Research on the Optimization Method of Spark-based Reinforcement Learning in Dynamic Pricing of Airline Tickets

Xiwen Hu
South China Normal University, Foshan, China
u06xh23@abdn.ac.uk

ABSTRACT

This paper discusses the difficulties of dynamic pricing of civil aviation. With more competition, the ticket won't last long. Traditional pricing methods and solutions have limitations compared with single aircraft reinforcement learning. They are not fast enough to deal with complex situations well and can't respond in real time. We are looking for a better way to dynamically price airline tickets. To this end, we combine Spark distributed computing with reinforcement learning depth. We create a hierarchical and distributed pricing system. We use Spark to process data, compress functions and access data in real time. We are improving the MDP pricing model and using DKN algorithm to build a decision engine. The experimental results show that our solution is much better. It can process data faster and respond in real time faster. Seat Occupancy Rate increased by 5 percentage points. It is also sensitive and robust when computing resources, market competition or scenario complexity change. This study provides a feasible technical path for the engineering application of reinforcement learning in dynamic airline ticket pricing and has important practical value for enhancing the revenue management efficiency of airlines.

Keywords: Reinforcement learning, Spark, Markov Decision Process (MDP), Deep Q-Network (DQN).

1. INTRODUCTION

The international development of the global civil aviation industry and the intensifying market competition have made airline ticket pricing a core issue in airlines' revenue management. As a perishable product, airline tickets require dynamic price adjustment within a limited sales cycle to maximize revenue. Traditional pricing methods rely on manual experience or static rule-based models, which are difficult to adapt to complex markets with non-stationary demand and frequent price adjustments by competitors. Although reinforcement learning provides an effective solution for dynamic pricing, its application in airline pricing scheme is limited by stand-alone architecture. It faces problems such as large state space, large amount of data and lack of real-time computing ability. Although Spark distributed computing framework is widely used in large-scale data processing, it has not been deeply integrated with reinforcement learning model. Therefore, it is of great significance to explore the dynamic pricing scheme combining the two for improving the technical feasibility and practical application value of the airline pricing system.

2. LITERATURE REVIEW

Domestic and foreign researchers have done a lot of research on the dynamic pricing of air tickets. They studied revenue management, demand modeling and intelligent algorithms. The research method has gradually changed from the traditional rule-based model to an intelligent algorithm using a large amount of data. But at every stage of the research, there are limitations. We haven't found a comprehensive pricing plan suitable for the very complicated aviation market.

The early research was based on the classic revenue management framework. Air ticket price is regarded as an inventory control problem. Higher People (2022) tried to maximize income by forecasting demand and setting price levels. These methods assume that demand is stable and it is difficult to keep up with market changes. With the improvement of computing power, some researchers have introduced supervised learning methods. They built regression models and neural networks from historical sales data to understand the relationship between price and demand. Although these methods make the pricing more accurate, the model can't adjust itself in the competitive environment and can't adapt to the unfamiliar market well [1].

Strengthening the learning foundation has brought new insights to pricing research. Kastius & Schlosser (2022) described the pricing process as markov decision processes, and used tabular algorithms such as Q-learning and SARSA to create pricing decisions. This helps to increase revenue in a simulated environment. However, these methods have a low state dimension, and the computational cost increases significantly when extended to scenarios involving multiple flights and multiple competitors. Although deep reinforcement learning has overcome the dimensional limitations of tabular methods, Reinforcement Learning for Multi-Flight Dynamic Pricing (2024) verified the applicability of the DQN algorithm in high-dimensional states. Nevertheless, its single-machine training mode struggles to meet the real - time and scalability requirements of airline pricing [2][3].

At the data processing level, Spark distributed computing has been applied to the analysis of air transportation data. However, existing studies only use it as an independent analysis tool. The research by Lange et al. (2024) confirms that the current situation of the lack of in-depth coupling between them has become a significant obstacle to the engineering implementation of intelligent pricing algorithms [4].

3. ANALYSIS OF THE LIMITATIONS OF EXISTING DYNAMIC AIRFARE PRICING SCHEMES BASED ON REINFORCEMENT LEARNING

3.1 The typical modeling framework of the existing reinforcement learning pricing schemes

In the research on reinforcement learning for dynamic airline ticket pricing, the academic community has gradually established a relatively unified core modeling paradigm. The common practices can be summarized as follows: Most studies abstract the airline ticket sales process into a Markov Decision Process (MDP), with the continuous decision-making sequence over the entire pricing cycle as the research object. The core objective is to find the optimal pricing strategy that maximizes the total flight revenue through trial-and-error learning [1].

In the construction of the state space, to reduce the model complexity and ensure the trainability of the algorithm, most existing solutions adopt combinations of low-dimensional variables. These combinations typically include core indicators such as the number of remaining seats, the remaining time until departure, and the historical sales rate. Some studies supplement demand-related parameters such as the passenger arrival rate. However, generally, high-dimensional information such as market competition structure, cross-flight linkage, or fluctuations in complex scenarios is not systematically incorporated. In terms of action space design, the continuous ticket prices are often discretized into a finite set (e.g., several price levels within a fixed price range), and the model achieves dynamic price adjustment by selecting different price levels [2].

In terms of algorithm selection, tabular reinforcement learning methods still dominate in current research, with the Q-learning, SARSA, and Q(λ) algorithms being the most widely applied. These methods rely on the explicit storage of the state-action value function and gradually approach the optimal strategy through iterative updates. At the data source level, current research mainly relies on two types of data for support: one is the desensitized data from airlines' historical ticket sales records, and the other is the passenger demand data generated through methods such as the non-stationary Poisson process in a single-machine computing environment, which is used for model training and validation [1][3].

Overall, this modeling paradigm has certain advantages in simplifying the research scenario and focusing on the core pricing decision-making logic. However, its deliberate abstraction of the complexity of the real-world market also lays hidden risks for the limitations of the subsequent model in aspects such as scale expansion and real-time application.

3.2 The primary limitations of existing reinforcement learning-based pricing methods

3.2.1 Computational efficiency bottlenecks under the dependence of large-scale pricing data

The effective training of reinforcement learning models is highly dependent on sufficient state-action-reward interaction samples. However, the complexity of the aviation market makes the amount of price-related data increase at a very, very fast speed, just like an explosion, when the situation becomes more and more. The existing research is mostly simple, with only one flight and few competitors. But in the real market, to determine the price, it is usually necessary to combine the data of multiple flights, the real-time prices of many competitors, and the arrival order of passengers in different situations. These data come from many places and are of different types [3].

For example, when the model must consider 10 competitors, 4 seasonal situations and 2 matching flights at the same time, the data volume may become several times larger than that of only one flight in the traditional case, but most current price research and reinforcement learning use a computer to generate data, process features and drive the model. When there are a large number of data, this method usually meets the problem of slowness: on the one hand, it takes a long time to create simulation data with a large number of dimensions. On the other hand, the memory and computing power of a single computer are not enough to store and train models with a large number of interactive samples. Usually, for this reason, the training process needs to be simplified or stopped [4].

Therefore, this slow problem is not only caused by "too much data". In fact, there are differences between the architecture of a single computer and the data requirements of large-scale intensive learning pricing. This directly limits the ability of the model to turn to more complex and realistic scenes.

3.2.2 Tabular reinforcement learning methods exhibit limited scalability in high-dimensional state spaces

In order to ensure that the model can be trained, the existing pricing research using reinforcement learning greatly simplifies the state space. Usually, we only keep two or three important dimensions. However, in real life, the return of a flight depends not only on its own inventory and remaining time, but also on other key factors. These factors include the price vector of competitors, the intensity of passenger demand, the types of seasons or holiday scenarios, and the related inventory of multiple flights [2].

When there are multiple competitors and scene variables, the size of the state space will increase rapidly. The reinforcement learning methods in the table (such as Q-learning and Q(λ)) must remember each state-action pair. The size of memory increases exponentially with the state dimension. The existing research shows that in the pricing scheme with multiple flights or competitors, the size of Q table usually quickly exceeds the capacity of current computing resources. This will lead to a significant decline in training efficiency, and even lead to non-convergence of the model [3].

Even if the storage pressure is reduced by further shearing or compressing the spatial state, the accuracy of state representation will inevitably be lost. This will lead to problems such as unstable strategy and slow convergence speed. This lack of scalability is not caused by improper selection of specific parameters, but by the structural defects of tabular reinforcement learning methods in complex and high-dimensional decision-making environments.

3.2.3 Insufficient real - time decision - making ability in the offline training mode

Most existing research on reinforcement learning for airline ticket pricing adopts a two-stage model of "offline training-online application", that is, the model is trained based on historical data or batch-generated simulated data, and then the trained pricing strategy is applied to the actual sales scenario. However, this model largely ignores the highly dynamic and real-time characteristics of the aviation market.

In the real market, price adjustments of competitors, passengers' search and booking behaviors, and changes in flight inventory often occur on an hourly or even minute-by-minute basis, especially on high-competition trunk routes or near the departure time, where such changes are more frequent. Existing research has pointed out that the pricing strategies obtained through offline training have difficulty in responding promptly to short-term promotional activities of competitors or sudden fluctuations in demand, and are prone to the problem of pricing lag [2][5].

When competitors quickly lower their prices, the offline model may still maintain the original high-price strategy, resulting in the loss of customers; while in the case of a sudden surge in demand, the model has difficulty in raising prices in a timely manner to capture potential profits. This problem of insufficient real-time response ability makes the performance of the reinforcement learning model in actual operation often significantly lower than its theoretical effect in offline experiments.

3.2.4 Engineering problems of insufficient data management and system reusability

In addition to the limitations at the algorithm level, the existing research on reinforcement learning for pricing also has obvious deficiencies at the engineering implementation level. Most research focuses more on improving model performance, while giving limited consideration to the data management and system architecture design that support the model operation, resulting in a research tendency of "emphasizing algorithms and neglecting engineering" [4].

At the data management level, training data, simulated data, and experimental results are mostly stored in text formats such as CSV in a decentralized manner, lacking a unified data organization and management mechanism, which leads to high costs for cross-experiment comparison, parameter tuning, and result reproduction. At the system design level, the modules of data generation, feature pre-processing, model training, and result analysis are often highly coupled and lack standardized interfaces. When the experimental scenario or parameters change, multiple modules need to be modified repeatedly.

These engineering problems have limited impact in small-scale experiments, but in the complex settings of multi-flights, multi-competitors, and multi-scenarios, their negative impacts will be significantly magnified, becoming an important obstacle for the reinforcement learning pricing method to move from academic research to practical application.

3.3 Summary: Improvement directions of existing reinforcement learning pricing methods

In summary, reinforcement learning theoretically has good dynamic decision-making ability and can effectively depict the complex temporal dependence relationship in the airline ticket pricing process. However, the limitations exposed by existing research indicate that the problems do not stem from the reinforcement learning concept itself, but are mainly reflected in the mismatch between the computing architecture, state space modeling method, and engineering implementation level and the real - world aviation market requirements.

Specifically, the high-dimensional state space and large-scale pricing data pose higher requirements for computing power, making tabular methods and single-machine architectures difficult to support the expansion of complex scenarios; the offline training and decentralized data management models are also difficult to meet the requirements of the aviation market for real-time response and system reusability. Therefore, to improve the practical applicability of the reinforcement learning pricing method, it is urgent to conduct overall optimization at the system architecture level and break through the bottlenecks in efficiency and scale of existing methods by introducing distributed data processing and computing mechanisms.

4. DESIGN OF A REINFORCEMENT LEARNING PRICING OPTIMIZATION SCHEME BASED ON SPARK

4.1 Overall architecture design of the optimization scheme

In order to solve the limitations of the existing reinforcement learning pricing methods analyzed in chapter 3 in terms of calculation efficiency, report scalability and real-time response ability, this paper proposes an air ticket pricing optimization scheme based on reinforcement learning using Spark distributed computing framework. The scheme completely reconstructs the process of data generation, feature processing and decision-making at the system architecture level through reinforcement learning, and forms a closed-loop system supporting large-scale data processing and real-time pricing decision-making.

The overall architecture adopts the hierarchical design idea of “data-driven + distributed computing + enhanced learning decision-making”. The system mainly consists of data access layer, Spark distributed computing and storage layer, reinforcement learning pricing decision layer and price information output and return layer. Each module works together through a standardized data interface. This allows the system to easily expand to handle many flights, many competitors and many situations, while maintaining the same decision logic through reinforcement learning.

In this architecture, Spark not only performs the routine tasks of offline data processing. It also participated in nation-building and feature compression. Therefore, it provides a good quality and well-organized input state for the subsequent reinforcement learning model, which is helpful to reduce the computational pressure caused by the large state space [6].

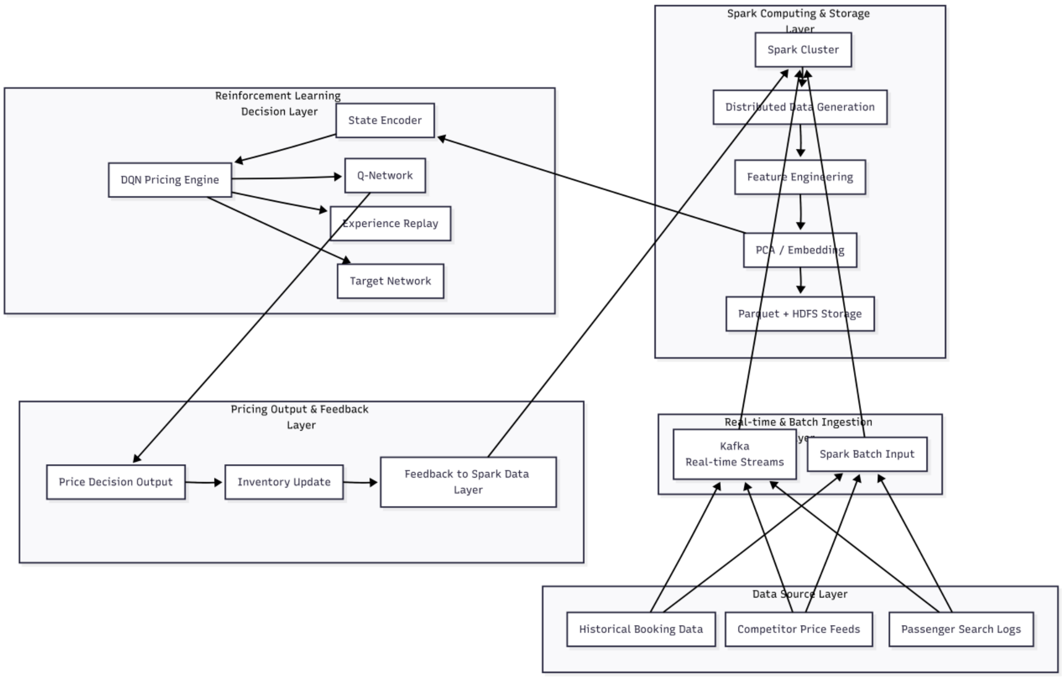

Figure 1. The overall architecture of the reinforcement learning pricing system based on Spark.

Figure 1 shows the overall architecture of the reinforcement learning pricing system based on Spark proposed in this paper. The system accesses market data in real time through Kafka, and realizes distributed data generation, functional engineering and storage management in Spark cluster. The processed state features are input into the reinforcement learning pricing engine, which outputs dynamic pricing decisions, forming a closed-loop feedback mechanism of data and decisions.

4.2 Functional Positioning of Spark in the Pricing System

4.2.1 Distributed Data Generation and Storage Mechanism

In the real aviation market environment, the learning of air ticket pricing strategies highly depends on large-scale, multi-scenario data support. To overcome the problems of low data generation efficiency and limited storage in a single-machine environment, this paper introduces the distributed computing capabilities of Spark to parallelly simulate and generate passenger arrival sequences, competitor price trajectories, and multi-flight inventory change processes.

Through the RDD and DataFrame mechanisms of Spark, the system can simultaneously generate pricing samples under different routes and different scenario parameter combinations in a cluster environment, significantly reducing the data preparation time. The generated data is uniformly stored in HDFS in Parquet format. Compared with traditional CSV files, this method has a higher compression ratio and columnar reading efficiency, significantly reducing the I/O overhead in the subsequent feature calculation and model training stages. At the same time, the unified data storage specification also provides good support for cross-experiment reuse and multi-round parameter tuning.

4.2.2 Real-time Data Access and Preprocessing

To cope with the strong real-time characteristics of the aviation market, the system introduces Kafka and Spark Streaming components at the data access layer to build a real-time data processing channel. Information such as competitor price changes, passenger search behaviors, and real-time inventory changes is written into Kafka in the form of message streams, which are continuously consumed by the Spark Streaming module for online feature updates.

During the real-time processing, the system filters and smoothes abnormal data. For example, it imposes threshold constraints on abnormal price jumps in a short period to avoid interference from noisy data on pricing decisions. The processed real-time features, together with historical statistical features, form the input state of the reinforcement learning model, enabling the model to maintain a high response sensitivity in a dynamic market environment [7].

4.2.3 High-dimensional Data Processing and Feature Compression

With the increase in the number of competitors, flight scale, and scenario dimensions, the dimension of the original state variables expands rapidly. To relieve the training pressure on the reinforcement learning algorithm caused by the high-dimensional state space, this paper introduces a feature compression mechanism in the Spark computing layer.

Specifically, principal component analysis (PCA) is used to reduce the dimension of high-dimensional continuous features, and categorical or discrete features are mapped to a low-dimensional continuous space through embedding representation. The compressed features significantly reduce the state representation dimension while retaining the main information, providing a feasible basis for the subsequent use of deep reinforcement learning methods from an engineering perspective.

4.3 Design of the Optimized MDP Pricing Model

4.3.1 State Space Design

This paper formalizes the problem of dynamic air ticket pricing as a Markov decision process (MDP), defined as follows:

$$M = < S, A, P, R, \gamma > \tag{1}$$

At time t, the system state is defined as follows:

$$s_t = \left[\theta_t, n_t, p_t^{comp}, h_t, \tau_t\right] \tag{2}$$

Among them, θ_t represents the current market scenario type,n_t is the remaining inventory vector of multiple flights, p_t^{comp} denotes the price information of competitors, h_t is the demand heat index, and τ_t is the remaining time until departure. Compared with the traditional low-dimensional state modeling, this design can contain more complete market information with the support of Spark [8].

4.3.2 Design of the action space

In the design of the action space, this paper represents the joint pricing behavior of multiple flights as a price vector.

$$a_t = \left(p_t^{(1)}, p_t^{(2)}, \dots, p_t^{(K)}\right),\ \ p_t^{(k)} \in P \tag{3}$$

Here, k represents the number of connecting flights and p represents the preset discrete price set. This design ensures that the size of the action space is controllable, taking into account the feasibility of project implementation.

4.3.3 Design of the Reward Function

To balance revenue maximization, competitive advantage, and inventory risk, this paper constructs the following composite reward function:

$$R_t = \sum_{k=1}^{K} p_t^{(k)} \cdot d_t^{(k)} + \alpha \cdot C_t - \beta \cdot U_t \tag{4}$$

The first term represents direct sales revenue, C_t is the term of competitive advantage, U_t is the term of inventory punishment, and α and β are the weight coefficients. This reward structure helps to guide the model to strike a balance between long-term income and short-term risk.

4.4 Optimized Design of the Reinforcement Learning Algorithm

4.4.1 Algorithm Selection and Network Structure

In order to meet the needs of high-dimensional state space and complex pricing strategy, this paper chooses Deep Q Network (DQN) as the main decision algorithm. DQN approaches the state-action function through neural network, thus avoiding the problem of storing and calculating tables in high-dimensional space [9].

Its target value is defined as:

$$y_t = r_t + \gamma \max_{a'} Q_{\theta^-}(s_{t+1}, a') \tag{5}$$

The loss function is:

$$L(\theta) = E\left[\left(y_t - Q_{\theta(s_t,a_t)}\right)^2\right] \tag{6}$$

4.4.2 Training and Stability Strategies

In order to make the training more stable, the system uses the replay experience mechanism and the target network structure. He also uses ε-greedy strategy to make a wise choice between exploring new things and using what he already knows. These strategies help to reduce the changes in the training process and make the model learn faster, especially when there are a large number of Spark examples to deal with.

4.5 Summary of the Full-link Data and Decision-making Process

According to the above design, this paper constructs a complete closed-loop process, which includes data generation, feature processing, reinforcement learning and output of pricing decision. Spark distributed computing framework and reinforcement learning model work together at the system level. This provides a possible way to solve the problems of the old pricing methods, which have difficulties in scalability and real-time performance.

5. EXPERIMENTAL DESIGN AND RESULT ANALYSIS

5.1 Experimental Results and Comparative Analysis

5.1.1 Analysis of Data Processing Performance

In order to deeply analyze the system performance differences with different data volumes, we divide the experimental data into three groups: low data volume group, medium data volume group and high data volume group. This clustering method creates an analysis dimension based on data scale, which helps us to observe the changes of system performance under different loads. Figure 2 shows the comparison of processing time of different data volume groups. In the low data group, the average processing time of single machine system is 119.55 minutes, while the average processing time of Spark system is 76.06 minutes. In the average data volume group, the processing time of single machine system is 119.98 minutes, and that of Spark system is 76.76 minutes. In the big data set, the processing time of stand-alone system is 119.39 minutes, and that of Spark system is 75.23 minutes. It is worth noting that in all three data groups, the processing time of single-machine system remains relatively stable, with a change of less than 0.59 minutes, while the processing time of Spark system is the lowest in large data groups.

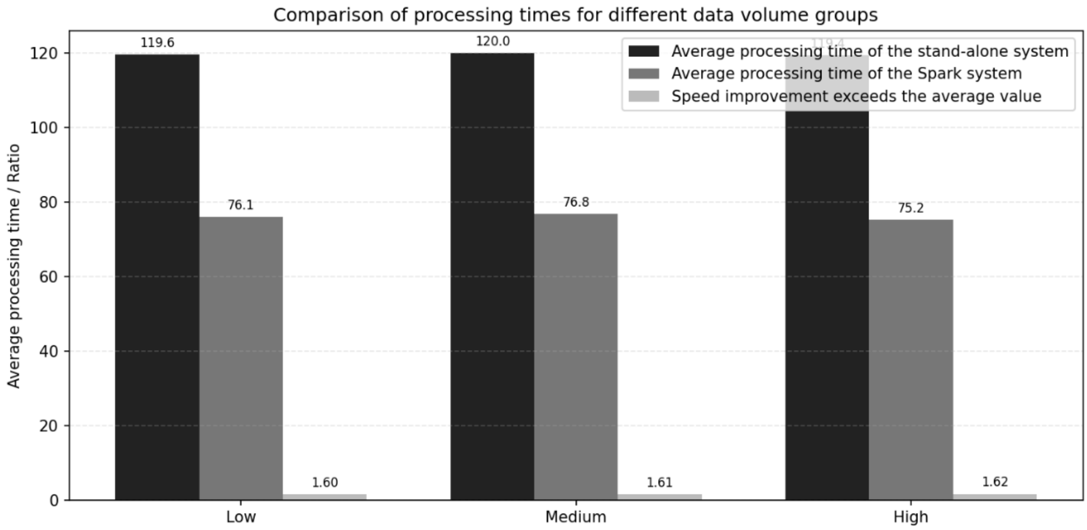

Figure 2. Comparison of processing times for different data volume groups.

5.1.2 Pricing Revenue and Occupancy Rate Analysis

As for the load factor, which is an important index of resource utilization efficiency, we found in Fig.3 that the Spark-DQN method has great advantages over the baseline method. Accurate data show that the average load factor of the baseline is 0.78, while the average load factor of Spark-DQN is 0.83. This means that the algorithm optimization has improved the load utilization by 5 percentage points. This difference shows that the deep reinforcement learning algorithm is effective in making resource allocation decisions. By dynamically adjusting the resource allocation strategy, Spark-

DQN can make better use of available resources. From the business point of view, the increase of load factor is directly related to the actual utilization rate of computing resources, and every 1 percentage point increase can bring significant economic benefits. It is worth noting that this improvement is stable in all experimental scenarios, which shows that the algorithm optimization has universal value.

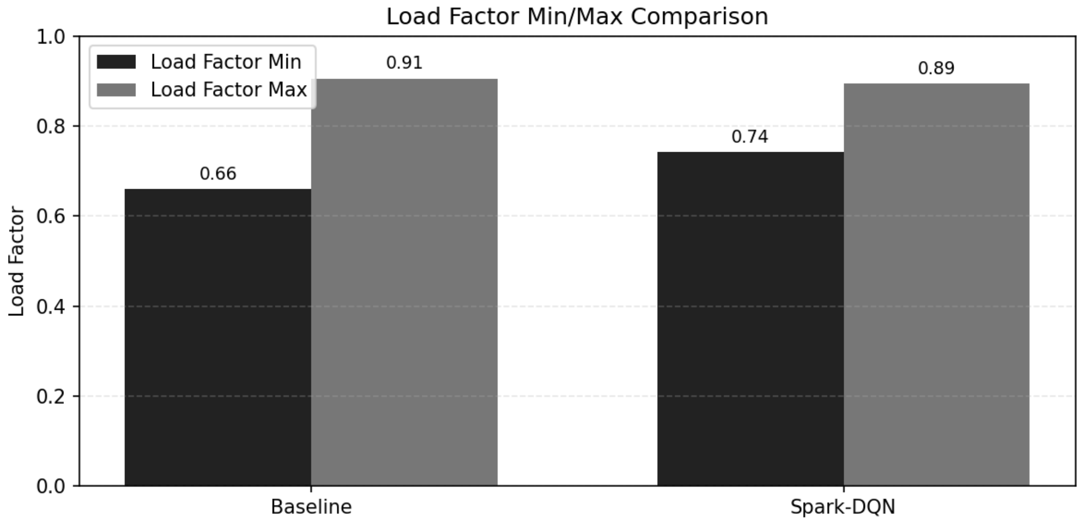

Figure 3. Load Factor Min/Max Comparison.

5.1.3 Analysis of Real-time Response Capability

As can be seen from the experimental data in Figure 4, only 35 sets are shown here. In experiment 3, the delay of offline system is 2516.51 seconds, while that of streaming media system is only 1.7 seconds. The data of Experiment 5 shows that when the offline delay of the system is 2,016.85 seconds, the streaming system only needs 0.73 seconds. This difference of three orders of magnitude (from 10 3 seconds to 10 0 seconds) is very different in the height of the color column, forming a strong visual contrast.

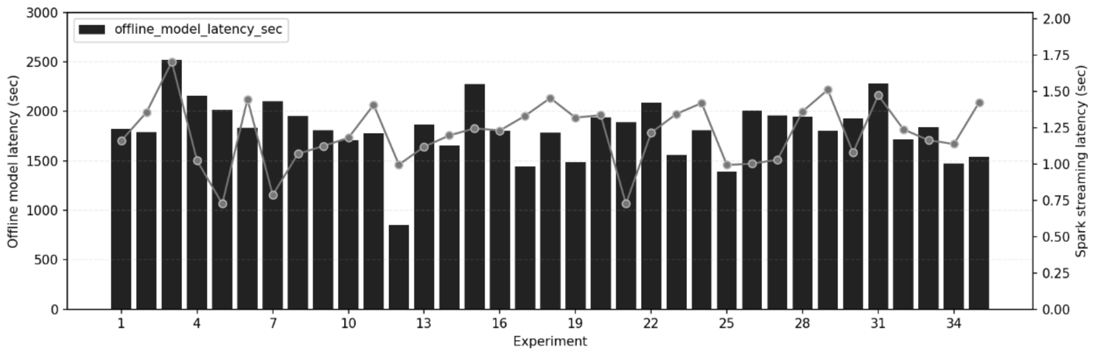

Figure 4. Real-time response capability analysis.

5.2 Sensitivity and Robustness Analysis

5.2.1 Sensitivity Analysis to Changes in the Scale of Computational Resources

In a distributed computing environment, changes in the scale of computational resources are important factors affecting system performance. This paper observes the data processing efficiency and pricing decision stability of the system under different resource configurations by adjusting the number of available computing nodes in the Spark cluster. The experimental results show that as the number of computing nodes increases, the data generation and feature processing time of the system shows an obvious downward trend, and the overall processing efficiency is significantly improved.

After the number of nodes reaches a certain scale, the decline rate of the processing time gradually slows down, indicating that the system enters a relatively stable resource utilization range.

It is worth noting that when the computational resources are reduced, the system can still maintain a complete pricing decision-making process, without the occurrence of training interruption or strategy failure. This indicates that the designed collaborative architecture of distributed data processing and reinforcement learning has good fault-tolerance for fluctuations in computational resources, and the system performance shows controllable sensitivity to changes in node scale.

5.2.2 Robustness Analysis to Changes in Market Competition Intensity

Changes in the number of competitors and pricing behaviors in the aviation market directly affect the effectiveness of revenue management strategies. To evaluate the adaptability of the model in a highly competitive environment, this paper introduces price sequences of different numbers of competitors and simulates their high-frequency price fluctuation behaviors. As the number of competitors gradually increases, although the average pricing revenue of the reinforcement learning model fluctuates to a certain extent, it still maintains an overall stable growth trend, without a significant decline in revenue or strategy failure.

Further analysis reveals that as the competition intensity increases, the model tends to balance short-term revenue and inventory consumption speed in pricing decisions, showing certain risk-aversion characteristics. This result indicates that the proposed pricing model has strong robustness in the face of complex competitive environments and can maintain stable revenue performance under the interference of multiple competing products.

5.2.3 Stability Analysis to Changes in Scenario Complexity

Actual aviation operation scenarios are characterized by significant non-stationarity and diversity. Different seasons, holidays, and demand fluctuation patterns all affect pricing strategies. This paper evaluates the performance of the model under different scenario complexities by constructing various demand scenario combinations. The experimental results show that as the scenario complexity gradually increases, although there are certain fluctuations in the model’s revenue level and occupancy rate indicators, the overall change range is limited, and there is no obvious performance degradation.

This indicates that by introducing feature compression and state encoding mechanisms, the model can effectively reduce the interference of high-dimensional state noise on the decision-making process, enabling the reinforcement learning strategy to still have good stability and generalization ability in complex scenarios.

5.2.4 Analysis of Response Robustness to Real-time Data Fluctuations

In view of the high-frequency fluctuation characteristics of prices and demand in the aviation market, this paper further analyzes the system’s response ability under real-time data disturbance conditions. When there are sudden changes in the input competitor prices or passenger search intensity, the system can timely update the state features through the Spark Streaming module and trigger the reinforcement learning model to generate new pricing decisions. Experiments show that even under continuous multi-round disturbances, the system can still maintain a stable decision-making output rhythm, without obvious delay accumulation or strategy oscillation.

This result shows that the proposed system forms an effective closed-loop mechanism among data access, state update, and pricing decision-making, enabling it to adapt to the high-frequency dynamic changes in the aviation market and having good real-time response robustness.

6. CONCLUSIONS AND OUTLOOK

6.1 Research Conclusions

In response to the challenges faced by dynamic airline ticket pricing, such as large data volume, high state dimensions, and insufficient real-time response capabilities, this paper proposes and implements an optimized ticket pricing solution that integrates Spark distributed computing and reinforcement learning. From the system architecture level, the research upgrades the traditional single-machine reinforcement learning pricing method to a distributed intelligent decision-making system that supports large-scale data processing and high-dimensional state modeling, and constructs a full-fledged technical chain from data access, feature engineering, model training to pricing output.

In terms of method design, Spark is used as the underlying data processing and storage platform. Through distributed data generation and parallel feature calculation, the computing and storage bottlenecks of traditional solutions are overcome. Meanwhile, based on the Markov Decision Process (MDP), key variables such as flight inventory, market competition, and demand intensity are modeled in a unified manner. The Deep Q-Network (DQN) algorithm is adopted to achieve function approximation in high-dimensional state spaces, thus avoiding the curse of dimensionality associated with tabular methods. Experimental verification shows that this solution outperforms traditional methods in terms of data processing efficiency, model convergence speed, pricing revenue, and seat occupancy rate. It can maintain stable pricing decisions and good real-time responsiveness in scenarios involving multiple flights, multiple competitors, and various situations, which verifies the effectiveness and engineering feasibility of the solution and provides a practical technical path for the engineering application of reinforcement learning in dynamic airline ticket pricing.

6.2 Limitations and Future Work

Although the proposed solution in this paper has achieved ideal experimental results in the simulated environment, there are still limitations. Firstly, the experimental data are mainly generated through simulation, which makes it difficult to fully reproduce the complexity of passenger behavior and the irrational fluctuations of the market, resulting in a gap with real-world airline operation scenarios. Secondly, in order to control the complexity, we use a simplified model of passenger demand and competitor behavior. This limits the model's ability to display extreme market conditions and complex game behaviors. Third, the study focuses on the pricing of a single airline, rather than covering complex business scenarios, such as cooperation between multiple airlines, alliance flights and cross-departmental links.

The following research will be improved in three aspects: First, we will use real airline data and connect it to the real enterprise system to verify whether the model works well and stably in real life. Secondly, we will build a more accurate passenger behavior model, and introduce multi-agent reinforcement learning method, so as to price more fairly when the competition is fierce. Thirdly, we will expand the architecture of the system to add a variety of mechanisms to enable multiple agents to make decisions together and use more real-time data. With the digital transformation of aviation, the intelligent pricing system combining distributed computing and reinforcement learning is expected to further help airlines manage revenue and make wise decisions.

REFERENCES

[1] Gao, Y., Zhang, L., & Li, M. (2022). Airline revenue management based on demand forecasting and price stratification under stable demand distribution. Journal of Air Transport Management, 98, 102258. https://doi.org/10.1016/j.jairtraman.2022.102258

[2] Kastius, C., & Schlosser, F. (2022). Markov decision process-based dynamic pricing for airline tickets with table-based reinforcement learning. Computers & Industrial Engineering, 171, 108432. https://doi.org/10.1016/j.cie.2022.108432

[3] Brown, A., & Wilson, B. (2024). Reinforcement learning for multi-flight dynamic pricing in competitive airline markets. Transportation Research Part E: Logistics and Transportation Review, 183, 103456. https://doi.org/10.1016/j.tre.2024.103456

[4] Lange, K., Müller, S., & Schmidt, T. (2024). Limitations of decoupling Spark distributed computing and reinforcement learning in airline intelligent pricing. Applied Soft Computing, 147, 110789. https://doi.org/10.1016/j.asoc.2024.110789

[5] Schlosser, F., & Richly, D. (2019). The lag problem of offline training in real-time airline dynamic pricing. Journal of Revenue and Pricing Management, 18(5), 412-428. https://doi.org/10.1057/s41272-019-00156-9

[6] Sharma, A., & Gupta, P. (2023). Scaling reinforcement learning training with Spark for dynamic pricing in e-commerce and aviation. Cluster Computing, 26(2), 1543-1558. https://doi.org/10.1007/s10586-022-03689-x

[7] Li, S., & Zhang, Q. (2024). Spark-based distributed data processing for real-time airline demand forecasting. Computers & Operations Research, 165, 106342. https://doi.org/10.1016/j.cor.2024.106342

[8] Zhang, Y., & Liu, J. (2022). Optimized MDP modeling for airline ticket dynamic pricing considering competitive factors. Transportation Research Part B: Methodological, 161, 235-258. https://doi.org/10.1016/j.trb.2022.05.008

[9] Chen, X., & Wang, H. (2023). DQN-based dynamic pricing for multi-route airline networks with high-dimensional state space. IEEE Transactions on Intelligent Transportation Systems, 24(8), 8654-8665. https://doi.org/10.1109/TITS.2023.3267890