F and 16GB of RAM.

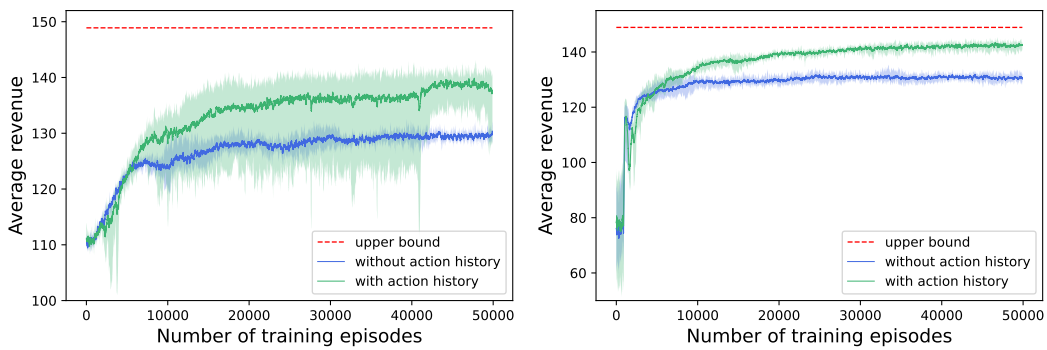
4.1 Comparison between MDP formulations in the presence of patient customers

We construct two different MDP formulations in this section. The only difference between them is the state s_t. We call an MDP that contains a history of actions $(a_{t-W}, \cdots, a_{t-1})$ 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 can not 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 $W = 11$. 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 section. To be more specific, initial patience levels for customers arriving in period t are uniformly distributed over $G_t = \{0, 1, \cdots, W\}$. For every $k \in G_t$ and $t \in \{1, \cdots, T\}$, reservation prices for customers in the group $\mathcal{G}_k^t$ are generated from the standard uniform distribution, and the number of customers in group $\mathcal{G}_k^t$ is set to be one. The total inventory is 300 when $T = 20$ and 500 when $T = 40$.



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



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

Figure 4.1: Average revenue of recent 100 episodes over five random seeds in training episodes

Figure 4.1 represents the average revenue over the recent 100 episodes for each training episode. 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 [18] 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

Table 4.1: Average revenue of evaluation episodes according to MDP formulation

Case	Customer type	T	Algorithm	Average revenue
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)

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.

The results in Figure 4.1 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 Figures 4.1a and 4.1c, 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 4.1 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 Figure 4.1. As in the training episodes, when the action history is included in the state, 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.

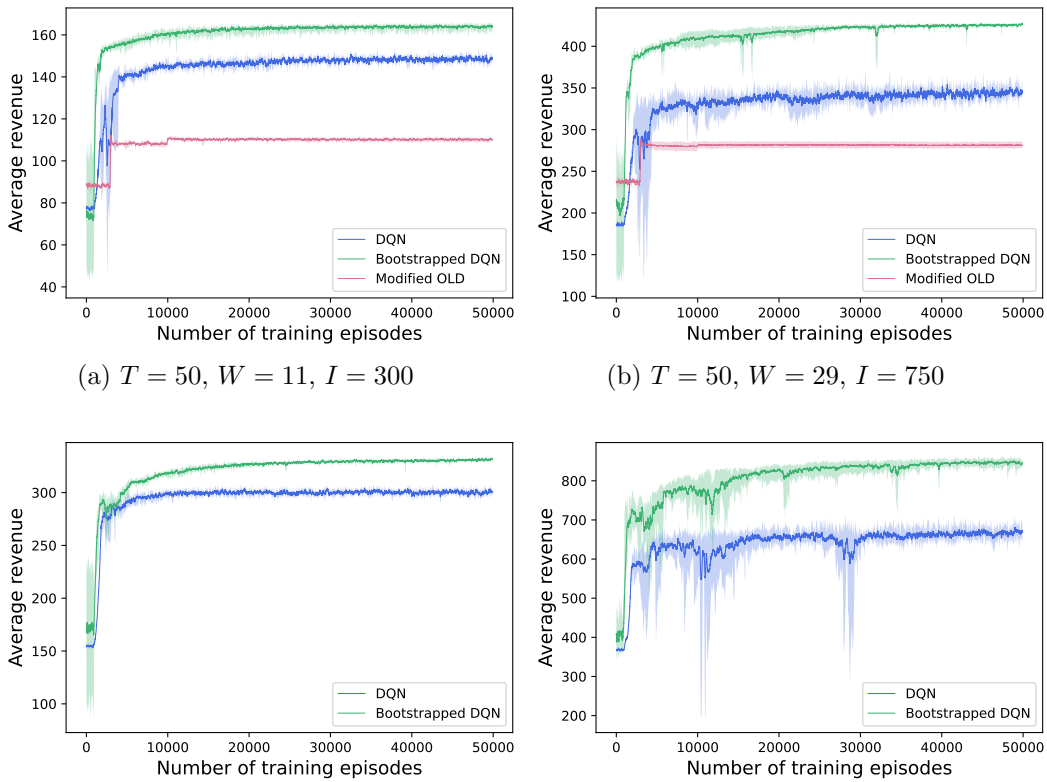
To verify that the improvement of performance on the RL algorithms truly comes from using action history, we additionally examine its effectiveness when there are no patient customers at all. Cases 5 and 6 represent the system where all customers are myopic (i.e., the patience levels of all customers 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 can not say that the satisfaction of the Markov property can improve the performances of the RL algorithms in our dynamic pricing problem. However, as Table 4.1 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.

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

In this section, 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 [15, 7, 31]. All elements that make up the MDP formulation are defined identically in the previous section. Both

leisure and business customers are assumed to arrive in the Poisson process. 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 I. To verify which algorithm is appropriate for this situation, we compare the performances of each algorithm for some instances. Figure 4.2 shows the learning curves of each algorithm where all elements in the figure are the same as in Figure 4.1. When $T = 100$, the modified OLD algorithm takes too much time to terminate 50,000 training episodes. Therefore, we do not present the learning curves of it in Figures 4.2c and 4.2d.

Figures 4.2a and 4.2b 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 can not 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 Figures 4.2b and 4.2d where the shaded area between the best and worst learning



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

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

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

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

Figure 4.2: Average revenue when the demand is non-stationary and the number of seats is limited

curves decreases as the number of training episodes increases.

As explained in the previous section, even though the average revenue of BDQN is higher than DQN in training episodes, we can not 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 1,000 evaluation episodes with 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 4.2 and Figure 4.3. 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.

Table 4.2: Average revenue of evaluation episodes over five random seeds

T	W	Algorithm	Mean	Maximum	Minimum	SD[a]
50	11	DQN	164.60	164.95	164.12	0.30
		BDQN	164.38	165.54	162.30	1.14
		Modified OLD	110.69	110.81	109.10	0.63
50	29	DQN	409.39	424.38	387.88	13.32
		BDQN	427.36	428.52	425.72	0.93
		Modified OLD	281.38	285.55	277.78	2.95
100	11	DQN	332.17	333.86	330.29	1.57
		BDQN	332.50	335.17	330.23	1.67
100	29	DQN	753.17	812.97	690.15	41.90
		BDQN	856.09	863.52	847.53	6.35

[1] SD: Standard deviation

Furthermore, we can figure out that BDQN gives more consistent revenue over multiple learning for the same problem compared to DQN from Figure 4.4. It shows distributions of total revenue per evaluation episode over five random seeds when $W = 29$. In Figure 4.4a, total revenue from seeds 1,2 and 4 have similar shapes of

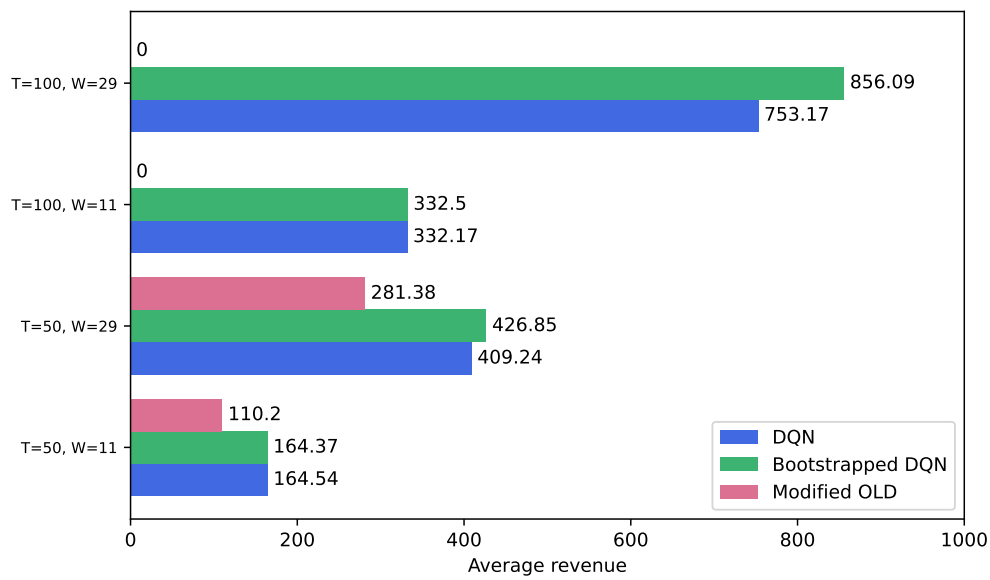
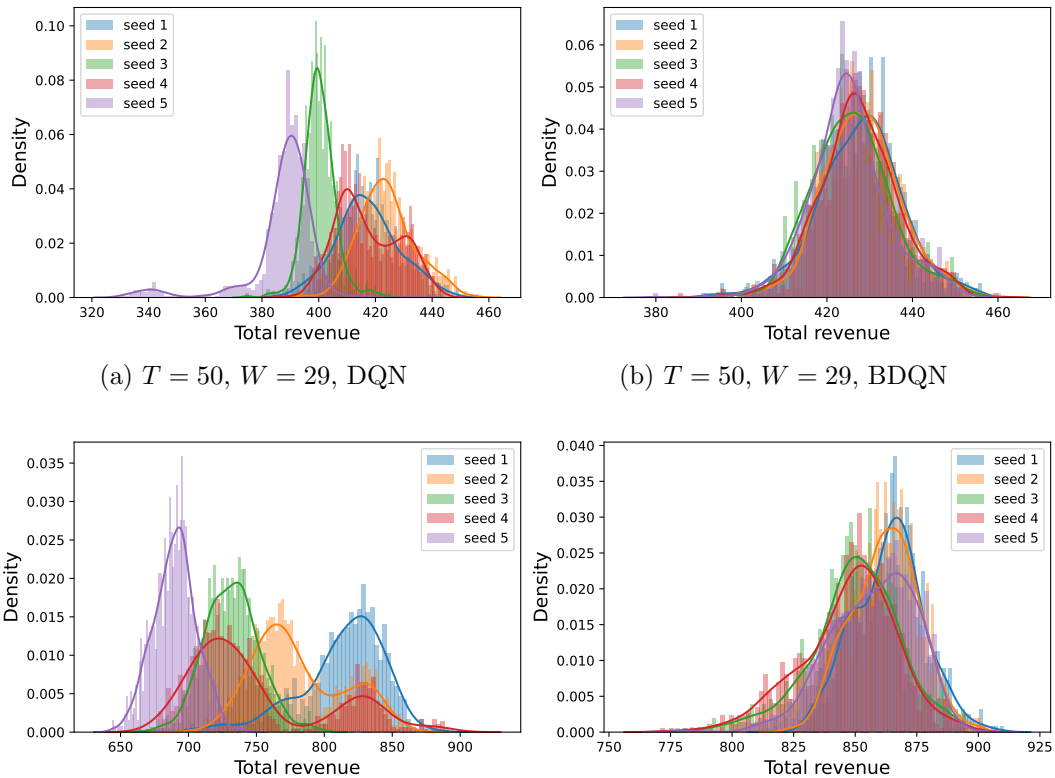


Figure 4.3: Mean of the average revenue in evaluation episodes over five random seeds

distributions but seed 3 and 5 show different shapes. In Figure 4.4c, 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 Figures 4.4b and 4.4d. 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. In addition, we compared the ratio of standard deviations with F-value to verify that the variance of the average revenue from DQN is higher than BDQN. When $W = 11$, there was no difference between the variances of DQN and

BDQN. However, when $W = 29$, we could observe that the variance of the average revenue from DQN is significantly higher than BDQN. Therefore, we can conclude that the BDQN algorithm might give the highest and stable revenue among the three pricing algorithms in the presence of patient and non-stationary customers with limited inventory.



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

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

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

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

Figure 4.4: Distributions of the total revenue in evaluation episodes over five random seeds

4.3 Structure of pricing policies from the BDQN algorithm

In this section, we analyze the structures of pricing policies calculated using the BDQN algorithm. 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 4.2, customers are segmented by leisure and business groups. Figure 4.5 shows trajectories of prices under non-stationary demand. In Figures 4.5a and 4.5b, 10,000 evaluation episodes are simulated to calculate average actions in each period, and from Figures 4.5c to 4.5f are examples randomly selected between those evaluation episodes. The first MDP formulation in Section 4.1 is used for the situation in which airlines construct pricing policies considering the effectiveness of patient customers, and the second MDP formulation in Section 4.1 is used to capture characteristics of policies when the airlines do not consider patient customers at all. In addition, Figure 4.6 shows how many seats remain and when selling periods are terminated according to each policy structure in Figures 4.5a and 4.5b.

From Figures 4.5a, 4.5b, 4.5e and 4.5f, 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 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 Figures 4.5a and 4.5b. 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 struc-

tures as shown in Figure 4.6. However, because higher average revenue is observed under the structure of green lines in Figure 4.5, we can conclude that it is a more appropriate structure for the non-stationary system with patient customers.

From comparing the green lines in Figures 4.5a and 4.5b, 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 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.

4.4 Non-stationary test for the distributions of reservation prices

In Sections 4.2 and 4.3, we set the arrival rates of the leisure and business customers to be changed as time passes. However, the distributions for reservation prices of the leisure and business customers are assumed not to be changed. We expect that the shift in the ratio of the two customer segments will make the distributions for reservation prices of all customers change as time passes. In this section, we present the results of statistical tests to verify that the average reservation prices of customers in Sections 4.2 and 4.3 are non-stationary. For each demand setting gen-

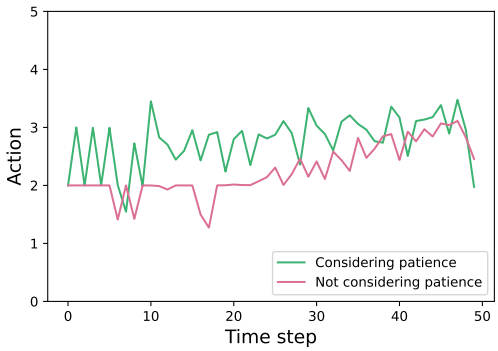
Table 4.3: Number of episodes with stationary demand

		Section 4.2		**Section 4.3**	
T	W	Total	New	Total	New
50	11	1	20	3,997	5,548
50	29	1	1	5,837	5,420
100	11	0	1		
100	29	33	0		

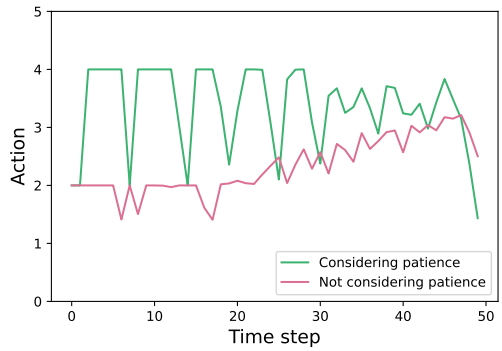
erated in those sections, we simulated 10,000 episodes with random policies. Then, we count the number of episodes in which the distributions of reservation prices turned out to be stationary using the Augmented Dickey-Fuller (ADF) test [12].

Table 4.3 shows the number of episodes that turned out to have stationary distributions of reservation prices among the 10,000 episodes at a significance level of 0.01. The first columns in each section indicate the number of episodes with stationary demand in terms of average reservation price for all existing customers in the system. The second columns are for the average reservation price for arriving customers in each period.

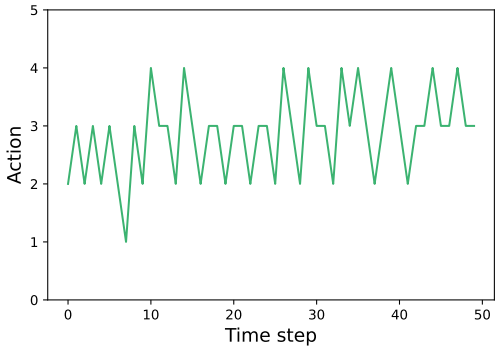
As shown in the first section in Table 4.3, most of the episodes have non-stationary distributions of reservation prices in experiments of Section 4.2. Because the experiments of Section 4.2 are designed to find out which algorithm is appropriate for highly non-stationary demand, these are consistent with our expectations. In experiments of Section 4.3, about half of episodes have stationary demand. They are designed to figure out the reasonable structure of policies in a realistic scale and variation of the demand in the airline industry. Therefore, we can say that the proportion of non-stationary demand is appropriate.



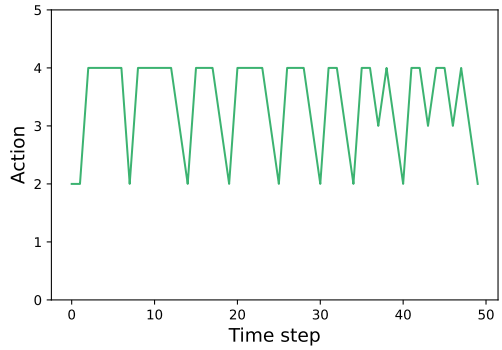
(a) Structure of the average price ($W = 11$)



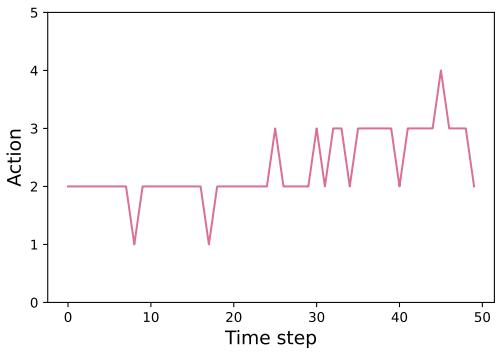
(b) Structure of the average price ($W = 29$)



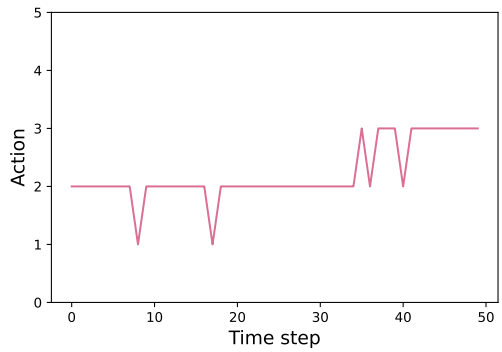
(c) One of policies considering patience ($W = 11$)



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

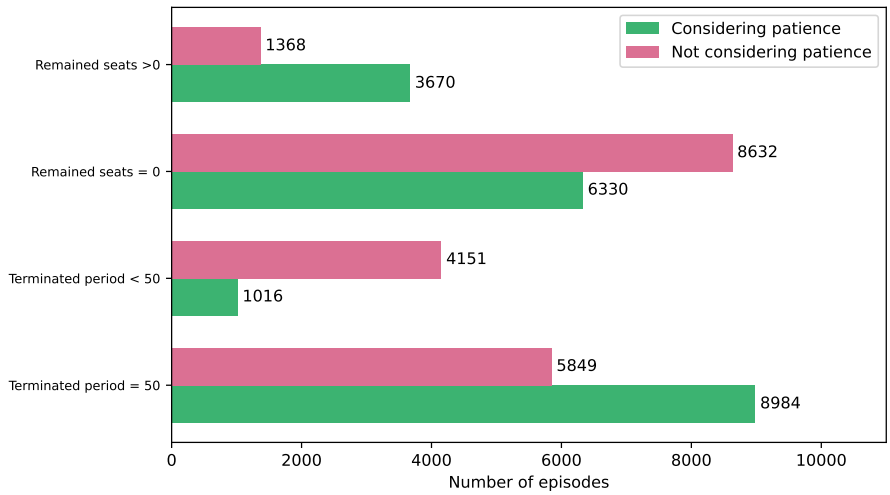


(e) One of policies that do not consider patience ($W = 11$)

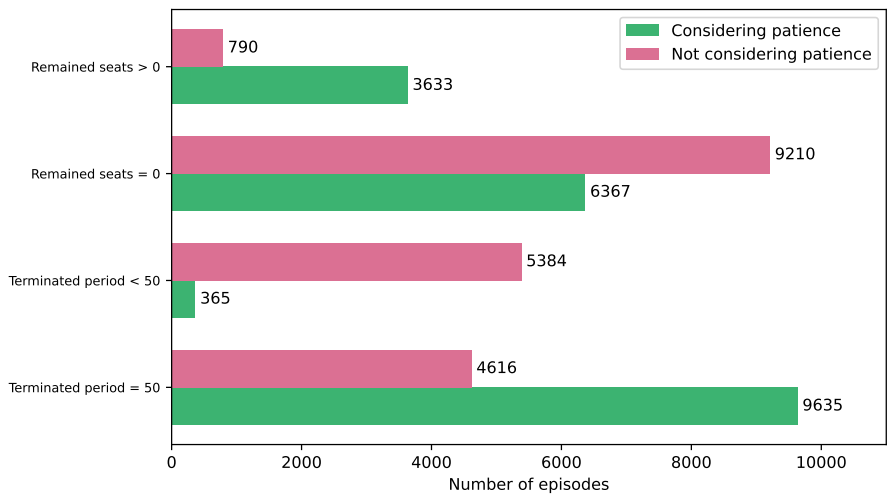


(f) One of policies that do not consider patience ($W = 29$)

Figure 4.5: Structures of pricing policies



(a) $W = 11$



(b) $W = 29$

Figure 4.6: Number of episodes grouped by terminated period and the number of remaining seats

Chapter 5

Conclusions

Patterns of purchasing behavior in the airline industry are becoming more complex. Customers do not leave the market immediately, even if they recognize that the current price is more expensive than their willingness to pay. They keep observing price changes and try to buy tickets at an acceptable price. Airlines get additional revenue if they consider these patterns in their pricing policies. The patient customer model is one of the models that capture the strategic behavior of demand. Some studies proposed algorithms to calculate the optimal policy in the presence of patient customers. Nevertheless, they assumed that sellers knew the distributions for reservation prices and patience levels of customers in advance. To the best of our knowledge, this thesis is the first who 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 the single-fare dynamic pricing problem in the airline industry. 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. In contrast to algorithms proposed by previous studies, they can solve

the problem without any assumptions on customer arrival rate and distributions of reservation prices and patience levels.

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 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. The mean of the average revenue over five random seeds from each DRL algorithm was higher than the modified OLD algorithm. However, the variance of DQN significantly increased in large-scale problems, whereas BDQN did not. This difference might come from the approach of exploration, which means that DQN failed to learn appropriate policies for the airline industry where 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.

The dynamic pricing framework we proposed in this thesis has some limitations. 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 should be considered in 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. In addition, we assumed that the arrival rates of business and leisure customers increase or decrease linearly. Therefore, we cannot say that the current settings of the deep reinforcement learning algorithms will calculate reasonable pricing policies for a more complicated structure of demand. However, as mentioned before, model-free reinforcement algorithms can be utilized without any assumptions of specific demand structures. We can extend the framework for various types of demand by changing parameters or structures of Q-networks.

Bibliography

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer, *Finite-time analysis of the multiarmed bandit problem*, Machine Learning, 47 (2002), pp. 235–256.

[2] Y. Aviv and A. Pazgal, *Optimal pricing of seasonal products in the presence of forward-looking consumers*, Manufacturing & Service Operations Management, 10 (2008), pp. 339–359.

[3] N. Aydin, Ş. İ. Birbil, J. Frenk, and N. Noyan, *Single-leg airline revenue management with overbooking*, Transportation Science, 47 (2013), pp. 560–583.

[4] R. Bautista-Montesano, R. Galluzzi, K. Ruan, Y. Fu, and X. Di, *Autonomous navigation at unsignalized intersections: A coupled reinforcement learning and model predictive control approach*, Transportation Research Part C: Emerging Technologies, 139 (2022), p. 103662.

[5] D. Bertsimas and S. De Boer, *Simulation-based booking limits for airline revenue management*, Operations Research, 53 (2005), pp. 90–106.

[6] O. Besbes and I. Lobel, *Intertemporal price discrimination: Structure and computation of optimal policies*, Management Science, 61 (2015), pp. 92–110.

[7] N. Bondoux, A. Q. Nguyen, T. Fiig, and R. Acuna-Agost, *Reinforcement learning applied to airline revenue management*, Journal of Revenue and Pricing Management, 19 (2020), pp. 332–348.

[8] P. Cao, M. Fan, and K. Liu, *Optimal dynamic pricing problem considering patient and impatient customers' purchasing behaviour*, International Journal of Production Research, 53 (2015), pp. 6719–6735.

[9] F. Caro and J. Gallien, *Clearance pricing optimization for a fast-fashion retailer*, Operations Research, 60 (2012), pp. 1404–1422.

[10] S. Dasu and C. Tong, *Dynamic pricing when consumers are strategic: Analysis of posted and contingent pricing schemes*, European Journal of Operational Research, 204 (2010), pp. 662–671.

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

[12] D. A. Dickey and W. A. Fuller, *Distribution of the estimators for autoregressive time series with a unit root*, Journal of the American Statistical Association, 74 (1979), pp. 427–431.

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

[14] R. J. Lawhead and A. Gosavi, *A bounded actor–critic reinforcement learning algorithm applied to airline revenue management*, Engineering Applications of Artificial Intelligence, 82 (2019), pp. 252–262.

[15] J. Li, N. Granados, and S. Netessine, *Are consumers strategic? structural estimation from the air-travel industry*, Management Science, 60 (2014), pp. 2114–2137.

[16] J. Li, L. Yao, X. Xu, B. Cheng, and J. Ren, *Deep reinforcement learning for pedestrian collision avoidance and human-machine cooperative driving*, Information Sciences, 532 (2020), pp. 110–124.

[17] Y. Liu and W. L. Cooper, *Optimal dynamic pricing with patient customers*, Operations Research, 63 (2015), pp. 1307–1319.

[18] I. Lobel, *Dynamic pricing with heterogeneous patience levels*, Operations Research, 68 (2020), pp. 1038–1046.

[19] C. Mao and Z. Shen, *A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network*, Transportation Research Part C: Emerging Technologies, 93 (2018), pp. 179–197.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, *Playing atari with deep reinforcement learning*, arXiv preprint arXiv:1312.5602, (2013).

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostro-

vski, et al., *Human-level control through deep reinforcement learning*, Nature, 518 (2015), pp. 529–533.

[22] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, *Deep exploration via bootstrapped dqn*, Advances in Neural Information Processing Systems, 29 (2016).

[23] D. F. Otero and R. Akhavan-Tabatabaei, *A stochastic dynamic pricing model for the multiclass problems in the airline industry*, European Journal of Operational Research, 242 (2015), pp. 188–200.

[24] V. Pandey, E. Wang, and S. D. Boyles, *Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations*, Transportation Research Part C: Emerging Technologies, 119 (2020), p. 102715.

[25] R. Rana and F. S. Oliveira, *Real-time dynamic pricing in a non-stationary environment using model-free reinforcement learning*, Omega, 47 (2014), pp. 116–126.

[26] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, *Prioritized experience replay*, arXiv preprint arXiv:1511.05952, (2015).

[27] D.-W. Seo, K. Chang, T. Cheong, and J.-G. Baek, *A reinforcement learning approach to distribution-free capacity allocation for sea cargo revenue management*, Information Sciences, 571 (2021), pp. 623–648.

[28] Z. Shou, X. Chen, Y. Fu, and X. Di, *Multi-agent reinforcement learning for markov routing games: A new modeling paradigm for dynamic traffic as-*

signment, Transportation Research Part C: Emerging Technologies, 137 (2022), p. 103560.

[29] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[30] H. Van Hasselt, A. Guez, and D. Silver, *Deep reinforcement learning with double q-learning*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, 2016.

[31] R. R. Varella, J. Frazão, and A. V. Oliveira, *Dynamic pricing and market segmentation responses to low-cost carrier entry*, Transportation Research Part E: Logistics and Transportation Review, 98 (2017), pp. 151–170.

[32] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, *Dueling network architectures for deep reinforcement learning*, in International Conference on Machine Learning, PMLR, 2016, pp. 1995–2003.

[33] C. J. C. H. Watkins, *Learning from delayed rewards*, PhD thesis, King's College, Cambridge, UK, 1989.

[34] M. D. Wittman, *Dynamic pricing mechanisms for airline revenue management: theory, heuristics, and implications*, PhD thesis, Massachusetts Institute of Technology, 2018.

[35] C. Yang, Y. Feng, and A. Whinston, *Dynamic pricing and information disclosure for fresh produce: An artificial intelligence approach*, Production and Operations Management, 31 (2022), pp. 155–171.

[36] H. Zhang and S. Jasin, *Online learning and optimization of (some) cyclic pricing policies in the presence of patient customers*, Manufacturing & Service Operations Management, 24 (2022), pp. 1165–1182.

국문초록

본 연구에서는 전략적 소비자가 존재하는 시장에서 항공사 동적 가격 결정 문제를 다루었다. 최근 소비자들은 항공사에서 동적 가격 정책을 시행하는 것을 인지하고 있기 때문에, 그들의 지불 용의보다 낮은 가격을 지불하기 위해 전략적으로 행동한다. 이러한 소비자 특성을 고려하여, 본 연구에서는 과거에 제시된 가격 기록을 상태 변수로 포함하는 마르코프 의사결정 과정 모델을 제안하였다. 이 때 고객 특성에 대한 확률 분포들은 사전에 알려져 있지 않다고 가정하였다. 문제 해결을 위해 심층 강화학습 방법론이 활용되었으며, 알고리즘 별 비교를 통해 전략적이고 동적인 수요 하에서 가장 적절한 가격 구조를 도출하는 알고리즘을 제시하였다. 또한 해당 가격 구조를 분석하여 전략적 수요로부터 추가적인 수익을 발생시키기 위한 경영적 통찰력을 제공하고자 하였다.

주요어: 항공사 수익관리, 동적 가격 정책, 강화학습, 심층 Q-신경망, 부트스트랩 기반 심층 Q-신경망, 전략적 소비자

학번: 2021-23948