

ORIGINAL RESEARCH

Dynamic pricing of differentiated products with incomplete information based on reinforcement learning

Cheng Wang^{1,2}  | Senbing Cui¹ | Runhua Wu¹ | Ziteng Wang³

¹College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, China

²Taizhou Research Institute, Zhejiang University of Technology, Taizhou, China

³Department of Industrial and Systems Engineering, Northern Illinois University, DeKalb, Illinois, USA

Correspondence

Cheng Wang, College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023, China.

Email: cwang@zjut.edu.cn

Funding information

Natural Science Foundation of Zhejiang Province, Grant/Award Numbers: LY18G010019, LY20G010011; National Natural Science Foundation of China, Grant/Award Number: L1924063; Humanities and Social Science Fund of Ministry of Education of China, Grant/Award Number: 19YJCZH020

Abstract

With the rapid development of the social economy, consumer demand is evolving towards diversification. To satisfy market demand, enterprises tend to improve competitiveness by providing differentiated products. How to price differentiated products becomes a hot topic. Traditionally, customers' preferences are assumed to be independent and identically distributed. With a known distribution, companies can easily make pricing decisions for differentiated products. However, such an assumption may be invalid in practice, especially for rapidly updating products. In this paper, a dynamic pricing policy for differentiated products with incomplete information is developed. An adaptive multi-armed bandit algorithm based on reinforcement learning is proposed to balance exploration and exploitation. Numerical examples show that the frequency of price adjustment affects the total profit significantly. Specifically, the more chances to adjust the price, the higher the total profit. Furthermore, experiments show that the dynamic pricing policy proposed in this paper outperforms other algorithms, such as Softmax and UCB1.

KEYWORDS

customer preference, differentiated product, dynamic pricing, incomplete information, reinforcement learning

1 | INTRODUCTION

With the rapid development of the social economy, consumption is constantly transforming and upgrading, tending to be personalised and diversified. To adapt to the challenge of the market environment, many enterprises adhere to the principle of diversified production, developing differentiated products vigorously. Moreover, the development of the Internet has further promoted the diversification of demands. According to the long tail effect [1], the market share of products with small demands and great variety can be comparable to the market share of products with large demands and less variety. Under this backdrop, customers who treasure a higher degree of uniqueness are willing to pay more for a personalised product than the standard one [2], and enterprises in the process of providing differentiated products lead to an increase in costs. How to price reasonably for differentiated products is an operation management problem worth studying.

From the customers' point of view, they comprehensively consider their preferences and the prices of differentiated products and make reasonable purchasing choices. From the perspective of companies, considering the purchasing behaviour of customers and production costs, what they are concerned about is how to set reasonable prices to maximise profit. If companies know the specific purchasing preference of customers, they can offer the right prices through optimisation methods such as dynamic programming [3] and mixed-integer programme [4]. Due to the asymmetry of information, customers' preferences are usually assumed to be independent and identically distributed random variables. If the company knows the distribution, the optimal prices can be designed according to contract theory [5] to maximise the expected profit. However, these assumptions sometimes fail to hold in real life, especially the rapidly iterative and updated differentiated products. It is hard for companies to grasp customers' preferences and make certain

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2022 The Authors. *IET Collaborative Intelligent Manufacturing* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

what type of distribution is, which we call incomplete information [6].

With digitalisation, taking advantage of the data-driven approach to make price decisions is an effective way. By collecting transaction information, the company tries and learns pricing decisions about different customers, understands and predicts customers' behaviours based on data analysis, to lay a judgement foundation for future price decisions. The rapid development of the Internet and E-commerce makes it possible technically. As shown in Figure 1, DJI's 'Mavic 2' series of unmanned aerial vehicles with different quality levels are sold at dynamic prices on JD.COM. By acquiring the historical price information of the Pro and the Zoom versions in real-time, the platform makes it easy to monitor price movements in the market and acquire customers' purchase results.

When a new differentiated product comes out, there are no historical records of the selling price information. It means that we do not know the specific customer preference and preference distributions. With incomplete information, how to take full advantage of data to achieve dynamic pricing of differentiated products is a thorny problem.

We contribute to the literature via the following ways. *First*, we propose an adaptive algorithm to deal with pricing decisions with incomplete information based on reinforcement learning. We design a learning process and record customers' purchase results as well as the number of times the price has been selected. We are the first to combine incomplete information with differentiated products and dynamic pricing issues. *Second*, to explore the factors affecting the total profit, we attempt different pricing strategies and adjustment frequencies according to the number of customers. We found a few factors that affect average returns. *Third*, we judge the possibility of the current price application and make dynamic price adjustments promptly based on a balance trick. By comparing different algorithms, we demonstrate our algorithm's superiority in the dynamic pricing problem of differentiated products.

The remainder of the paper is organised as follows: We review the relevant literature in Section 2. We make some

problem descriptions in Section 3. In Section 4, we describe the learning model and assumptions. In Section 5, combining reinforcement learning, we propose an adaptive algorithm to solve the dynamic pricing of differentiated products. Numerical studies are conducted in Section 6. Finally, we conclude the paper and discuss some future work in Section 7.

2 | LITERATURE REVIEW

2.1 | Dynamic pricing of differentiated products

Dynamic pricing is an effective business strategy to adjust the price of products in time and allocate the right services to the right customers [7]. In recent years, the problem of dynamic pricing has drawn much attention in different fields, such as inventory management [8], smart grid [9] and distribution network [10].

Many scholars have studied dynamic pricing from the term of product differentiation. Akcay et al. [11] adopted stochastic dynamic programming to study the joint dynamic pricing of multi-level products and analysed the nature of the optimal price under heterogeneous product structure. Liu and Zhang [12] studied the dynamic pricing problem of products with different levels in a competitive environment and proved that price skimming arises as to the unique pure-strategy Markov perfect equilibrium. Facing the uncertainty about the cost quality of vertically differentiated products, Keskin and Birge [13] adjusted the pricing by dynamic learning and made noisy observations on the underlying cost model through customers' purchasing decisions. Then, they characterised how optimal product differentiation depends on the 'informativeness' of quality choices. Chen et al. [14] presented a model for designing two-stage dynamic pricing strategies by considering a centralised system. Then they derived equilibrium prices and optimal pricing strategies for the seller under markdown pricing policy using the equilibrium theory and backward induction method. The results indicated a non-monotonicity

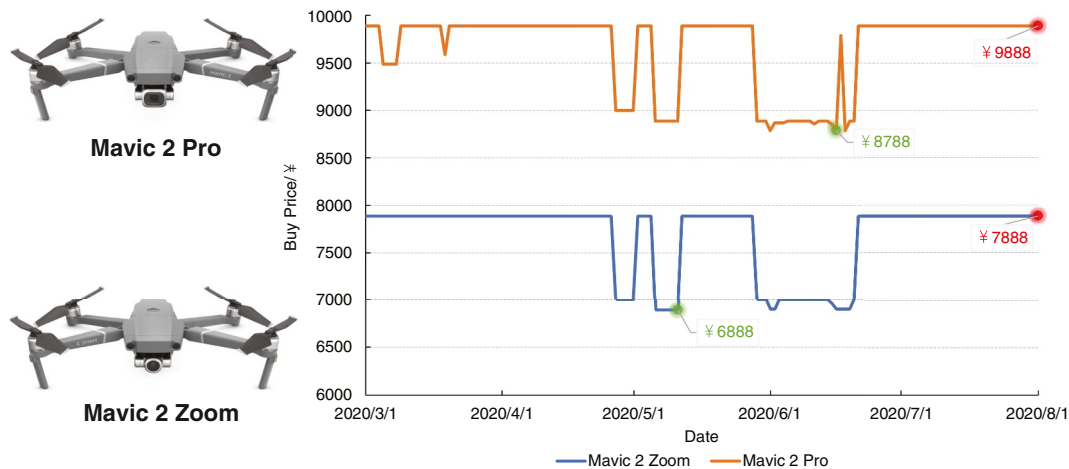


FIGURE 1 Dynamic historical price comparison of different unmanned aerial vehicles (UAV) types

concerning the coexistence of consumer strategic behaviour and reference price effect. Laussel et al. [15] find that there is a Markov perfect equilibrium for the behaviour of consumers delaying purchases and waiting for a lower price. By providing static equilibrium time (which can obtain full promised profits) from suppliers, suppliers' profits will expand. For the pricing problem of online differentiated products, Cohen et al. [16] propose to use feature vectors to describe products and use binary search method on Lowner–John ellipsoids set to determine prices based on historical sales prices. Experiments show that the worst-case regret value of this algorithm is quadratic in the dimensionality of the feature space and logarithmic in the time horizon. Considering that consumers adopt the widely used Multinomial Logit model to select differentiated products, Javanmard et al. [17] provide a multi-dimensional feature pricing strategy (M3P) that achieves a logarithmic complexity in the T-period under heterogeneous price sensitivity for products with features of dimension d .

These studies solved the dynamic pricing problem based on product differentiation and customer choices. In their work, they assumed in advance that the distribution of customer preferences and demand curves are known and then calculated the probabilities of customers' choices based on prior knowledge so that dynamically adjusting the corresponding pricing strategy. However, the company always lacks some information about customers for a variety of reasons in real life. It can neither grasp the distribution of consumer preferences of all potential customers nor directly judge the specific choice of a customer. Unreasonable pricing may prevent customers from getting what they want, and the failure of transactions may result in the loss of the enthusiasm of enterprises to provide differentiated products. Therefore, what we consider is how to dynamically adjust the price of differentiated products with incomplete information to maximise the expected profit.

2.2 | Reinforcement learning

Reinforcement learning is an area of machine learning inspired by behaviourist psychology. It mainly focusses on how the agent takes different actions in the environment to maximise the cumulative reward. With the rapid development of artificial intelligence, a great number of breakthroughs in reinforcement learning have been reported for nearly two decades, especially like deep Q-network in Atari and AlphaGo [18]. And many research fields have adopted reinforcement learning to solve decision-making problems, such as electricity allocation [19], intelligent operation, and maintenance [20] as well as engineering manufacturing [21].

A typical example of studying decision-making in reinforcement learning is the multi-armed bandit problem (MAB). It is originally a stateless mathematical model extracted from the scene of a multi-arm slot machine in a casino [22]. Pulling one arm of the slot machine may generate positive revenue. A gambler does not know which arm he chooses every time to maximise profits within a limited number of times. Because the probability of getting a reward after pulling each arm is

different and he does not know what the probability distribution is. The difficulty of MAB problems lies in how to balance the exploration–exploitation dilemma. For other arms with fewer attempts or unknown, a certain number of trial opportunities must be allocated to avoid missing higher-yielding options (exploration). For known arms with a higher probability of getting rewards, more attempts should be made to obtain cumulative rewards (exploitation). At the same time, more exploration also means higher opportunity risks.

Multi-armed bandit problem overcomes the exploration and exploitation dilemma of sequential decision-making under uncertainty, and it solves most online decision-making problems in real life.

For instance, Huo and Fu [23] incorporated risk awareness into the classic MAB setup and introduced a new portfolio construction algorithm. By filtering assets according to the topological structure of the financial market, the optimal multi-armed bandit strategy was combined with consistent risk measurement minimisation to achieve a balance between risk and return. Durand et al. [24] aimed to design an adaptive allocation strategy based on the Contextual bandit framework to explore promising treatment methods by allocating more samples, thereby improving the efficiency of data collection. Liu et al. [25] proposed a contextual multi-armed bandit model with a non-linear reward function that used a distributed representation of text for online response selection. In learning the bandit, they proposed a customised Thompson sampling method that was applied to a polynomial feature space in approximating the reward. Ding et al. [26] studied anomaly detection in interactive environments by allowing the system to actively communicate with human experts and perform a limited number of queries on real anomalies. Bouneffouf and Rish [27] proposed to study the freshness of user content through the MAB problem. They introduced a Thompson sampling algorithm based on freshness perception, which manages recommendations for new documents according to the user's risk of the situation.

MAB also applies to dynamic pricing. Companies can conduct price experiments (changing prices) to understand demand and maximise long-term profits. Misra et al. [28] proposed an experimental policy for dynamic pricing. In their work, they derived an extended bandit algorithm that balances the learning of immediate profits and future profits, combining economy theory. They assumed that the company has incomplete information about all products' demand curves. The likelihood of buying different products (at a certain price) is inferred from the observable heterogeneity of different customers, and the range of pricing is determined by fitting the demand curve of the product through the purchase results.

2.3 | Contributions of this work

Our research study focusses on dynamic pricing about product differentiation, which is customer-oriented and takes into account the benefits of customers and providers. We recognise that customers with heterogeneity have different quality

demands for the same basic products, so the rationality of price segmentation should be taken into consideration. In the model, we do not know the specific customers' preferences and their distribution, nor can we determine the product price according to the individual demand of different customers. Meanwhile, we distinguish some of the contributions work in the literature on solving dynamic pricing problems as shown in Table 1.

In our work, combining the MAB problem in reinforcement learning, we embrace dynamic pricing about product differentiation as a sequential decision-making problem and design a training and learning process based on transaction results. We regard different pricing as actions and the final reward as the result of taking different actions. In order to obtain a larger long-term profit, we find a good balance between exploration (for example, trying new prices) and exploitation (choosing the price that has been used) through changing dynamic parameters adaptively. Even under incomplete information, we can figure out the optimal strategy to obtain the maximum expected profit within limited resources.

3 | PROBLEM DESCRIPTION

This section describes the basic background of the problem. Without loss of generality, we assume that a monopoly provider can provide a set of differentiated products (a basic product has additional attributes with different quality levels) with potential customers in the market. Each customer has an independent purchasing demand, which is determined by the customer's consumption preference v . All customers' preferences are assumed to be independent and identically distributed, following a kind of distribution. If the provider knows the distributions, we argue that he has complete information. Otherwise, he has incomplete information.

Generally, the better the product quality, the higher the production cost and the price. We assume that all customers are rational. They buy any product according to their willingness to pay, which we use utility value V to indicate. In other

words, when comparing products of different quality levels, the customer will choose a product with the greatest utility value by taking into account quality and price. Therefore, for the provider, how to set dynamic prices for different customers in order to maximise expected profit is an important problem, which can be divided into two situations.

3.1 | Under complete information

Information resources greatly influence the decision-making process. If the provider has complete information in advance, including the distribution of consumption preferences v , then the customers' payment function will be fully understood. We can model interactions between the provider and customers as a 1-leader, N-followers Stackelberg game [29]. The schematic diagram of the model is shown in Figure 2.

The provider is a leader and sets price combinations for differentiated products. Customers are followers who observe the price combination and make choices for buying or not buying. If the leader knows the distribution of consumption preferences v , the probability that a follower buys a kind of quality level product could be calculated through the function relationship between price and utility. The provider has the first-mover advantage and determines the profit-maximising price based on knowing the customer's utility V about quality and price. Then, the customer makes his decision based on the given price for deriving his maximum utility.

The model can be solved by backward induction [30]. The leader considers what the optimal response of the follower is, then he picks a price that maximises the payoff, and anticipates the response of the follower. The follower actually observes the strategy and chooses the expected product as a response in equilibrium. However, without some information similar to individual demand or the distribution of customer consumption preferences, it is difficult for providers to make some judgements, which may result in selecting disadvantaged strategies in the game.

	Dynamic pricing	MAB	Game theory	Incomplete information	Product differentiation
Liu and Zhang [12]			✓		✓
Akçay et al. [11]	✓				✓
Keskin and Birge [13]	✓				✓
Misra et al. [28]		✓		✓	
Our work	✓	✓		✓	✓

TABLE 1 Contributions to pricing in different literature



FIGURE 2 1-leader and N-followers Stackelberg game under complete information

3.2 | Under incomplete information

Moreover, we consider the other situation. Under incomplete information, the strategy to find a Nash equilibrium between two players may not work. We put forward another thought as shown in Figure 3.

When a potential customer arrives, the provider sets a price combination. The customer decides to buy or not to buy the offered product after first judging whether it satisfies his own utility. If he does not buy, it means that the price may not be suitable, and the provider should try another new price combination. On the contrary, the customer will rationally choose the product with the greatest utility value. The provider has a belief that he can estimate the probability of the customer's successful purchase by recording the purchase results and update it about the expected profit. Whether to update the current price is involved in the exploration–exploitation dilemma, which will be mentioned in Section 5. Such a price experiment does not need to know what the customers' consumption preference is. After many price experiments, we can obtain the maximum profit as much as possible by training and learning pricing strategies.

Similar to the MAB problem, what we do is balancing the exploration–exploitation dilemma to solve the dynamic pricing problem with incomplete information. Our goal is to find the optimal prices within limited experiments or time on the premise of obtaining maximum profit.

4 | MODELLING AND ASSUMPTIONS

Consider a monopolist provider, offering various differentiated products to potential customers. We suppose that the provider has been able to produce the highest quality product. He provides compatible products with different quality levels that are denoted as q_k , $k = 1, 2, \dots, K$. Compatibility means that high-quality products have more attributes than low-quality products. K presents the maximum number of quality levels that the provider can provide to customers. We let $q_1 > q_2 > \dots > q_K > 0$, $q_H = q_1$, $q_L = q_K$, where q_L is the lowest quality level that has the least subset of product features to meet the basic requirements

of all customers, and q_H is the highest quality level. We let the corresponding marginal cost of different quality levels as $c_k > 0$, $k = 1, 2, \dots, K$. To simplify the model, according to the relevant literature [31], the marginal cost function here is expressed as linearly increasing: $c_k = c_0 + \alpha q_k$, $\alpha > 0$, $k = 1, 2, \dots, K$. c_0 represents the fixed cost of a basic product, and the marginal cost of the product increases with the raising of the quality level.

Suppose that there are M potential customers arriving with equal interval time throughout the transaction, and each customer can only purchase one product at most. Confronting customers who are independent and identically distributed with regard to the consumption preference, what the provider should do first is pricing for them. There exists a set of prices $\mathbf{p} = \{p_1, p_2, \dots, p_K\} | p_1 \geq p_2 \geq \dots \geq p_K \geq 0$, $p_k \in \mathbb{R}^+$, where p_k represents the price of products with different quality levels. We are going to price all of customers in terms of $\mathbf{p} = (p_1, p_2, \dots, p_i, \dots, p_M)$, where $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iK}) \in P$ presents the provider's pricing for the i -th customer.

We argue that all customers are rational, and they will compare carefully the utility value of products with different quality levels. Let $x_{ik} \in \{0, 1\}$ represent the purchase decision of customer i , where $x_{ik} = 0$ means that customer i does not purchase product k ; otherwise, $x_{ik} = 1$. Combining customer's consumption preference v , we have the i -th customer's utility function,

$$V_i(\mathbf{x}_i, \mathbf{p}_i) = \sum_{k=1}^K (v_i q_k - p_{ik}) x_{ik} + \beta_i \mathbb{1}_{\{\sum_{k=1}^K x_{ik}=1\}} \quad (1)$$

where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iK})$ represents the decision of the customer i , β_i presents the external addition of white Gaussian noise, which is the estimation error of the model, and $\mathbb{1}_{\{\cdot\}}$ is the indicator function, which equals 1 if the condition is satisfied, otherwise, 0.

As for the provider's profit W , it depends on decision variables $\mathbf{x}$ and $\mathbf{p}$, where $\mathbf{x} = (x_1, x_2, \dots, x_i, \dots, x_M)$ represents the decisions of all customers. In order to achieve the maximum cumulative profit, we consider how each customer i makes a choice when offered the price $\mathbf{p}_i$. Thus, the provider's profit function is given by

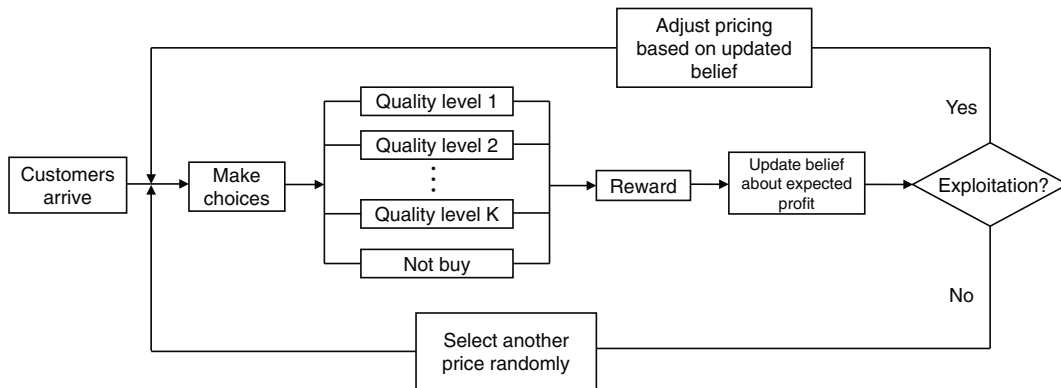


FIGURE 3 Dynamic pricing under incomplete information

$$W(\mathbf{x}, \mathbf{p}) = \sum_{i=1}^M \sum_{k=1}^K (p_{ik} - c_k) x_{ik} \quad (2)$$

Therefore, a bilevel programming model could be formulated as follows.

$$\left\{ \begin{array}{l} \max_{\mathbf{p}} W(\mathbf{x}, \mathbf{p}) = \sum_{i=1}^M \sum_{k=1}^K (p_{ik} - c_k) x_{ik} \\ \text{subject to} \quad p_{ik} \geq c_{ik} \quad (k = 1, 2, \dots, K, i = 1, 2, \dots, M) \\ p_{ik} \geq p_{i,k+1} \quad (k = 1, 2, \dots, K-1, i = 1, 2, \dots, M) \\ \text{where } \mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iK}), i = 1, 2, \dots, M \text{ solves} \\ \left\{ \begin{array}{l} \max_{\mathbf{x}_i} V_i(\mathbf{x}_i, \mathbf{p}_i) = \sum_{k=1}^K (v_i q_k - p_{ik}) x_{ik} + \beta_i \mathbb{1}_{\{\sum_{k=1}^K x_{ik}=1\}} \\ \text{subject to} \quad \sum_{k=1}^K x_{ik} \leq 1 \\ x_{ik} \in \{0, 1\} \quad (k = 1, 2, \dots, K) \end{array} \right. \end{array} \right. \quad (3)$$

At the first level, the provider maximises the total profit gained by deciding different prices (the decision variable $\mathbf{p}$) and offers eligible products for all potential customers. By the first constraint, the price of any quality level product should be higher than its marginal cost. A high-quality level product should be priced higher than the low-quality level one by the second constraint. The customer's strategy about whether to purchase a certain product k or not relies on the utility $V_i(\mathbf{x}_i, \mathbf{p}_i)$, which is determined by $\mathbf{x}_i$ and $\mathbf{p}_i$.

At the second level, each customer makes a decision on buying a specific quality for gaining maximal utility. By the first constraint, it means that a customer can only buy one product at most.

Therefore, the pricing problem of differentiated products could be regarded as a multiple decision-making problem [32].

Based on the above assumptions, one of the difficulties in optimising the objective function is how to accurately estimate the utility and selection of customers without knowing the distribution of consumer preference v . Generally, strong assumptions were given in the previous relevant work. Under complete information, the provider knows the distribution of

consumption preference and demand, so as to calculate the probability that customer i buys the k -th quality level product. However, our study highlights how this objective function that relies on consumer preference can be solved in the presence of incomplete information.

As shown in Figure 4, the provider offers the price $\mathbf{p}_{i-1}$ to the customer $i-1$ who has an unknown preference v_{i-1} . B_{i-1} represents the belief that the customer $i-1$ successfully buying any product at the current price. The customer is rational and makes purchasing decision with maximum utility. Therefore, whether the customer accepts price $\mathbf{p}_{i-1}$ or not, there will be a profit denoted as W_{i-1} . When the next customer i arrives, B_{i-1} is updated to B_i . It reflects the dynamic nature of time series that B_i depends on not only $\mathbf{p}_{i-1}$ but also W_{i-1} and B_{i-1} .

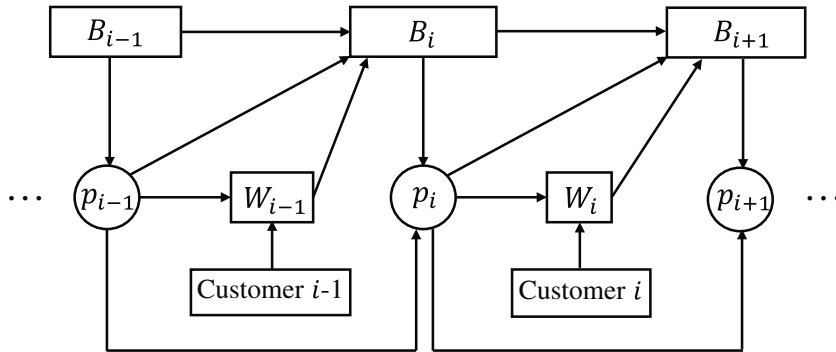


FIGURE 4 The dynamic pricing process over time series

5 | DYNAMIC PRICING POLICY FOR DIFFERENTIATED PRODUCTS

This section discusses solutions to the programming (3), where v_i are considered as independent identically distributed random variables with unknown distribution. A dynamic pricing policy is designed according to reinforcement learning.

As mentioned in Section 2, we regard a set of price combinations as alternative actions, which is equivalent to a set of arms for a slot machine. The provider provides a price combination for a potential customer by adopting different pricing strategies, and the positive profit generated thereafter is equivalent to the reward that can be obtained by pulling down an arm of the slot machine. For each round of selection, the main difficulty we challenge is how to balance the exploration–exploitation dilemma.

1. *Exploration*: Exploration aims to explore the search space more frequently and find diverse solutions. Here, we select other possible price combinations to get more information about the expected profit.
2. *Exploitation*: Exploitation uses known local information in the search process to obtain better solutions. Here, we select the known price combination, which can generate the optimal expected profit in the current solution set.

The flowchart in Figure 5 shows how the proposed policy is implemented to obtain the optimal profit, in which the inputs and outputs of the algorithm are specified. The proposed algorithm starts at the beginning of the transaction. Inputs to the algorithm include the cost c_k and quality coefficient of different product levels q_k as well as a set of price combinations P_{discrete} . In general, a large number of continuous variables may result in the curse of dimensionality [33]. For the sake of simplicity, we discretise prices $P_{\text{discrete}} \subset P$ and sort pricing combinations by $a = 1, 2, \dots, \|P_{\text{discrete}}\|$. Let $P_{\text{discrete}}(a) \in P$ denote the a^{th} price combination in the discrete set. The price relationship among different products has been mentioned in Section 4.

Since we do not know the right price in advance according to incomplete information, we can only estimate the true profit through the approximate solution. The aforementioned belief B is composed of Q and N together and will be updated continuously to decide how to implement exploitation as the transaction progress. At the beginning, we initialise $Q^{(i)}(a)$ and $N^{(i)}(a)$ to 0 for all $i = 1, 2, \dots, M$ and $a = 1, 2, \dots, \|P_{\text{discrete}}\|$, which, respectively, represent the estimated profit that can be obtained by selecting price combination a , and the number of times of this action a was selected until the i th customer arriving. We denote the true profit generated by each price as $Q^*(a)$, which is gradually approximated by the maximum estimated profit $Q^{(i)}(a)$ in each price learning process.

Obviously, the provider can directly select a price combination randomly, but such a strategy does not seem to find the optimal solution to the problem. It is impossible to accurately know which prices are relatively excellent and which prices will

lead to poor profits. Random selections usually produce poor performance without reference to an estimated probability distribution. To allocate the weight between exploration and exploitation, the parameter ϵ is generally used in the research study. It is compared to a randomly generated number. For the next decision, we believe that there is a probability of ϵ to explore another price and a probability of $1 - \epsilon$ to exploit the optimal price in the current solutions. An algorithm with a trick like this is called ϵ -greedy. However, the value of ϵ is usually constant, the error will increase linearly as the number of learning increases [34].

Thus, we achieve the balance between exploration and exploitation through the following trick on the basis of traditional ϵ -greedy. We set a larger value of ϵ to increase the probability of strategy exploration at the beginning so that we can attempt different price combinations as much as possible. Then, as the number of customers increases and the iteration progresses, after finding more and better price combinations, the value of ϵ gradually decreases so as to increase the probability of strategy exploitation ($1 - \epsilon$). Thus, we have $p_i = P_{\text{discrete}}(a^*)$, where

$$a^* = \begin{cases} \arg \max_a Q^{(i)}(a), & \text{with probability } 1 - \epsilon \\ \text{random } \{1, 2, \dots, \|P_{\text{discrete}}\|\}, & \text{with probability } \epsilon \end{cases} \quad (4)$$

The advantage of the dynamic ϵ -decaying greedy policy is to allocate the weight of exploration and exploitation in stages to reduce the inevitable errors and accelerate the convergence speed of the algorithm. Then, we express the relationship function between the number of customers and the parameter ϵ as $\epsilon = 1/\sqrt{i}$, where $i \in \{1, 2, \dots, M\}$ indicates the arriving customer.

The pseudocode of the proposed algorithm is given as Algorithm 1, where n_p represents the frequency of price adjustment, specifically the number of customers being served at the same price before the next price adjustment. It is necessary to note that ‘random{.}’ represents a price a is randomly selected from the price set with an equiprobability, and ‘random[.]’ means to generate a number uniformly distributed in the interval.

Algorithm 1 ϵ -decaying greedy

Require: The quality level q_k , the cost c_k , discrete price combinations P_{discrete} , the number of all potential customers M , the number of customers being served at the same price n_p ;

Ensure: The cumulative profits r .

```

1:  $r = 0$ ;
2:  $Q^{(1)}(a) = 0, N^{(1)}(a) = 0, \forall a = 1, 2, \dots, \|P_{\text{discrete}}\|$ ;
3: for  $i = 1$  to  $M$  step  $n_p$  do
4:   if  $(i - 1) \% n_p == 0$ 
```

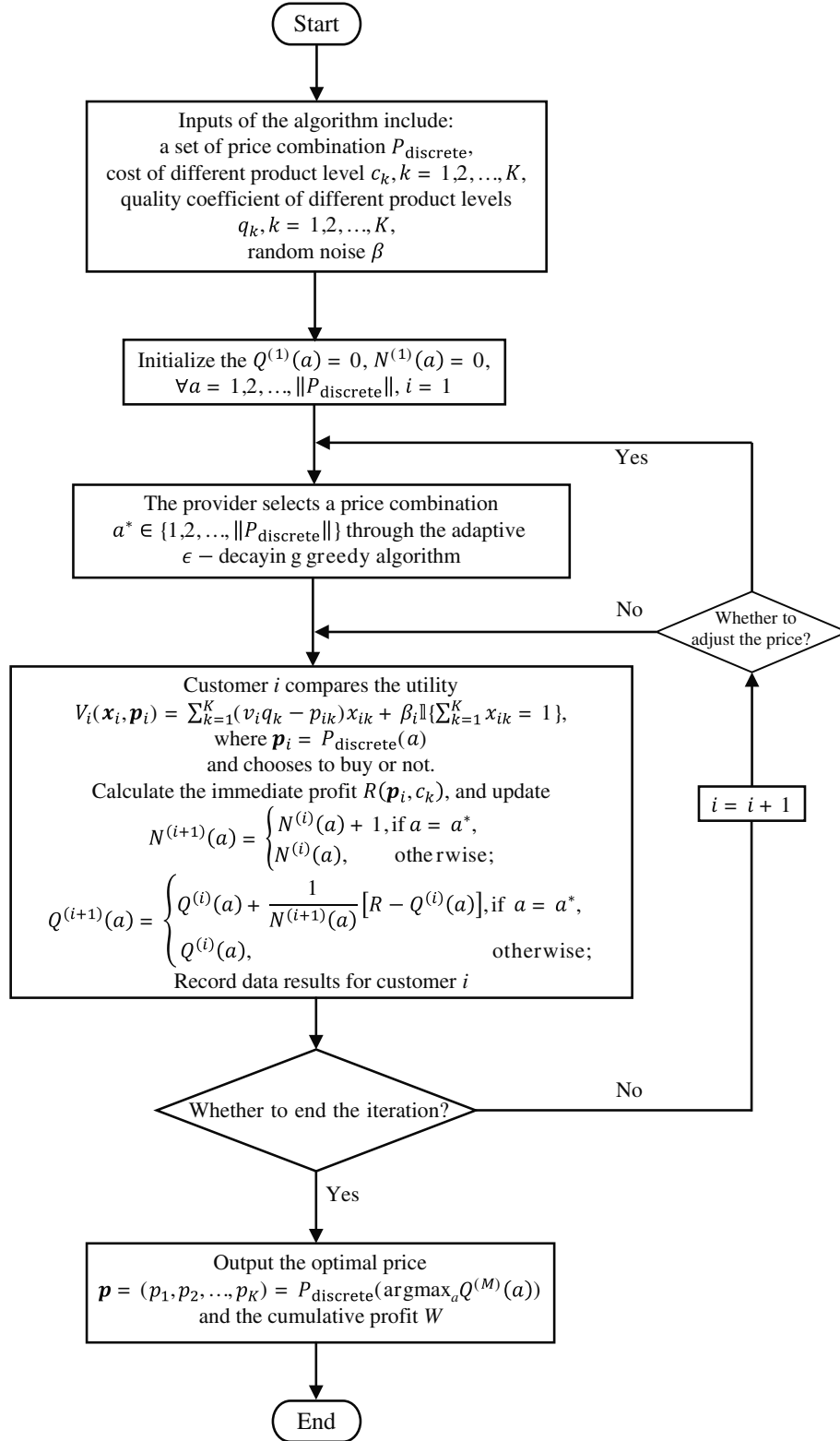


FIGURE 5 Flowchart for implementing the bandit algorithm applies in dynamic pricing problem

```

5:  $\epsilon = 1/\sqrt{i}$ ;
6: if random[0, 1] < 1 -  $\epsilon$  then
7:    $a^* = \arg \max_a Q^{(i)}(a)$ ;
8: else
9:    $a^* = \text{random} \{1, 2, \dots, \|P_{\text{discrete}}\|\}$ ;
10:  $v = R(a^*, c_k, q_k)$ ;
  
```

```

11:  $r = r + v;$ 
12:  $N^{(i+1)}(a) = \begin{cases} N^{(i)}(a) + 1, & \text{if } a = a^*, \\ N^{(i)}(a), & \text{otherwise;} \end{cases}$ 
13:  $Q^{(i+1)}(a) = \begin{cases} Q^{(i)}(a) + \frac{1}{N^{(i+1)}(a)} [v - Q^{(i)}(a)], & \text{if } a = a^*, \\ Q^{(i)}(a), & \text{otherwise;} \end{cases}$ 
14: end for

```

6 | NUMERICAL RESULTS

This section presents the results of numerical simulations to demonstrate the efficacy of our proposed dynamic pricing algorithm of differentiated products. All the simulations are implemented on an Intel Core 2.50 GHz PC.

6.1 | The numerical setting of parameters

To explore the influence of the distribution of unknown customer preference on the results, we assume that customers' consumption preference follows one of the two distributions with the similar mean and variance, namely, Gaussian distribution $N(\mu, \sigma^2)$ and uniform distribution $U(a', b')$, where the Gaussian distribution is truncated to limit the value of the preference v to the interval of $[0, 1]$ to ensure that $v \geq 0$. There are 1000 potential customers in the transaction. We propose several different pricing strategies for customers by taking into account the impact of price adjustments frequency, $n_p \in \{1, 2, 4, 5, 8, 10\}$. It will result in a different number of price experiments that the provider attempts to take. Providing a price combination for a customer ($n_p = 1$) or a group of customers ($1 < n_p < M$) is regarded as a price experiment; with these six strategies, we could attempt 1000, 500, 250, 200, 125 and 100 price experiments, respectively. Meanwhile, we do not know what the customer's specific utility is, so we add a Gaussian perturbation parameter β to reflect randomness in real situations. These scenarios are presented in Table 2.

Suppose that the provider only provides two-quality level products with different quality q_H and q_L . The price and cost of high-quality products are higher than those of low-quality products, and the marginal cost function is linear. Although

some prices are unlikely to appear in actual situations, 45 price combinations we set can divide the accuracy of the data simulation within a certain range and will not affect final results. Table 3 shows features of products in terms of two quality levels and other parameters. It should be pointed out that all parameter values in this simulation scenario are specific and may change according to different price experiments set by the provider. However, it will not distort the analysis of numerical results.

Based on the above hypothetical scenario, simulations are run with iterations, which can distinguish between the excellent and poor price combinations and get the average profit of 500 trials (the profit generated by each customer). Figure 6 shows the convergence of the average profit for Strategy 1 ($n_p = 1$). During 0–100 episodes, the rapid increment of the average profit value displays that we have more opportunities to explore in the process of adjusting the price combinations and gradually find better prices that can generate more profits. The average profit that the provider can obtain by taking the current price is about 0.225. During 100–800 episodes, the trend of increasing average profit is slowing down, indicating that the opportunity to explore better-performing prices is slowly decreasing, and we exploit the previous optimal price more often. Over 800 episodes, the curve gradually converges to about 0.275. It shows that as iterations progress, the proportion of the optimal price we exploit is far greater than the price of exploring new unknown effects. We always have some trade-offs in balancing exploration and exploitation. If we redistribute the number of price adjustments, as previously assumed, the number of customers being serviced will be different; the results we want may change.

TABLE 3 Parameter settings for two quality level products

Parameters on the product side	
Maximum quality level number	$K = 2$ (high-quality and low-quality)
Quality coefficient	$q_1 = 1.0, q_2 = 0.5$
Marginal cost	$c_1 = 0.05, c_2 = 0.01$
Price combinations	$P_k = (p_1, p_2)$, where $k = 1, 2$ and $p_1 > p_2$ $p_1 \in \{0.1, 0.2, 0.3, \dots, 0.9\}$ $p_2 \in \{0.0, 0.1, 0.2, \dots, 0.8\}$

TABLE 2 Parameter settings for heterogeneous customers

Parameters on the customer side						
Customers' consumption preference	$v \in [\underline{v}, \bar{v}], \underline{v} = 0, \bar{v} = 1.0$					
Consumption preference distribution	$N(\mu, \sigma^2)$			$U(a', b')$		
	$\mu = 0.5, \sigma^2 = 0.1$			$a' = 0, b' = 1$		
Pricing strategies	S1	S2	S3	S4	S5	S6
	$n_p = 1$	$n_p = 2$	$n_p = 4$	$n_p = 5$	$n_p = 8$	$n_p = 10$
Total number of customers	$M = 1000$					
Choice random disturbance	White Gaussian noise β ($\mu' = 0, \sigma'^2 = 0.01$)					

6.2 | Sensitivity analysis of parameters n_p and ϵ

In addition to the basic parameters, we also need to consider the effects of other parameters. The number of customers between each price adjustment (n_p), the hyperparameters of the model itself (ϵ) etc. also affect the final average return.

6.2.1 | The influence of parameter n_p on the results

As shown in Figure 7, we compare the total expected profit that the provider can obtain under different pricing frequencies. When analysing the data, the box plots effectively help us identify the characteristics of the data: intuitively identify outliers in the dataset and determine the degree of data

dispersion and bias (observe interquartile range and the length of the whisker) [35].

The first box plot (Strategy 1 ($n_p = 1$)) shows that the median line is higher than the other five (about 270) and the data distribution is closer to the left-skewed distribution. It indicates that the learning effect of the algorithm will be better in the case of more price adjustments, and we are able to obtain more total profits. The size and number of outliers (marked with “+”) and the length of the whisker can reflect the superiority of different price adjustment strategies. Take the last box plot (Strategy 6 ($n_p = 10$)) as an example, outliers are smaller and the length of the whisker is longer. It indicates that the performance of our pricing strategy for 10 customers is unstable. By comparing the interquartile range of the six boxes, we can determine that the price adjustment frequency does affect the total expected profit.

After obtaining the total profit from training, we retrospect pricing selections and distinguish good or poor price

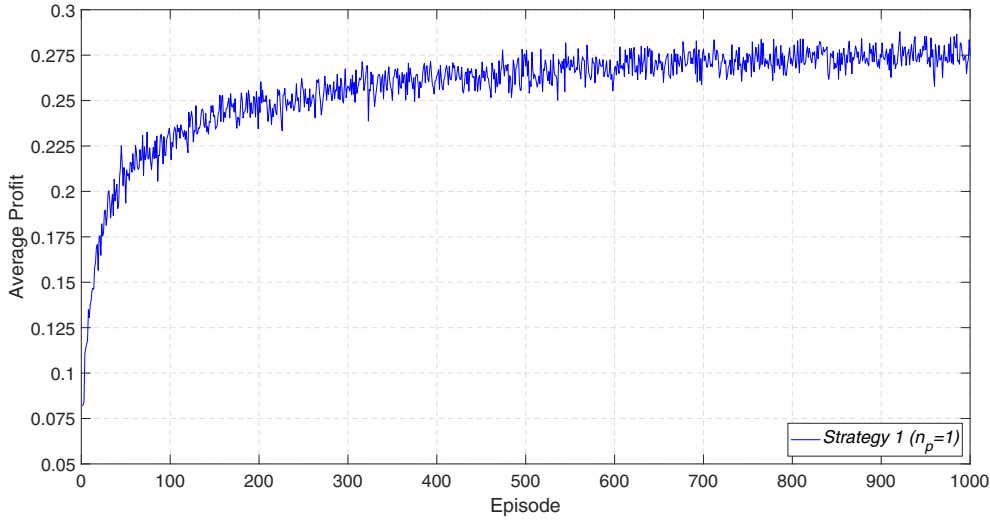


FIGURE 6 Convergence of average profit for strategy 1 ($n_p = 1$)

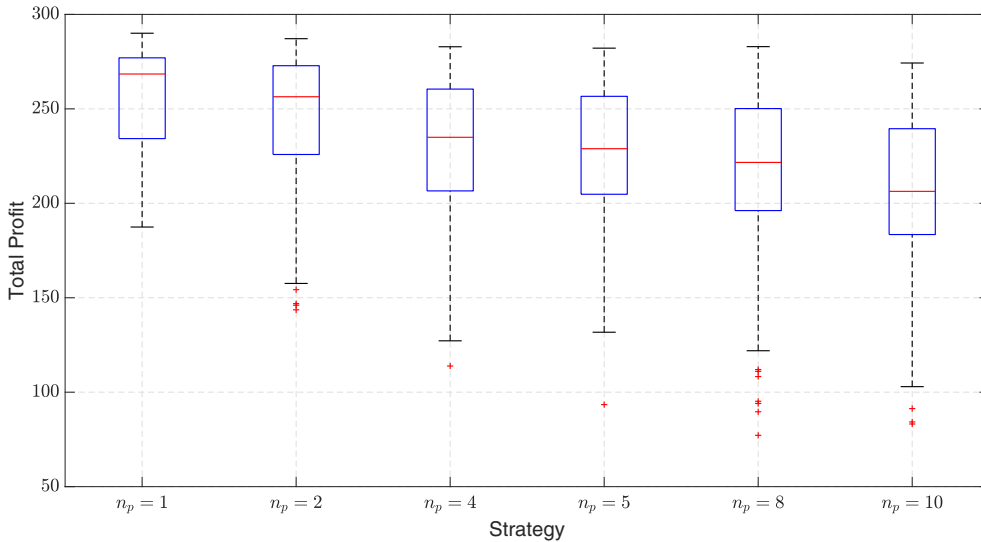


FIGURE 7 Boxplot of total profit under different pricing strategies

combinations. The frequency of price combinations being selected in some of the different strategies is shown in Figure 8. Even if the number of customers of each strategy is different, training results show a similar character of good price combinations. Because consumption preferences of customers are normal distributed with a mean value of 0.5, considering specific values, most customers are more suitable for price positioning below unit 0.5. Obviously, price combinations such as (0.3, 0.1), (0.3, 0.2), (0.4, 0.2), and (0.4, 0.3) are selected more often than others. It implies that we can find and abandon prices that perform worse, thereby reducing redundant price adjustments. However, carefully comparing the frequency changes in the four different strategies, the first subgraph signifies the accuracy and stability of the data; our judgement on excellent price selections will not be disturbed by other poor-effect prices. In the fourth subgraph, the selection frequency of some price changes is very close, which is not conducive to distinguishing the performance of price combinations. As the training progresses, we can always find some better price combinations. It confirms that the price adjustment frequency affects the results of the simulation analysis.

In addition, we consider using hypothesis testing to identify the significant difference in total profit under different pricing frequencies n_p . Because the data obtained by our experiments is not normally distributed, we use the Friedman test [36]. It is a non-parametric alternative test to repeated-measures ANOVA when assumptions such as normal distribution are not satisfied [37]. Non-parametric tests are more robust than parametric tests because they are rank-based tests and are relatively less affected by extreme values. The null hypothesis we give is that the factor n_p has no significant difference in the total profit. Through the Friedman test, the p -value is 0.000, which is less than the significance level

($\alpha = 0.05$). It indicates that the null hypothesis is not true, and there is a significant difference between the overall distribution of multiple samples caused by the pricing frequency. From the above conclusion, we can judge that the higher the frequency of price adjustments, the better the profits obtained.

6.2.2 | The influence of parameter ϵ on the results

In the traditional ϵ -greedy algorithm, the value of ϵ is always fixed, which requires us to manually adjust its size to estimate the optimal value. It has a significant disadvantage: it does not work well to balance exploration and exploitation. Therefore, we change the value of ϵ in the algorithm to explore whether the value of ϵ has a significant difference on the results by controlling the condition of $n_p = 1$. As shown in Figure 9, the larger the value of ϵ (such as $\epsilon = 0.5$), the faster the curve converges, but the smaller the average profit. When the value of ϵ is small (such as $\epsilon = 0.1$), the result is reversed. However, it should be noted that when ϵ is equal to 0.01, the effect is not ideal. Because of the limited pricing frequencies, efficiency is often used as a criterion to evaluate the performance of parameters. Different from the fixing ϵ , the adaptive ϵ -decaying shows excellent properties. It combines characteristics that the early training stage needs more exploration and the later stage needs more exploitation. Its performance is optimal both in comparison convergence rate and maximum average profit.

From Figure 10, with other conditions remain unchanged, the growth trend of the average cumulative profit is related to the value of ϵ .

Differences in characteristics between groups were analysed using the Friedman test. The null hypothesis we give is

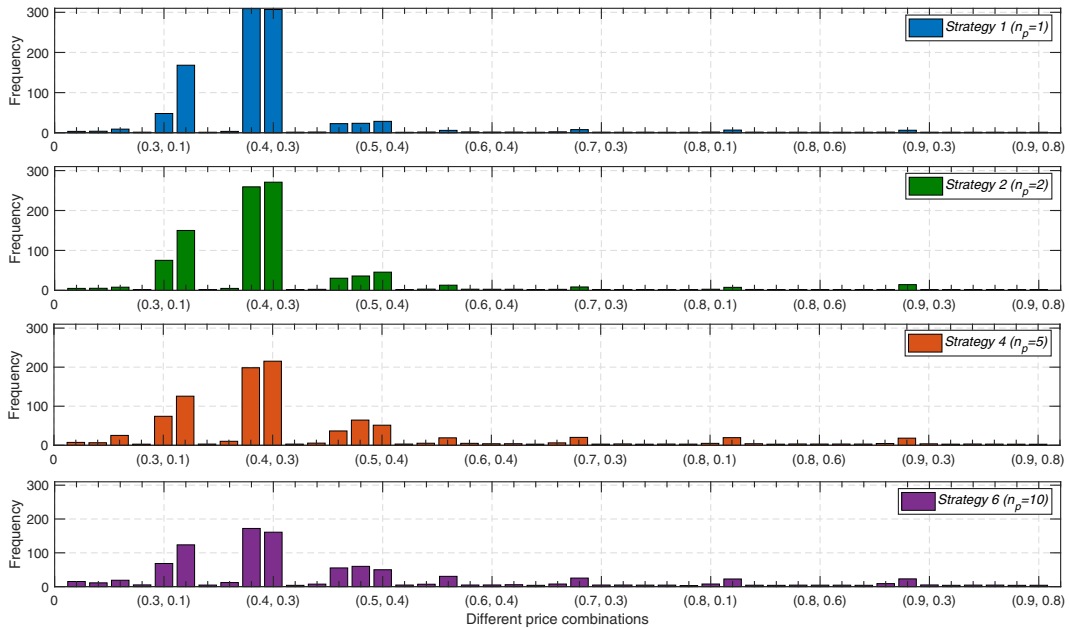


FIGURE 8 Histogram of frequencies for each price combination selected under different strategies

that there is no significant difference in the overall profits obtained under different values of ϵ . It was found that the cumulative profit difference is significant, and the p -value (0.000) is much less than the significance level ($\alpha = 0.05$), which indicates that at least one group differs from the other groups, and the null hypothesis is rejected.

6.3 | Performance comparison of different policies

In order to further explore the matching degree and accuracy of our adaptive algorithm with price adjustment frequencies, we consider two other bandit algorithms for solving our problem under the same conditions, *Softmax* and *UCB1*, which are shown in Algorithm 2 and Algorithm 3, respectively.

Algorithm 2 Softmax [38].

Require: The quality level q_k ; The cost c_k ;
The number of price combinations P_{discrete} ;
The number of all potential customers M ;
Ensure: The cumulative profits r .

- 1: $r = 0$;
- 2: $Q^{(1)}(a) = 0, N^{(1)}(a) = 0, \forall a = 1, 2, \dots, \|P_{\text{discrete}}\|$;
- 3: **for** $i = 1, 2, \dots, M$ **do**
- 4: $\tau = 1/\sqrt{i}$;
- 5: $\Pr(a) = \frac{e^{\frac{Q(a)}{\tau}}}{\sum_{a=1}^{\|P_{\text{discrete}}\|} e^{\frac{Q(a)}{\tau}}}, \forall a = 1, 2, \dots, \|P_{\text{discrete}}\|$;
- 6: a^* : select a randomly from $\{1, 2, \dots, \|P_{\text{discrete}}\|\}$ according to $\Pr(a)$;
- 7: $v = R(a^*, c_k, q_k)$;

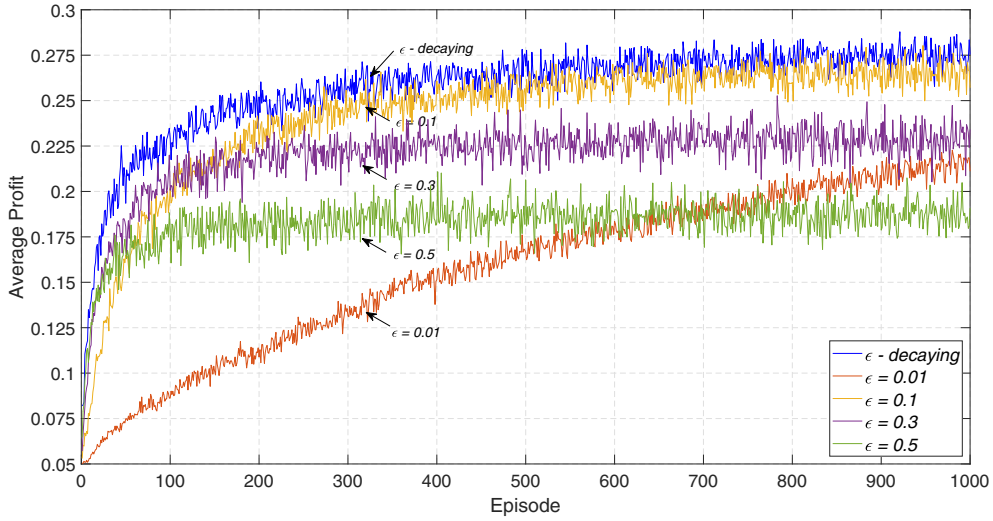


FIGURE 9 Convergence of average profit curves under different ϵ -values

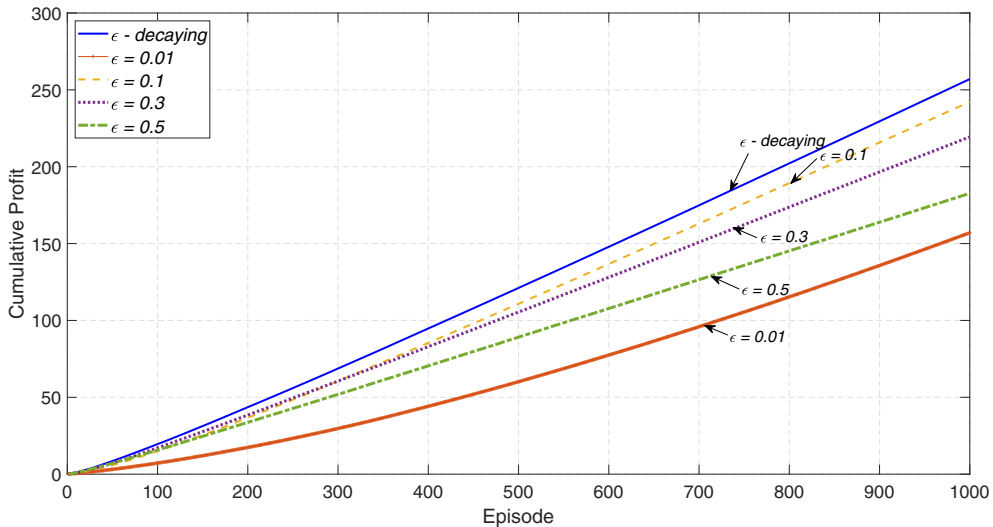


FIGURE 10 The average cumulative profit of 500 independent repeated trials

```

8:   $r = r + v$ ;
9:   $N^{(i+1)}(a) = \begin{cases} N^{(i)}(a) + 1, & \text{if } a = a^*, \\ N^{(i)}(a), & \text{otherwise;} \end{cases}$ 
10:  $Q^{(i+1)}(a) = \begin{cases} Q^{(i)}(a) + \frac{1}{N^{(i+1)}(a)} [v - Q^{(i)}(a)], & \text{if } a = a^*, \\ Q^{(i)}(a), & \text{otherwise;} \end{cases}$ 
11: end for

```

Algorithm 3 UCB1 [39].

Require: The quality level q_k ; The cost c_k ;
The number of price combinations P_{discrete} ;
The number of all potential customers M ;

Ensure: The cumulative profits r .

```

1:  $r = 0$ ;
2:  $Q^{(1)}(a) = 0, N^{(1)}(a) = 0$ ;
3: for  $i = 1, 2, \dots, M$  do
4:   $\text{max\_upper\_bound} = 0$ ;
5:   $a^* = 0$ ;
6:  for  $a = 1, 2, \dots, \|P_{\text{discrete}}\|$  do
7:    if  $N(a) > 0$  then
8:       $\theta_i = \sqrt{\frac{2 \ln(i+1)}{N(a)}}$ ;
9:       $\text{upper\_bound} = Q^{(i)}(a) + \theta_i$ ;
10:   else
11:      $\text{upper\_bound} = \infty$ ;
12:   if  $\text{upper\_bound} > \text{max\_upper\_bound}$  then
13:      $\text{max\_upper\_bound} = \text{upper\_bound}$ ;
14:      $a^* = a$ ;
15:    $v = R(a^*, c_k, q_k)$ ;
16:    $r = r + v$ ;
17:    $N^{(i+1)}(a) = \begin{cases} N^{(i)}(a) + 1, & \text{if } a = a^*, \\ N^{(i)}(a), & \text{otherwise;} \end{cases}$ 
18:    $Q^{(i+1)}(a) = \begin{cases} Q^{(i)}(a) + \frac{1}{N^{(i+1)}(a)} [v - Q^{(i)}(a)], & \text{if } a = a^*, \\ Q^{(i)}(a), & \text{otherwise;} \end{cases}$ 

```

```

19: end for
20: end for

```

The result is shown in Figure 11; the random selection policy is a control algorithm we set, that is, the price combination is selected randomly at each price adjustment. Taking the average profit as the evaluation indicator, we find that the maximum convergence values of the *Softmax* algorithm and the ϵ -decaying greedy algorithm are about 0.275, but the former's convergence rate is slower than the latter's. Moreover, we mark the average profit value that the provider can obtain under complete information as 0.3345. Figure 11 shows that the data of the *UCB1* algorithm fluctuates greatly, and the dispersion degree of the normal value is expected to be higher than that of the other algorithms.

According to the related study [34], the *UCB1* algorithm does not have any parameters that need to be configured during actual operation. In each round of price adjustment, we record the confidence interval of each price combination's reward, and the operation is to record how many times each price combination has been selected. Then, those price combinations with high confidence intervals will be more frequently retained as selections in the future. Therefore, as a 'deterministic' algorithm, it is easy to fall into the local optimum during the early selection process, and the curve will appear to oscillate. As the iteration progresses, the *UCB1* algorithm chooses more price combinations with high confidence intervals and gradually finds the optimal price combination. However, one of its shortcomings is that the convergence rate is too slow, which requires us to conduct more price experiment times. It may result in producing more costs in practical applications.

Similarly, we design the Friedman test to verify the significance among proposed policies. The conclusion we got is that there is a significant difference among different algorithms when the customer preference follows a Gaussian distribution, and the same is true for the uniform distribution.

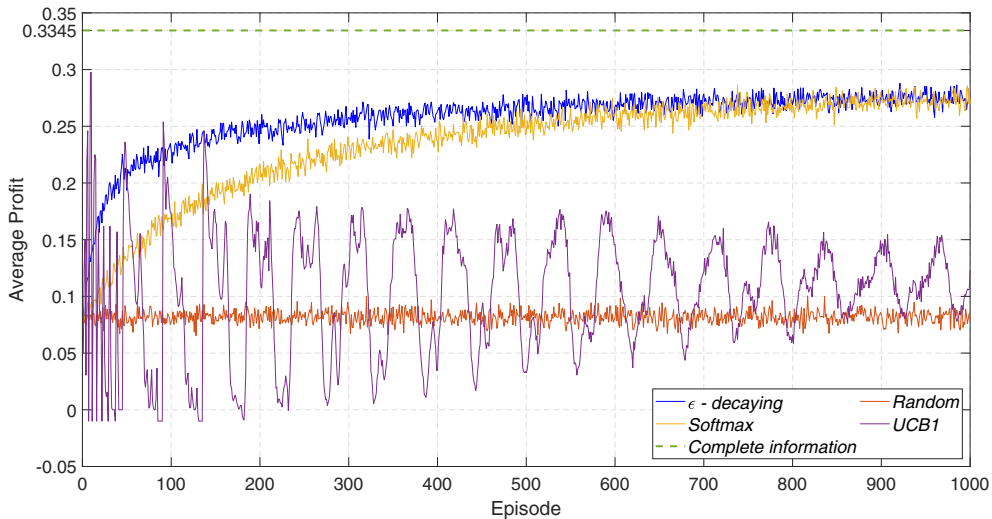


FIGURE 11 The average profit convergence curve obtained under different policies

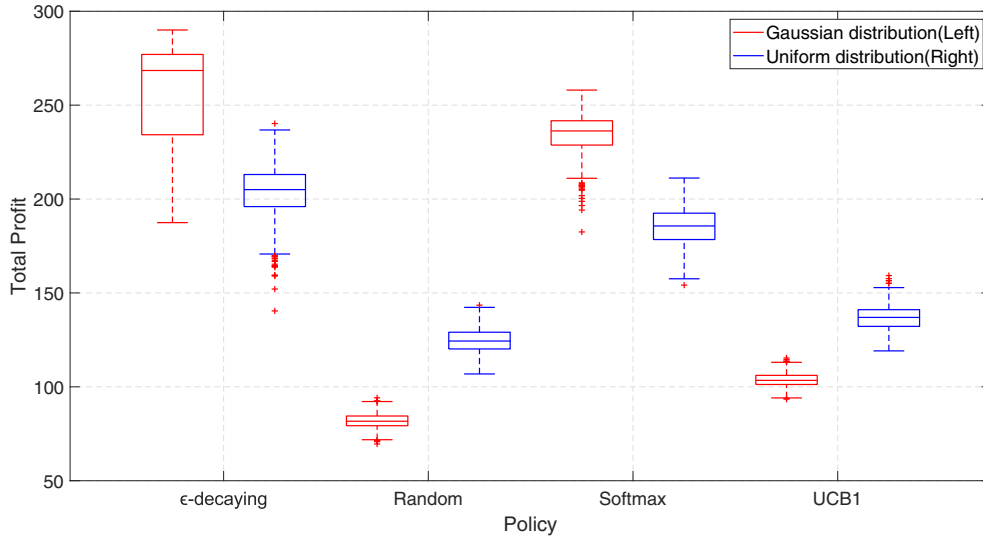


FIGURE 12 Boxplot of total profit under different policies

Figure 12 shows that under different customers' preference distributions, the provider can adopt different policies to implement dynamic pricing to obtain the expected total profit. The results display that the distribution type of customers' preference does affect the total profit. From the boxplot, there are differences in the performance of different policies. If the provider adopts the random algorithm or the *UCB1* algorithm to make dynamic pricing for customers, he/she can obtain more total profits when consumption preference follows a uniform distribution. We believe that each price combination has the same probability of being selected under the random policy. Then, the factor that affects the results is consumer preference, which may lead to a difference in the distribution of customers. We have more opportunities to price customers with high preference and obtain relatively more profits under the uniform distribution, although the number of customers within a certain preference interval may not be as concentrated as under the Gaussian distribution.

Moreover, the improved algorithm we proposed is the most robust among the four bandit algorithms, and it shows better training performance regardless of the distribution conditions.

6.4 | Discussion of numerical results

Considering differentiated products with uncertain consumers' preferences, the price combination has a significant impact on average returns. We choose Gaussian distribution and uniform distribution as the assumptions of the consumers' preference distribution in the customer's utility function and then use the MAB series of algorithms to determine the optimal price combination. Multi-armed bandit problem series algorithms are continuously explored and developed through different strategies, and an appropriate price combination is selected at each step. Compared with the traditional ϵ greedy algorithm,

we adopt a parameter decay strategy. As the iteration progresses, *ϵ -decaying greedy* algorithm prefers to use the existing optimal price combination rather than explore unknown price combinations. By conducting numerical experiments and hypothesis testing, we found that the higher the frequency of price adjustments, the higher the final benefit obtained.

For a given number of customers, the frequency of price adjustments will be limited, which will further affect the average revenue. At this time, the algorithm needs to converge faster and obtain better average returns. Experiments show that the traditional ϵ greedy algorithm is more dependent on the value of ϵ , and the size of ϵ has different effects on the convergence speed and average profit of the model, and the size of these parameters needs to be adjusted manually. The ϵ decay strategy makes the algorithm explore more frequently in the early stage and develop more in the later stage so that it can converge faster and achieve better average returns. Compared with other algorithms such as Random selection strategy, *UCB1* and the *Softmax* algorithm, regardless of the conditional distribution, our *ϵ -decaying greedy* algorithm has faster convergence and better returns than other algorithms.

7 | CONCLUSION

This paper studies the dynamic pricing problem of differentiated products with incomplete information. Combining with reinforcement learning, we figure out the optimal pricing strategy in a data-driven way by learning the recorded purchase results.

How suppliers set prices in the face of no sales records and unknown consumers' preferences can have a big impact on their profits. We regard this problem as a sequential decision-making problem, and then we propose an adaptive algorithm called *ϵ -decaying greedy*. The algorithm relies less on information such as customer preferences and preference

distribution. We adjust the parameter ϵ with changes in the number of customers during the training process to achieve balancing the exploration of new price selections and the exploitation of the known best prices. The simulation results show that the frequency of price adjustment is an important factor affecting the total profit. Comparing with other existing algorithms, the performance of our algorithm is relatively better in different preference distributions, which can effectively provide different prices for customers with different demands, reduce the number of the provider's pricing and find the optimal pricing strategy.

Our paper obviously has some limitations, and these limitations will be some future research directions. First, we assume that customer preferences are Gaussian and uniform, and the actual distribution of consumers' preferences can be quite complex. Models built using Gaussian and uniform distributions can cause actual returns to differ from ideal returns in applications. Second, the utility function we assume is a relatively simple additive function, and the noise used is Gaussian noise. In practice, the impact of noise on the environment is very complex. In addition, we assume that the supplier is a monopoly, and differentiated products may bring potential competitors. If there are competitors, the model is more complex.

In the future, we will focus on more characteristics of customers. Whereas, the amount of feature information is huge. Once the feature space jumps from lower to higher dimensions, the current method may cause dimensional disaster and make it impossible to calculate. We hope to analyse the observable consumption characteristics in combination with neural networks to further simulate a more accurate consumers' preference distribution and express the needs of more customers. Meanwhile, we further consider the restrictions on price adjustments. Within limited price adjustments, we may not allocate equally the frequency of price adjustments according to the number of customers. Under such a circumstance, how to reasonably allocate price adjustment frequency will be an important direction for future research.

ACKNOWLEDGMENTS

This work was supported partially by the Natural Science Foundation of Zhejiang Province [grant number LY18G010019 and LY20G010011], the Humanities and Social Science Fund of Ministry of Education of China [grant number 19YJCZH020], and the National Natural Science Foundation of China [grant number L1924063].

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Cheng Wang  <https://orcid.org/0000-0003-2475-1944>

REFERENCES

1. Brynjolfsson, E., Hu, Y., Simester, D.: Goodbye pareto principle, hello long tail: the effect of search costs on the concentration of product sales. *Manag. Sci.* 57(8), 1373–1386 (2011)
2. Choi, T.M., et al.: Optimal pricing in mass customization supply chains with risk-averse agents and retail competition. *Omega-International Journal of Management Science.* 88, 150–161 (2019)
3. Koch, S., Klein, R.: Route-based approximate dynamic programming for dynamic pricing in attended home delivery. *Eur. J. Oper. Res.* 287(2), 633–652 (2020)
4. Guerrero-Lorente, J., Gabor, A.F., Ponce-Cueto, E.: Omnichannel logistics network design with integrated customer preference for deliveries and returns. *Comput. Ind. Eng.* 144, 106443 (2020). <https://doi.org/10.1016/j.cie.2020.106433>
5. Altug, M.S.: Supply chain contracting for vertically differentiated products. *Int. J. Prod. Econ.* 171, 34–45 (2016)
6. Wang, Y.L., et al.: Adversarial behaviours in mixing coins under incomplete information. *Appl. Soft Comput.* 96, 106605 (2020)
7. Lin, K.Y.: Dynamic pricing with real-time demand learning. *Eur. J. Oper. Res.* 174(1), 522–538 (2006)
8. Feng, Q., Luo, S., Shanthikumar, J.G.: Integrating dynamic pricing with inventory decisions under lost sales. *Manag. Sci.* 66(5), 2232–2247 (2020)
9. Lu, R.Z., Hong, S.H., Zhang, X.F.: A dynamic pricing demand response algorithm for smart grid: reinforcement learning approach. *Appl. Energy.* 220, 220–230 (2018)
10. Etebari, F.: A simultaneous facility location, vehicle routing and dynamic pricing in a distribution network. *Appl. Soft Comput.* 83, 105647 (2019)
11. Akcay, Y., Natarajan, H.P., Xu, S.H.: Joint dynamic pricing of multiple perishable products under consumer choice. *Manag. Sci.* 56(8), 1345–1361 (2010)
12. Liu, Q., Zhang, D.: Dynamic pricing competition with strategic customers under vertical product differentiation. *Manag. Sci.* 59(1), 84–101 (2013)
13. Keskin, N.B., Birge, J.R.: Dynamic selling mechanisms for product differentiation and learning. *Oper. Res.* 67(4), 1069–1089 (2019)
14. Chen, K.H., et al.: Dynamic pricing in the presence of reference price effect and consumer strategic behaviour. *Int. J. Prod. Res.* 58(2), 546–561 (2020)
15. Laussel, D., Van Long, N., Resende, J.: Quality and price personalization under customer recognition: a dynamic monopoly model with contrasting equilibria. *J. Econ. Dynam. Control*, 25 (2020)
16. Cohen, M., Lobel, I., Leme, R.: Feature-based dynamic pricing. *Manag. Sci.* 66, 4921–4943 (2020)
17. Javanmard, A., Nazerzadeh, H., Shao, S.: Multi-product dynamic pricing in high-dimensions with heterogeneous price sensitivity. *IEEE Int. Symp. Inf. Theor.* 2652–2657 (2020)
18. Mnih, V., et al.: Human-level control through deep reinforcement learning. *Nature.* 518(7540), 529–533 (2015)
19. Kiran, P., Chandrakala, K.: New interactive agent based reinforcement learning approach towards smart generator bidding in electricity market with micro grid integration. *Appl. Soft Comput.* 97, 106762 (2020)
20. Compare, M., et al.: Reinforcement learning-based flow management of gas turbine parts under stochastic failures. *Int. J. Adv. Manuf. Technol.* 99(9–12), 2981–2992 (2018)
21. Hu, L., et al.: Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *J. Manuf. Syst.* 55, 1–14 (2020)
22. Almeida, C.P., et al.: Hyper-heuristics using multi-armed bandit models for multi-objective optimization. *Appl. Soft Comput.* 95, 106520 (2020)
23. Huo, X.G., Fu, F.: Risk-aware multi-armed bandit problem with application to portfolio selection. *R. Soc. Open Sci.* 4(11), 171377 (2017)
24. Durand, A., et al.: A machine learning approach for online automated optimization of super-resolution optical microscopy. *Nat. Commun.* 9, 1–16 (2018)
25. Liu, B., et al.: Customized Nonlinear Bandits for Online Response Selection in Neural Conversation Models, Thirty-Second Aaai Conference on Artificial Intelligence/Thirtieth Innovative Applications of Artificial

- Intelligence Conference/Eighth Aai Symposium on Educational Advances in Artificial Intelligence. Assoc Advancement Artificial Intelligence, Palo Alto (2018)
26. Ding, K.Z., Li, J.D., Liu, H.: Acm, Interactive Anomaly Detection on Attributed Networks, Proceedings of the Twelfth Acm International Conference on Web Search and Data Mining. Assoc Computing Machinery, New York (2019)
 27. Bouneffouf, D., Rish, I.: A Survey on Practical Applications of Multi-Armed and Contextual Bandits. *arXiv preprint arXiv:1904.10040* (2019)
 28. Misra, K., Schwartz, E.M., Abernethy, J.: Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Market. Sci.* 38(2), 226–252 (2019)
 29. Pakseresht, M., et al.: Co-reconfiguration of product family and supply chain using leader-follower stackelberg game theory: Bi-level multi-objective optimization. *Appl. Soft Comput.* 91, 106230 (2020)
 30. Peng, Y., Lu, Q.: Complex dynamics analysis for a duopoly stackelberg game model with bounded rationality. *Appl. Math. Comput.* 271, 259–268 (2015). <https://doi.org/10.1016/j.amc.2015.08.138>
 31. Li, M.Q., et al.: Optimal versioning strategy for information products with behavior-based utility function of heterogeneous customers. *Comput. Oper. Res.* 40(10), 2374–2386 (2013)
 32. Xue, Y.X., et al.: An integrated linguistic mcdm approach for robot evaluation and selection with incomplete weight information. *Int. J. Prod. Res.* 54(18), 5452–5467 (2016)
 33. Krashennnikova, E., et al.: Reinforcement learning for pricing strategy optimization in the insurance industry. *Eng. Appl. Artif. Intell.* 80, 8–19 (2019)
 34. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multi-armed bandit problem. *Mach. Learn.* 47(2-3), 235–256 (2002)
 35. Mirzargar, M., Whitaker, R.T., Kirby, R.M.: Curve boxplot: generalization of boxplot for ensembles of curves. *IEEE Trans. Visual. Comput. Graph.* 20(12), 2654–2663 (2014)
 36. Friedman, M.: Comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* 11(1), 86–92 (1940)
 37. Zimmerman, D.W., Zumbo, B.D.: Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks. *J. Exp. Educ.* 62(1), 75–86 (1993)
 38. Koulouriotis, D.E., Xanthopoulos, A.: Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems. *Appl. Math. Comput.* 196(2), 913–922 (2008)
 39. Garivier, A., Moulines, E.: On Upper-Confidence Bound Policies for Switching Bandit Problems, Vol. 6925 of Lecture Notes in Artificial Intelligence, pp. 174–188. Springer-Verlag Berlin, Berlin (2011)

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Wang, C., et al.: Dynamic pricing of differentiated products with incomplete information based on reinforcement learning. *IET Collab. Intell. Manuf.* 1–16 (2022). <https://doi.org/10.1049/cim2.12050>