International Journal of Artificial Intelligence,
ISSN 0974-0635; Int. J. Artif. Intell.
2024, Volume 22, Issue Number 2

A Deep Reinforcement Learning Approach to Dynamic Airline Ticket Pricing and Customer Response Analysis

Youness BOUTYOUR [1] **and Abdellah IDRISSI** [2]

Intelligent Processing Systems and Security (IPSS) Team
Mohammed V University
Rabat, MOROCCO
[1] youness.boutyour@um5r.ac.ma

[2] idrissi@um5r.ac.ma

ABSTRACT

In the ever-changing landscape of airline ticket sales, efficient dynamic pricing strategies are crucial for maximizing revenue while catering to diverse customer preferences. This paper explores the application of Deep Reinforcement Learning (DRL) algorithms, namely REINFORCE, PPO, A2C, SAC, and TD3, in the context of airline ticket pricing. Leveraging a synthetic dataset and a Generalized Linear Model, these algorithms were rigorously evaluated. Our study reveals that TD3 outperforms other models, showcasing rapid convergence and robust reward optimization capabilities. We also provide a comparative analysis of training times, essential for practical implementation. Through extensive experimentation and computational analysis, this research contributes valuable insights into the efficacy of DRL techniques in dynamic pricing. The findings not only offer benchmarks for airline industry applications but also illuminate the broader potential of advanced machine learning methods in revenue management across various sectors. This study underscores the pivotal role of artificial intelligence in shaping the future of pricing strategies, providing a roadmap for businesses aiming to stay competitive in today's dynamic markets.

Keywords: Artificial Intelligence, Deep Reinforcement Learning, Revenue Optimization, Dynamic Ticket Pricing.

2012 Computing Classification System: Computing methodologies →Machine learning →Learning paradigms →Reinforcement learning →Sequential decision making.

1 Introduction

The aeronautical industry is marked by its inherent complexity, with airlines continuously striving to optimize their operations for maximum efficiency and profitability (Talluri and Ryzin, 2004). Among the multifaceted challenges faced by airlines, determining optimal ticket prices stands as a pivotal yet intricate task (Wittman and Belobaba, 2018; Betancourt, Hortaçsu, Oery and Williams, 2022). Traditional approaches to pricing, although rooted in statistical models, struggle to adapt swiftly to the dynamic nature of market demands (Abdella, Zaki, Shuaib and Khan, 2021). In the contemporary landscape, where technology revolutionizes business

paradigms, airlines seek innovative solutions to enhance their revenue strategies. This research delves into the realm of Dynamic Pricing, a sophisticated strategy that tailors prices based on real-time market dynamics (Abdella et al., 2021). Unlike conventional methods, dynamic pricing responds promptly to shifts in demand, thereby enabling airlines to capitalize on fluctuations and maximize revenue potential (Shukla, Kolbeinsson, Otwell, Marla and Yellepeddi, 2019). Leveraging the power of Reinforcement Learning (RL), a subset of artificial intelligence (Silver, Huang, Maddison, Guez, Sifre, van den Driessche, Schrittwieser, Antonoglou, Panneershelvam, Lanctot, Dieleman, Grewe, Nham, Kalchbrenner, Sutskever, Lillicrap, Leach, Kavukcuoglu, Graepel and Hassabis, 2016), this study explores how deep reinforcement learning algorithms can enhance airline ticket pricing strategies. The fusion of Dynamic Pricing and Reinforcement Learning offers a promising avenue for airlines to optimize their revenue streams in an increasingly competitive market. By infusing intelligence into pricing decisions, airlines can anticipate customer behavior, adapt to market changes, and strategically allocate ticket prices. This research aims to unravel the intricate interplay between dynamic pricing, artificial intelligence, and the aeronautical industry, providing valuable insights that can revolutionize the way airlines approach revenue management (Talluri and Ryzin, 2004; Cheng, Zou, Zhuang, Liu, Xu and Zhang, 2019).

1.1 Research Problem Statement

Dynamic pricing in the airline industry, guided by real-time customer characteristics, presents a multifaceted research problem. This problem revolves around the development and optimization of deep reinforcement learning algorithms to tailor ticket prices based on a comprehensive understanding of customer features, ensuring maximized revenue while accommodating evolving market dynamics.

1.2 Objectives of the Study

This research pursues several key objectives:

- To develop and implement deep reinforcement learning algorithms for dynamic pricing of airline tickets that consider a wide range of customer characteristics.
- To analyze the performance of these algorithms in terms of revenue optimization and speed of convergence.
- To provide valuable insights into the integration of artificial intelligence, dynamic pricing, and the aviation industry, offering a foundation for airlines to enhance their revenue management strategies.

1.3 Structure of the paper

The structure of this paper is organized in order to facilitate a comprehensive exploration of the dynamic pricing using deep reinforcement learning for airline tickets. In section 2, we provide an in-depth review of the existing literature, examining the evolution of pricing strategies

in the aviation industry. Section 3 dives into the methodology, elucidating the integration of reinforcement learning algorithms into dynamic pricing models. We detail the data sources and experimental setup in section 4, which precedes the presentation and discussion of our findings in section 5. Finally, the paper concludes in section 7, summarizing the key takeaways and outlining avenues for future research in the dynamic pricing landscape.

2 Literature Review

2.1 Historical Pricing Strategies in the Aeronautical Industry

The aviation industry has a long history of employing various pricing strategies to maximize revenue. This section explores some of the traditional pricing methods used in the past (ROOS, MILLS and WHELAN, 2010).

2.1.1 Traditional Fixed Pricing

One of the earliest pricing strategies in the airline industry was fixed pricing, where ticket prices remained constant regardless of the time of booking or the remaining number of seats. This fixed pricing strategy can be represented as:

$$P_{\text{fixed}} = \text{Constant} \tag{2.1}$$

Where P_{fixed} represents the fixed ticket price.

While fixed pricing provided simplicity, it often resulted in suboptimal revenue generation, especially during peak demand periods (Zhang, 2021).

2.1.2 Yield Management and Seat Inventory Control

The introduction of yield management revolutionized pricing strategies in the aviation industry (Justin, Payan and Mavris, 2021). Yield management involves dynamically adjusting ticket prices based on various factors such as booking lead time, demand forecasts, and seat availability. The core concept of yield management can be mathematically expressed as:

$$P_{\text{yield}} = f(\text{Demand}, \text{Lead Time}, \text{Seat Availability}) \tag{2.2}$$

Where:

- P_{yield} represents the dynamically adjusted ticket price.
- Demand denotes the expected demand for flights.
- Lead Time is the time remaining until the flight's departure.
- Seat Availability indicates the number of available seats on the flight.

Yield management techniques improved revenue optimization by aligning prices with market demand and flight occupancy (Gabor, Kardos and Oltean, 2022).

2.1.3 Competitive Pricing Strategies

In a highly competitive industry like aviation, airlines often engage in competitive pricing strategies to attract passengers. Competitive pricing involves setting ticket prices based on the fares offered by rival airlines for similar routes (Wang, Zhang and Zhang, 2018). This strategy can be mathematically represented as:

$$P_{\text{competitive}} = \text{Competitor's Fare} \pm \text{Markup} \tag{2.3}$$

Where:

- $P_{\text{competitive}}$ represents the ticket price based on competitive pricing.
- Competitor's Fare is the fare offered by a rival airline for a comparable route.
- Markup represents the additional price added to the competitor's fare.

Competitive pricing aims to capture market share by offering competitive fares, and the markup is determined strategically to balance revenue and market positioning (Asker, Fershtman and Pakes, 2022; Betancourt et al., 2022).

2.2 Reinforcement Learning

The method of reinforcement learning, we intend to employ, is founded on the Markov Decision Process (MDP) as elucidated in Sutton's seminal work (Sutton and Barto, 2018), which is visually represented in Fig. 1 (Boutyour and Idrissi, 2023).
In reinforcement learning, the agent isn't given specific guidance on the actions to execute. Instead, it learns to identify the most rewarding actions through a process of trial and error (Sarhadi, Akbari and Karimi, 2022). Our objective at each time step is to determine a policy π that maximizes the expected cumulative reward along the trajectory, as given by the value function:

$$V(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i R_{t+i+1} \,\middle|\, S_t = s\right] \tag{2.4}$$

To achieve this, we have to assess the current policy and improve it as necessary (Sutton and Barto, 2018; Colpas, Patricia, Carrascal, Isabel, Aziz, Melo and Alberto, 2023).

Figure 1: Illustration depicting the dynamic interplay between the agent and its environment.

2.3 Policy Evaluation

Our initial objective is to evaluate the value function under a specified policy π. This evaluation is performed according to the Bellman equation (Sutton and Barto, 2018), which is expressed as follows:

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s \right] \tag{2.5}$$

We can iteratively compute v_π using the following update equation:

$$v_{k+1}(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s \right] \tag{2.6}$$

This iterative process is guaranteed to converge to the true value function for any initial value function (Puterman, 2014; Sutton and Barto, 2018).

2.4 Policy Improvement

After evaluating the value function, our goal is to improve the policy accordingly. To achieve this, we introduce the q-function under policy π (Sutton and Barto, 2018):

$$q_\pi(s, a) = \mathbb{E} \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a \right] \tag{2.7}$$

The Policy Improvement Theorem asserts that for any pair of policies π and π', and for all states s, if $q_\pi(s, \pi'(s)) \geq v_\pi(s)$, then it follows that $v_{\pi'}(s) \geq v_\pi(s)$. Consequently, we have the opportunity to enhance the policy in the following manner:

$$\pi'(s) = \underset{a}{\text{argmax}}\, q_\pi(s, a) \tag{2.8}$$

This policy improvement process is performed for all states (Sutton and Barto, 2018).

2.5 Policy Iteration

Through a repetitive process of policy evaluation and policy improvement, we acquire a series of consistently enhancing policies and value functions (Sutton and Barto, 2018):

$$\pi_0 \xrightarrow{\text{Evaluate}} v_{\pi_0} \xrightarrow{\text{Improve}} \pi_1 \xrightarrow{\text{Evaluate}} v_{\pi_1} \xrightarrow{\text{Improve}} \dots \xrightarrow{\text{Improve}} \pi_* \xrightarrow{\text{Evaluate}} v_* \tag{2.9}$$

Here, the arrow $\xrightarrow{\text{Evaluate}}$ indicates a policy evaluation step, while $\xrightarrow{\text{Improve}}$ denotes a policy improvement step. The entire process is known as policy iteration.

2.6 Value Iteration

A drawback of policy iteration lies in its necessity for policy evaluation in every iteration, a process that tends to be computationally demanding, particularly in scenarios with extensive state spaces. To mitigate this, we can truncate policy evaluation after just one sweep (Sutton and Barto, 2018), resulting in value iteration.
In value iteration, the update formula for the value function is as follows:

$$\begin{aligned} v_{k+1}(s) &= \max_a q(s,a) \\ &= \max_a \mathbb{E}\left[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a\right] \end{aligned} \tag{2.10}$$

This iterative process is repeated until the value function converges to v_*. Subsequently, the final policy, denoted as $\pi(s) = \operatorname{argmax}_a q(s,a)$, is then recorded.

2.7 Generalized Policy Iteration

Generalized policy iteration (GPI), a fundamental concept in reinforcement learning, emphasizes the idea of letting policy evaluation and policy improvement interact independently, regardless of the granularity of the two processes, as opposed to ensuring each process completes before the other begins1. This approach has influenced the development of modern reinforcement learning algorithms, enabling more flexible and efficient learning paradigms in dynamic environments. By decoupling policy evaluation and improvement, GPI methods have paved the way for sophisticated algorithms that can adapt rapidly to changing circumstances, a crucial feature in complex and dynamic domains like airline ticket pricing (Busoniu, Babuska, De Schutter and Ernst, 2017).

2.8 Temporal-Difference Learning

In cases where we lack a complete model of the environment, we need to estimate the value function based solely on experience. Two common methods for this purpose are Monte Carlo methods and Temporal-Difference (TD) learning (Sutton and Barto, 2018). In this paper, we will utilize the Temporal-Difference method.

To estimate $v_\pi(S_t)$ using only experience, we take the average over all experienced trajectory values. Temporal-Difference learning employs $G_i = [R_{t+1}]_i + \gamma[v_\pi(S_{t+1})]_i$ to estimate the value of the i-th trajectory starting at S_t, with v_π being the estimated value function at the i-th time S_t is visited. This value function can be initialized to any value at the start. We have the incremental formula for updating the value function (Sutton and Barto, 2018):

$$v_\pi(S_t) \leftarrow v_\pi(S_t) + \alpha[R_{t+1} + \gamma v_\pi(S_{t+1}) - v_\pi(S_t)] \tag{2.11}$$

This formula also applies to the q-function (Sutton and Barto, 2018):

$$q_\pi(S_t, A_t) \leftarrow q_\pi(S_t, A_t) + \alpha[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) - q_\pi(S_t, A_t)] \tag{2.12}$$

2.9 Policy Gradient Methods

In scenarios where state spaces are exceedingly large, the task of identifying an optimal policy or determining the optimal value function turns impractical, constrained by the limitations of available resources and time constraints. Instead of maintaining a tabular representation of the value function and action values, policy gradient methods utilize parameterized functions, such as artificial neural networks (ANNs), to approximate these functions (Sutton and Barto, 2018).

Policy gradient approaches center their attention on acquiring a parameterized policy that directly makes action choices without reliance on a value function for guidance. Although it remains an option to employ a value function for policy parameter learning, it is noteworthy that such utilization is not mandatory for the process of action selection.
The policy $\pi(a|s,\boldsymbol{\theta})$ and value function $v_\pi(s,\boldsymbol{w})$ are parameterized by vectors $\boldsymbol{\theta}$ and $\boldsymbol{w}$, respectively. The performance measure (objective) is defined as:

$$J(\boldsymbol{\theta}) = \begin{cases} v_{\pi_{\boldsymbol{\theta}}}(s_0) & \text{for episodic tasks} \\ \lim_{t\to\infty} \mathbb{E}[R_t|S_0, A_{0:t-1} \sim \pi] & \text{for continuing tasks} \end{cases} \tag{2.13}$$

Policy gradient methods aim to maximize this performance measure by performing updates that approximate gradient ascent in J (Sutton and Barto, 2018)

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\widehat{\nabla J(\boldsymbol{\theta}_t)} \tag{2.14}$$

In this context, $\widehat{\nabla J(\boldsymbol{\theta}_t)}$ represents a probabilistic approximation, the expected value of which closely estimates the gradient of the performance metric in relation to $\boldsymbol{\theta}_t$ (Sutton, McAllester, Singh and Mansour, 1999).

2.9.1 The Policy Gradient Theorem

The policy gradient theorem, a fundamental concept in reinforcement learning, has been explored extensively in recent research (Sutton and Barto, 2018). It states that:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla\pi(a|s,\boldsymbol{\theta}) \tag{2.15}$$

Here, μ is the stationary distribution of states. This leads to the following expression:

$$\begin{aligned} \nabla J(\boldsymbol{\theta}) &\propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla\pi(a|s,\boldsymbol{\theta}) \\ &= \mathbb{E}_\pi \left[\sum_a q_\pi(S_t,a) \nabla\pi(a|S_t,\boldsymbol{\theta}) \right] \\ &= \mathbb{E}_\pi \left[q_\pi(S_t,A_t) \frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \right] \end{aligned} \tag{2.16}$$

This principle is further extended to encompass a baseline function $b(s)$ (Schulman, Wolski, Dhariwal, Radford and Klimov, 2017):

$$\begin{aligned} \nabla J(\boldsymbol{\theta}) &\propto \sum_s \mu(s) \sum_a \left(q_\pi(s,a) - b(s)\right) \nabla\pi(a|s,\boldsymbol{\theta}) \\ &= \mathbb{E}_\pi \left[\sum_a \left(q_\pi(S_t,a) - b(S_t)\right) \frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \right] \end{aligned} \tag{2.17}$$

This discounted version applies as well:

$$\nabla J(\boldsymbol{\theta}) \propto \mathbb{E}_\pi \left[\gamma^t \left(q_\pi(S_t,A_t) - b(S_t)\right) \frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \right] \tag{2.18}$$

A logical starting point is to use an estimate of the state value, denoted $\hat{v}(S_t, \boldsymbol{w})$, in which $\boldsymbol{w}$ represents the weight vector to be learned.

Algorithm 1 Proximal Policy Optimization (PPO) Algorithm

Input: Policy function $\pi(a|s, \boldsymbol{\theta})$, Value function estimator $\hat{v}(s, \boldsymbol{w})$
Define step size $\alpha > 0$, Clipping parameter $\epsilon > 0$
Initialize Value function parameters $\boldsymbol{w} \leftarrow \mathbf{0}$, Policy parameters $\boldsymbol{\theta} \leftarrow \mathbf{0}$
loop (for each iteration)
    Collect set of transitions using policy $\pi(\cdot|s, \boldsymbol{\theta})$
    Compute advantage estimates $A(s, a) = R(s, a) - \hat{v}(s, \boldsymbol{w})$ ▷ Advantage calculation for policy improvement
    Optimize value function $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \nabla_v L_v(\boldsymbol{w})$ ▷ Updating value function for better estimation
    Calculate policy loss $L(\boldsymbol{\theta}) = \sum_t \min\left(\frac{\pi(a_t|s_t,\boldsymbol{\theta})}{\pi_{\text{old}}(a_t|s_t,\boldsymbol{\theta})} A(s_t, a_t), \text{clip}(\epsilon, 1-\epsilon) A(s_t, a_t)\right)$ ▷ Policy loss includes clipping for stable updates
    Update policy $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ ▷ Policy parameters update using gradient ascent
end loop

2.9.2 Actor-Critic Methods

Recalling the Temporal-Difference method, we use $q_\pi(s, a) = \mathbb{E}[R_{t+1} + \gamma \hat{v}(S_{t+1}, \boldsymbol{w}) - \hat{v}(S_t, \boldsymbol{w}_t) | S_t = s, A_t = a]$ (Sutton and Barto, 2018). By taking a single-sample estimate and selecting $\hat{v}(S_t, \boldsymbol{w})$ as the baseline, we have the following update equation for $\boldsymbol{\theta}$:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha_{\boldsymbol{\theta}} \left(R_{t+1} + \gamma \hat{v}(S_{t+1}, \boldsymbol{w}) - \hat{v}(S_t, \boldsymbol{w}_t)\right) \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta}_t) \\ &= \boldsymbol{\theta}_t + \alpha_{\boldsymbol{\theta}} \delta_t \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta}_t) \end{aligned} \tag{2.19}$$

As for the weight vector $\boldsymbol{w}$ used for the estimated value function, at each time step t, we aim to update it by minimizing the squared error $\left[v_\pi(S_t) - \hat{v}(S_t, \boldsymbol{w}_t)\right]^2$:

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \alpha'_{\boldsymbol{w}} \nabla \left[v_\pi(S_t) - \hat{v}(S_t, \boldsymbol{w}_t)\right]^2 \\ &= \boldsymbol{w}_t - \alpha_{\boldsymbol{w}} \left(v_\pi(S_t) - \hat{v}(S_t, \boldsymbol{w}_t)\right) \nabla \hat{v}(S_t, \boldsymbol{w}_t) \end{aligned} \tag{2.20}$$

Since $v_\pi(s) = \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$ (Sutton and Barto, 2018), taking a single-sample estimate gives us the update equation for $\boldsymbol{w}$:

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \alpha_{\boldsymbol{w}} \left(R_{t+1} + \gamma \hat{v}(S_{t+1}, \boldsymbol{w}) - \hat{v}(S_t, \boldsymbol{w}_t)\right) \nabla \hat{v}(S_t, \boldsymbol{w}_t) \\ &= \boldsymbol{w}_t - \alpha_{\boldsymbol{w}} \delta_t \nabla \hat{v}(S_t, \boldsymbol{w}_t) \end{aligned} \tag{2.21}$$

Actor-critic methods are based on these update formulas, as described in the provided algorithm 2.

Algorithm 2 Actor-Critic Algorithm

Input: Policy function $\pi(a|s, \boldsymbol{\theta})$ parameterized by differentiable variables
Input: State-value function estimator $\hat{v}(s, \boldsymbol{w})$ with differentiable variables
Define learning rates $\alpha_{\boldsymbol{w}} > 0$, $\alpha_{\boldsymbol{\theta}} > 0$
Set Initial parameters $\boldsymbol{\theta} \leftarrow \mathbf{0}$, $\boldsymbol{w} \leftarrow \mathbf{0}$
repeat (for each episode)
    Set starting state S (beginning of the episode)
    repeat (for each step t within the episode)
        Select A from $\pi(\cdot|S, \boldsymbol{\theta})$
        Execute action A, observe next state S' and reward R
        Calculate TD error $\delta \leftarrow R + \gamma\hat{v}(S', \boldsymbol{w}) - \hat{v}(S, \boldsymbol{w})$
        Update critic $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_{\boldsymbol{w}}\delta\nabla\hat{v}(S, \boldsymbol{w})$
        Update actor $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}}\gamma^t\delta\nabla \ln \pi(A|S, \boldsymbol{\theta})$
        Transition to new state $S \leftarrow S'$
    until end of episode
until convergence or specified condition

Algorithm 3 REINFORCE Algorithm

Input: Policy function with differentiable parameters $\pi(a|s, \boldsymbol{\theta})$
Define step size $\alpha_{\boldsymbol{\theta}} > 0$
Set Initial policy parameter $\boldsymbol{\theta} \leftarrow \mathbf{0}$
for each episode **do**
    Set initial state S
    while not end of episode **do**
        Select action a based on $\pi(\cdot|S, \boldsymbol{\theta})$
        Execute action a, observe next state S' and reward R
        Update $\boldsymbol{\theta}$ as $\boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}}\gamma^t R\nabla \ln \pi(a|S, \boldsymbol{\theta})$
        Update state $S \leftarrow S'$
    end while
end for

3 Methodology

3.1 Problem Formulation

In this section, we formally define the problem of dynamic pricing for airline tickets, considering customer features and customer responses denoted $response$ after pricing.

3.1.1 Definition of the Pricing Problem

Dynamics pricing constitutes a strategic pricing approach implemented by airlines to enhance revenue generation through the adaptive modulation of ticket prices, guided by an array of multifaceted factors. The overarching aim of this endeavor is to ascertain the most advantageous

fare for each flight, predicated upon a comprehensive assessment of factors including the time remaining until departure, seat availability, and the unique characteristics of prospective passengers. Our research primarily aims to establish a reinforcement learning-based approach for optimizing this pricing strategy through the analysis of historical data.

3.1.2 Mathematical Representation for the MDP problem

To mathematically formulate the problem, we define key components:

- State (S_t): The state at each time step includes information about the number of days until the flight, the remaining seats available for sale, and customer features represented by the vector X. This vector X contains 20 entries generated from various probability distributions, such as normal distribution and binomial distribution.
- Action (A_t): The action represents the ticket price chosen for a particular day. It's a continuous variable, allowing for a wide range of pricing decisions.
- Reward (R_t): The reward corresponds to the revenue generated on a given day based on the pricing decision made. It reflects the financial outcome of the pricing strategy.

3.1.3 Features Vector

The features vector X comprises 21 entries, each representing a different customer characteristic. These characteristics are randomly generated and cover a range of factors, including demographic information, past purchasing behavior, and other relevant features. Table 1 provides an example of a feature vector X representing a customer's attributes.

Feature	Distribution	Values
Age	normal	[18,90]
Day_Left	normal	[1,120]
Seat_Left	normal	[0,100]
feature_i	uniform	variable
...	...	...
response	binary	0 or 1

Table 1: Features vector X representing a customer's attributes.

3.1.4 Customer Response

Upon proffering a price, designated as "price," to a client, the ensuing customer response, denoted as "r," is ascertained. This response is characterized as a binary variable, wherein $r = 1$ signifies the acquisition of an airline ticket by the customer, whereas $r = 0$ signifies the absence of a purchase. The generation of this response is contingent upon a generalized

linear model that integrates the customer's normalized attributes and the aforementioned presented price. The determination of the response can be expressed through the subsequent overarching equation:

$$r = f(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \gamma price) \tag{3.1}$$

Where:

- r is the binary response variable (0 or 1).
- $x_1, x_2, \ldots, x_n$ are the normalized customer feature values.
- c is the normalized price.
- $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ are coefficients associated with customer features, with β_0 representing the intercept.
- γ is the coefficient associated with the price.
- $f(\cdot)$ is a link function, typically the logistic function, mapping the linear combination to a probability between 0 and 1.

In this formulation, the customer features have been normalized to have zero mean and unit variance to ensure that they are on a consistent scale for modeling. The coefficients β_i and γ are parameters that can be estimated from historical data to model the relationship between normalized customer features, normalized price, and the likelihood of purchase.
The response is then generated by comparing the calculated probability to a random value from a uniform distribution, where a probability greater than or equal to the random value results in a purchase ($response = 1$), and a probability less than the random value results in no purchase ($response = 0$).
This generalized response model with normalized features allows for effective modeling of customer behavior in response to pricing decisions for airline tickets.

3.2 Generalized Linear Model (GLM) model

In this section, we describe the application of a Generalized Linear Model (GLM) as a fundamental component of our pricing strategy. The GLM serves as a key step in modeling customer responses based on the provided features. Specifically, we choose the logistic function as the link function in our GLM to handle binary responses (0 or 1) efficiently.
The logistic function, denoted as $g(x)$, is commonly used in GLMs for binary classification problems. It transforms the linear combination of predictor variables into probabilities between 0 and 1. The logistic function is defined as:

$$g(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

Where: - x represents the linear combination of predictor variables.
In the context of our pricing problem, we use the logistic function to model the probability of a customer making a purchase ($r = 1$) given the feature vector X and pricing ($price$):

$$P(r = 1|X, price) = \frac{1}{1 + e^{-\beta X - \gamma price}} \tag{3.3}$$

Where: - $P(r = 1|X, price)$ is the probability of a customer making a purchase. - X is the feature vector representing customer characteristics. - $price$ is the price offered to the customer. - β and γ are the model parameters to be estimated during training. Our GLM aims to learn these parameters (β and α) from the training dataset to accurately predict customer responses. The trained GLM will serve as a crucial component in our Dynamic Pricing using DRL algorithms, providing the probability of customer purchase for different pricing scenarios.

3.3 Deep Reinforcement Learning models

In this subection, we delve into the application of DRL models for dynamic pricing in the airline industry. DRL leverages neural networks and reinforcement learning techniques to optimize pricing strategies over time. We explore several DRL algorithms tailored to our pricing problem, including:

- **REINFORCE Model:** We begin with the REINFORCE algorithm, a policy gradient method that aims to maximize the expected cumulative reward. We detail its application to dynamic pricing and discuss the key components such as the policy network (see Algo. 3).
- **Proximal Policy Optimization (PPO) Model:** Next, we investigate the PPO algorithm, which strikes a balance between stability and sample efficiency in policy optimization. We describe how PPO is adapted to the airline pricing problem, emphasizing its advantages(see Algo. 1).
- **Advantage Actor-Critic (A2C) Model:** The A2C algorithm combines elements of policy gradients and value-based methods. We illustrate its utilization in dynamic pricing and explain the actor-critic architecture used for more effective learning (see Algo. 2).
- **Soft Actor-Critic (SAC) Model:** SAC introduces a soft actor-critic framework that optimizes policies with entropy regularization. We explore how this approach enhances exploration and discuss its relevance to pricing decisions (see Algo. 4).
- **Twin Delayed Deep Deterministic Policy Gradients (TD3) Model :** Finally, we examine the TD3 algorithm, which is an extension of the DDPG algorithm designed for continuous action spaces. We showcase its application in the airline pricing domain and highlight its strengths (see Algo. 5).

These DRL models are assessed and compared in terms of their performance, stability, and suitability for dynamic pricing. Each model contributes to our exploration of optimal pricing strategies, using knowledge learned from the trained Generalized Linear Model (GLM) to enhance decision-making.

Algorithm 4 Soft Actor-Critic (SAC) Algorithm

Input: Policy function with parameters $\pi(a|s, \boldsymbol{\theta})$
Input: Parameterized value function $\hat{v}(s, \boldsymbol{w})$
Define learning rates $\alpha_{\boldsymbol{w}} > 0$, $\alpha_{\boldsymbol{\theta}} > 0$
Initialize parameters $\boldsymbol{w} \leftarrow \mathbf{0}$, $\boldsymbol{\theta} \leftarrow \mathbf{0}$
repeat (for each training iteration)
    Retrieve batch (s, a, r, s') from experience replay ▷ Leverage past experiences to improve learning efficiency
    Calculate target value $y = r + \gamma \min_{a' \sim \pi(\cdot|s', \boldsymbol{\theta})} Q(s', a', \boldsymbol{w})$ ▷ Target value estimation with minimization over future actions
    Minimize value function loss using MSE: $\mathcal{L}_{\mathsf{V}} = \mathbb{E}[(y - Q(s, a, \boldsymbol{w}))^2]$ ▷ Optimizing the value function by reducing prediction error
    Generate actions for policy update: $a \sim \pi(\cdot|s, \boldsymbol{\theta})$ ▷ Sampling actions from the policy for the current state
    Formulate policy loss: $\mathcal{L}_{\pi} = \mathbb{E}[\alpha \log(\pi(a|s, \boldsymbol{\theta}) - Q(s, a, \boldsymbol{w}))]$ ▷ Policy improvement by maximizing expected return and entropy
    Apply updates to policy and value function parameters
until convergence or termination criterion met ▷ Iterate until the model sufficiently learns the optimal policy

3.4 Performance Metrics

To assess the performance of our dynamic pricing models, we use two critical performance metrics: the moving average reward generation and convergence speed. By taking into account both of these metrics, we acquire a comprehensive grasp of the algorithms' strengths and weaknesses.

3.4.1 Moving Average Reward

Moving Average Reward is a crucial metric that measures the overall profitability of our pricing strategies. It provides insights into the efficiency of our DRL models in maximizing revenue. We calculate the moving average reward by taking the average of the rewards obtained over a sliding window of episodes. The window size is chosen to smoothen the reward curve and highlight the model's ability to consistently generate higher rewards.

3.4.2 Speed of Convergence

The Speed of Convergence measures how quickly our DRL models adapt to dynamic pricing strategies. It quantifies the number of episodes or iterations required for the model to reach a stable pricing policy with satisfactory performance. Faster convergence is desirable as it allows airlines to quickly respond to changing market conditions and customer behaviors.

Algorithm 5 Twin Delayed Deep Deterministic Policy Gradient (TD3) Algorithm

Input: Policy function with differentiable parameters $\pi(a|s, \boldsymbol{\theta})$
Define step sizes $\alpha_{\boldsymbol{\theta}} > 0$
Initialize policy parameters $\boldsymbol{\theta} \leftarrow \mathbf{0}$
repeat (for each iteration)
    Retrieve a batch of transitions (s, a, r, s') from the replay buffer
    Calculate target value $y = r + \gamma \min(Q_1(s', \pi(s', \boldsymbol{\theta})), Q_2(s', \pi(s', \boldsymbol{\theta})))$
    Minimize mean squared error loss to optimize Q-functions
    Generate action samples for policy update: $a \sim \pi(\cdot|s, \boldsymbol{\theta})$
    Evaluate policy loss: $\mathcal{L}_\pi = -\mathbb{E}[Q_1(s, \pi(s, \boldsymbol{\theta}))]$
    Update the policy parameters using gradient descent
until convergence or termination criterion is met

4 Experimental Results

4.1 Dataset Generation Process

4.1.1 Synthetic Data Generation Methodology

In this subsection, we describe the method used to generate the synthetic dataset employed in our experiments. Generating a realistic dataset is crucial for the evaluation of our pricing strategies using DRL models. We follow a carefully designed procedure to create a dataset that captures various aspects of customer behavior and response to airline ticket prices.
We begin by defining the underlying features that represent each customer. These features include gender, age, days left until the flight, seat availability, time of day (day or night), and 16 different random variables, which incorporate price and other factors affecting customer decisions. These features are represented as an array denoted by $\mathbf{x}$ with 21 entries, each generated randomly from various probability distributions, including normal and binomial distributions.
After generating the feature vectors for customers, we apply a linear formula to simulate customer responses, which is added to our features vector.

4.1.2 Features and Characteristics of the Synthetic Dataset

The dataset is meticulously crafted to mirror real-world airline ticket purchase situations, encompassing an extensive array of features that hold sway over customer decisions.
Several pivotal features incorporated in the dataset comprise:

- Gender: Representing the gender of the customer.
- Age: Reflecting the age of the customer.
- Days Left Until Flight: Indicating the number of days remaining until the flight's departure.
- Seat Availability: Describing the availability of seats on the flight.

- Time of Day: Categorizing the time as day or night.
- 16 Random Variables: Incorporating various factors, including price, that affect customer choices.

Additionally, the dataset includes the customer responses (0 for not purchasing, 1 for purchasing), which are generated based on a linear model. We generate a total of 3000 entries and save the data as a CSV file. This synthetic dataset serves as the foundation for evaluating the performance of our dynamic pricing strategies using DRL models and GLM model as a response predictor for all new customers.

4.2 Experimental Setup

In this subsection, we describe the experimental methodology employed to assess the performance of our models based on DRL algorithms. Our experiments are conducted as shown in Fig. 2.

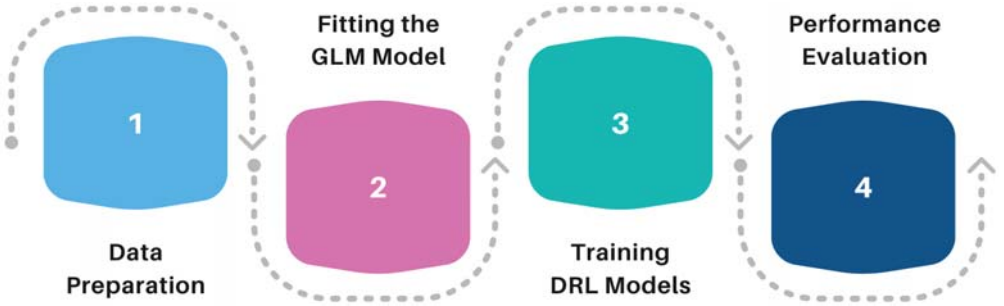


Figure 2: Our experimental process.

4.2.1 Models:

Our experiments are conducted in a systematic manner to rigorously evaluate the effectiveness of our proposed pricing strategies. Figure 2 shows our experimental process, which is structured as follows:

1. **Data Preparation**: We utilize the synthetic dataset generated in Section 4.4.1 as the basis for our experiments. This dataset includes customer features and their corresponding responses.
2. **Fitting the GLM Model**: Before training the DRL models, we fit a Generalized Linear Model (GLM) to the dataset. The GLM helps map customer features to responses using the chosen link function, which in our case is the logistic function.
3. **Training DRL Models**: Building upon the Generalized Linear Model (GLM) as a foundational reference, our next step involves the training of a suite of DRL models, encompassing REINFORCE, PPO, A2C, SAC, and TD3. These models are systematically trained

on our synthetic dataset, equipped with comprehensive customer features and their corresponding responses. The central objective of this training process is to empower the models with the ability to make dynamic pricing decisions while prioritizing revenue maximization.

4. **Performance Evaluation**: We measure and compare the moving average reward generated by each pricing model. Additionally, we analyze the convergence speed of the DRL models.

Through these experiments, we aim to demonstrate the capabilities of our DRL-based pricing strategies in optimizing revenue for airline ticket sales while considering various customer features.

	PPO	A2C	REINFORCE	TD3	SAC
Hyper-parameters	actor_lr = 1e-4 critic_lr = 1e-4 gamma = 0.99 num_itr = 3000 episode_size = 300 batch_size = 5 num_updates_per_itr = 5 clip = 0.5	actor_lr = 1e-4 critic_lr = 1e-4 gamma = 0.99 num_itr = 3000 episode_size = 300	actor_lr = 1e-4 gamma = 0.99 num_itr = 3000 episode_size = 300	actor_lr = 1e-4 critic_lr = 1e-4 gamma = 0.99 num_itr = 3000 episode_size = 300	actor_lr = 1e-4 critic_lr = 1e-4 gamma = 0.99 num_itr = 3000 episode_size = 300 polyak = 0.995

Table 2: Hyper-parameters used for the experiments.

4.2.2 Training and Evaluation

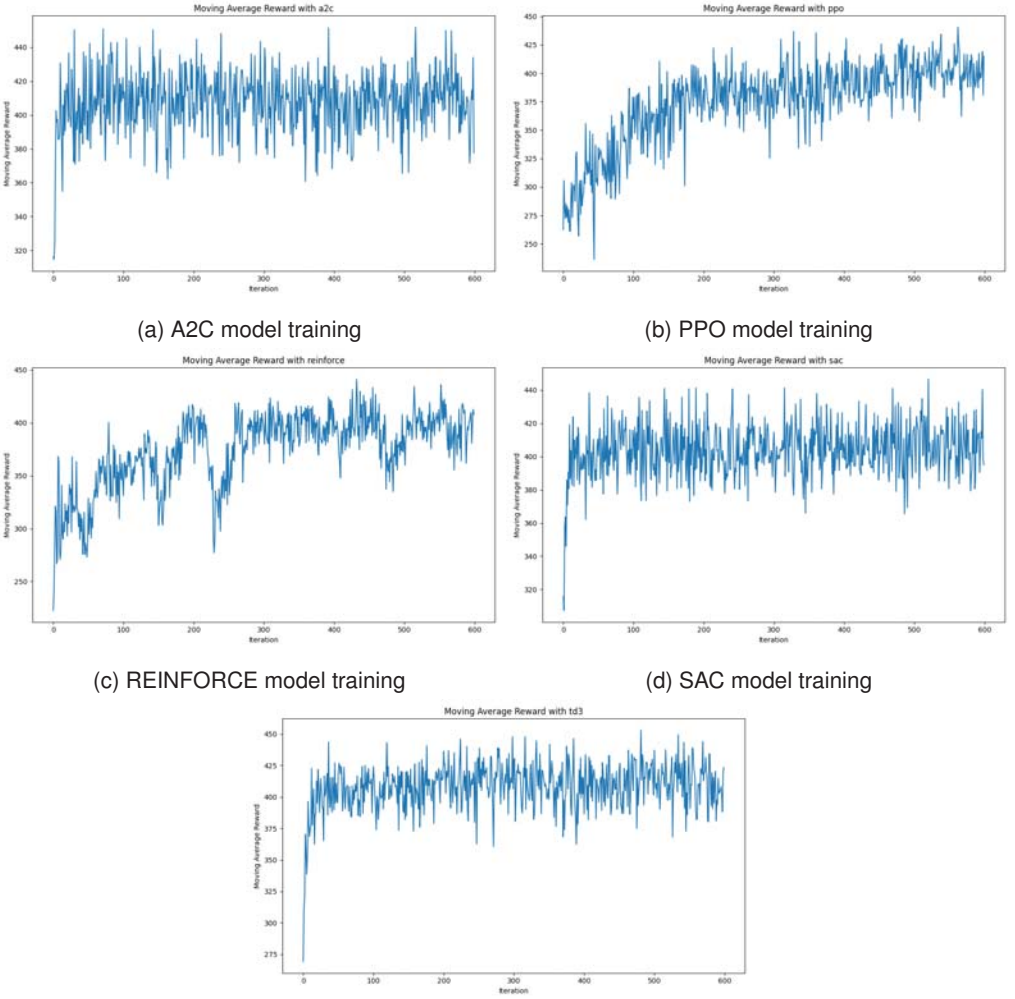
To train and evaluate our DRL models, we employed a high-performance computing setup. The training process was carried out on a GPU for efficient computation. Specifically, we take advantage of an NVIDIA GeForce RTX 4070 GPU, which offers considerable computational power for training neural networks and running simulations.

4.3 Results

In this subsection, we present the results obtained from our experiments, showcasing the performance of various DRL models in dynamic pricing scenarios. We evaluate the models based on multiple metrics, including reward optimization, convergence speed, and training time.

4.3.1 Reward Comparison

Our experiments demonstrate the effectiveness of the DRL models in optimizing reward for airline ticket sales. Figure 3 illustrates the reward trajectories for each model over the course of training iterations. As observed, TD3 exhibits remarkable performance, achieving rapid reward growth. SAC follows closely, displaying a competitive reward increase throughout the training process. A2C and PPO show consistent improvements, albeit at a slightly slower pace. REINFORCE, while effective, exhibits a comparatively slower revenue optimization trend.



(a) A2C model training

(b) PPO model training

(c) REINFORCE model training

(d) SAC model training

(e) TD3 model training

Figure 3: Moving Average Reward of DRL models

4.3.2 Convergence Speed

Analyzing the convergence speed of the models provides crucial insights into their efficiency. Figure 3 illustrates the convergence behavior of each model. Remarkably, TD3 stands out as the fastest converging model, reaching optimal pricing strategies swiftly. SAC and A2C follow suit, showcasing rapid convergence. PPO exhibits commendable convergence speed, albeit slightly slower. REINFORCE, while effective, converges at a comparatively moderate pace.

4.3.3 Training Time

Efficient use of computational resources is capital in practical applications of DRL models. Table 3 summarizes the training times for each DRL model. Notably, TD3 demonstrates noteworthy efficiency, requiring only 5 hours and 36 minutes for training. SAC follows closely, with a training time of 6 hours and 53 minutes. A2C, while highly effective, required 8 hours and 16 minutes for training. PPO and REINFORCE, while delivering competitive results, necessitated 12 hours and 49 minutes and 11 hours and 32 minutes, respectively.

	REINFORCE	**PPO**	**A2C**	**SAC**	**TD3**
Training time	11h32min	12h49min	8h16min	6h53min	5h36min

Table 3: Training time for DRL models.

5 Discussion

In this section, we explore the ramifications and subtleties of our discoveries offering context and valuable insights, into the outcomes.

5.1 Model Performance Discrepancy

The disparities in performance observed among the DRL models emphasize the critical role of algorithm selection in dynamic pricing strategies employing DRL techniques. TD3's swift convergence and effective reward optimization position it as a compelling option for real-time pricing applications, ensuring agile and responsive decision-making. SAC and A2C, exhibiting a harmonious blend of convergence speed and reward optimization, provide a dependable and balanced alternative for dynamic pricing implementations. On the other hand, PPO, although effective, might necessitate additional fine-tuning, particularly in tailored airline pricing contexts, owing to its relatively slower convergence speed. Careful consideration of these factors is pivotal when selecting the most suitable DRL model for specific pricing scenarios, ensuring optimal performance and responsiveness in dynamic pricing strategies.

5.2 Computational Efficiency

The considerable disparity in training durations across the models highlights the crucial aspect of computational efficiency. TD3's ability to yield competitive outcomes within a significantly

reduced timeframe underscores its practical applicability, especially in resource-intensive settings. This rapid training capability positions TD3 as a feasible choice for airlines aiming for agile and adaptive pricing approaches. Additionally, SAC, while requiring a slightly longer training duration, strikes an appealing balance between efficiency and performance, making it a compelling option in the realm of dynamic pricing strategies.

6 Future Work

In future research, there exist several promising avenues that warrant exploration, particularly within the context of airline competitors' pricing environments and the application of Multi-Agent Reinforcement Learning (MARL) approaches. One critical direction involves the refinement of DRL models through the adoption of more intricate architectures and advanced training methodologies. This evolution equips these models to effectively capture nuanced patterns in customer behavior, enhancing their adaptability within the competitive landscape.

Furthermore, integrating real-time data streams, a key consideration in airline pricing, can augment pricing strategies by ensuring responsiveness to the ever-fluctuating market dynamics shaped by both competitors and customer preferences. Investigating how exogenous factors, including economic indicators and global events, influence customer choices is essential for the development of robust pricing strategies in an environment marked by dynamic competitive forces.

Expanding the horizons of DRL techniques, these approaches hold promise not only for airline ticket sales but also for a range of other industries, encompassing hospitality, e-commerce, transportation services, and general product pricing. Assessing the transferability and generalizability of MARL-based pricing models across these diverse domains constitutes a significant step toward advancing the field of dynamic pricing within highly competitive markets. It is also true that these techniques can be adapted in various fields of application such as what was developed for example in (Abourezq and Idrissi, 2014; Abourezq, Idrissi and Yakine, 2016; Abourezq and Idrissi, 2015; Abourezq, Idrissi and Rehioui, 2020; El Handri and Idrissi, 2020b; El Handri and Idrissi, 2020a; El Handri and Idrissi, 2020c; Er-Rafyg, Abourezq, Idrissi and Bouhouch, 2022; Idrissi, Li and Myoupo, 2006; Idrissi, 2012b; Idrissi, 2012a; Idrissi and Fedoua, 2014; Idrissi and Abourezq, 2014; Idrissi, Rehioui, Laghrissi and Retal, 2015; Idrissi and Zegrari, 2015; Idrissi, El Handri, Rehioui and Abourezq, 2016; Essadqi, Idrissi and Amarir, 2018; Laghrissi, Retal and Idrissi, 2016; Retal and Idrissi, 2018; Rehioui, Idrissi, Abourezq and Zegrari, 2016; Rehioui and Idrissi, 2017; Rehioui and Idrissi, 2019; Zankadi, Hilal, Idrissi and Daoudi, 2022; Zankadi, Idrissi, Daoudi and Hilal, 2023; Zegrari, Idrissi and Rehioui, 2016; Zegrari and Idrissi, 2020). We firmly believe that these DRL technologies and their adaptations can revolutionize several socio-economic, environmental and industrial sectors.

7 Conclusion

This study presents a comprehensive analysis of Deep Reinforcement Learning (DRL) models applied to dynamic pricing within the airline industry. Through extensive experimentation, we evaluated and compared five DRL algorithms, namely REINFORCE, PPO, A2C, SAC, and TD3. Our findings highlight the efficiency and adaptability of these models in optimizing revenue while considering diverse customer features. TD3 emerged as a standout performer, demonstrating swift convergence and suggesting its potential for real-world applications in airline ticket pricing. The comparison of training times provides practical insights into the computational resources required for deploying these models effectively. This research not only provides valuable benchmarks for DRL-based pricing strategies but also underscores the significance of machine learning techniques in revolutionizing revenue management practices. As industries continue to evolve, leveraging advanced AI methods like DRL remains essential to staying ahead of the market curve. The insights gained from this study pave the way for future advancements in dynamic pricing methodologies and represent a promising trajectory for the intersection of artificial intelligence and business strategy.

References

Abdella, J. A., Zaki, N., Shuaib, K. and Khan, F. 2021. Airline ticket price and demand prediction: A survey, *Journal of King Saud University - Computer and Information Sciences* **33**(4): 375–391.
URL: *https://doi.org/10.1016/j.jksuci.2019.02.001*

Abourezq, M. and Idrissi, A. 2014. A cloud services research and selection system, *IEEE ICMCS*.

Abourezq, M. and Idrissi, A. 2015. Integration of qos aspects in the cloud service research and selection system, *International Journal of Advanced Computer Science and Applications* **6**(6): 1–13.

Abourezq, M., Idrissi, A. and Rehioui, H. 2020. An amelioration of the skyline algorithm used in the cloud service research and selection system, *International Journal of High Performance Systems Architecture* **9**(2-3): 136–148.

Abourezq, M., Idrissi, A. and Yakine, F. 2016. Routing in wireless ad hoc networks using the skyline operator and an outranking method, *Proceedings of the International Conference on Internet of Things and Cloud Computing - ICC'16*, ACM Press, pp. 1–10.

Asker, J., Fershtman, C. and Pakes, A. 2022. Artificial intelligence, algorithm design, and pricing, *AEA Papers and Proceedings* **112**: 452–456.
URL: *https://doi.org/10.1257/pandp.20221059*

Betancourt, J., Hortaçsu, A., Oery, A. and Williams, K. 2022. Dynamic price competition: Theory and evidence from airline markets, *Technical report*, National Bureau of Economic

Research.
URL: *https://doi.org/10.3386/w30347*

Boutyour, Y. and Idrissi, A. 2023. Deep reinforcement learning in financial markets context: Review and open challenges, *Modern Artificial Intelligence and Data Science: Tools, Techniques and Systems* pp. 49–66.
URL: *https://doi.org/10.1007/978-3-031-33309-5%5F5*

Busoniu, L., Babuska, R., De Schutter, B. and Ernst, D. 2017. *Reinforcement learning and dynamic programming using function approximators*, CRC press.

Cheng, Y., Zou, L., Zhuang, Z., Liu, J., Xu, B. and Zhang, W. 2019. An extensible approach for real-time bidding with model-free reinforcement learning, *Neurocomputing* **360**: 97–106.
URL: *https://doi.org/10.1016/j.neucom.2019.06.009*

Colpas, A.-C., Patricia, P., Carrascal, O.-C., Isabel, A., Aziz, B. S., Melo, P. n.-M. and Alberto, M. 2023. Rtla-har: A model proposal based on reinforcement and transfer learning for the adaptation of learning in human activity recognition., *International Journal of Artificial Intelligence* **21**(1): 154 – 175.
URL: *https://www.scopus.com/inward/record.uri?eid=2-s2.0-85160076616&partnerID=40&md5=3a57ec943be9caacfab8a43733ad94df*

El Handri, K. and Idrissi, A. 2020a. Comparative study of top_k based on fagin's algorithm using correlation metrics in cloud computing qos, *International Journal of Internet Technology and Secured Transactions* **10**(1-2): 143–170.

El Handri, K. and Idrissi, A. 2020b. Parallelization of top-k algorithm through a new hybrid recommendation system for big data in spark cloud computing framework, *IEEE Systems Journal* **15**(4): 4876–4886.

El Handri, K. and Idrissi, A. 2020c. Parallelization of top_k algorithm through a new hybrid recommendation system for big data in spark cloud computing framework, *IEEE Systems Journal* **15**(4): 4876–4886.

Er-Rafyg, A., Abourezq, M., Idrissi, A. and Bouhouch, A. 2022. Courses recommendations using skyline bnl algorithm, *International Journal of Artificial Intelligence™* **20**(1): 68–86.
URL: *http://www.ceser.in/ceserp/index.php/ijai/article/view/6858*

Essadqi, M., Idrissi, A. and Amarir, A. 2018. An effective oriented genetic algorithm for solving redundancy allocation problem in multi-state power systems, *Procedia Computer Science* **127**(C): 170–179.

Gabor, M. R., Kardos, M. and Oltean, F. D. 2022. Yield management—a sustainable tool for airline e-commerce: Dynamic comparative analysis of e-ticket prices for romanian full-service airline vs. low-cost carriers, *Sustainability* **14**(22): 15150.
URL: *https://doi.org/10.3390/su142215150*

Idrissi, A. 2012a. How to minimize the energy consumption in mobile ad-hoc networks, *arXiv preprint arXiv: 1307.5910* **-**(-): 1–10.

Idrissi, A. 2012b. Some methods to treat capacity allocation problems, *Journal of Theoretical and Applied Information Technology* **37**(2): 141–158.

Idrissi, A. and Abourezq, M. 2014. Skyline in cloud computing, *Journal of Theoretical & Applied Information Technology* **60**: 12.

Idrissi, A., El Handri, K., Rehioui, H. and Abourezq, M. 2016. Top-k and skyline for cloud services research and selection system, *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, pp. 1–10.

Idrissi, A. and Fedoua, Y. 2014. Multicast routing with quality of service constraints in the ad-hoc wireless networks, *Journal of Computer Science* **10**(10.3844/jcssp.2014.1839.1849): 1839–1849.

Idrissi, A., Li, C. M. and Myoupo, J. F. 2006. An algorithm for a constraint optimization problem in mobile ad-hoc networks, *18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, IEEE, pp. 555–562.

Idrissi, A., Rehioui, H., Laghrissi, A. and Retal, S. 2015. *An improvement of DENCLUE algorithm for the data clustering*, IEEE.

Idrissi, A. and Zegrari, F. 2015. A new approach for a better load balancing and a better distribution of resources in cloud computing, *International Journal of Advanced Computer Science and Applications* **6**(10): 1–6.

Justin, C. Y., Payan, A. P. and Mavris, D. 2021. Demand modeling and operations optimization for advanced regional air mobility, *AIAA AVIATION 2021 FORUM*, American Institute of Aeronautics and Astronautics.
URL: *https://doi.org/10.2514/6.2021-3179*

Laghrissi, A., Retal, S. and Idrissi, A. 2016. Modeling and optimization of the network functions placement using constraint programming, *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons.

Rehioui, H. and Idrissi, A. 2017. A fast clustering approach for large multidimensional data, *International Journal of Business Intelligence and Data Mining* **15**(3): 349–369.

Rehioui, H. and Idrissi, A. 2019. New clustering algorithms for twitter sentiment analysis, *IEEE Systems Journal* **14**(1): 530–537.

Rehioui, H., Idrissi, A., Abourezq, M. and Zegrari, F. 2016. Denclue-im: A new approach for big data clustering, *Procedia Computer Science* **100**(83): 560–567.

Retal, S. and Idrissi, A. 2018. A multi-objective optimization system for mobile gateways selection in vehicular ad-hoc networks, *Computers & Electrical Engineering* **73**(January 2019): 289–303.

ROOS, N. D., MILLS, G. and WHELAN, S. 2010. Pricing dynamics in the australian airline market, *Economic Record* **86**(275): 545–562.
URL: *https://doi.org/10.1111/j.1475-4932.2010.00653.x*

Sarhadi, A., Akbari, J. and Karimi, A. 2022. Economic based scheduling and load balancing algorithms in cloud computing using learning automata, *International Journal of Artificial Intelligence* **20**(1): 87 – 116.
URL: *https://www.scopus.com/inward/record.uri?eid=2-s2.0-85127744749&partnerID=40&md5=edb168b6f08d1e8f74b7662d3abd4361*

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. 2017. Proximal policy optimization algorithms, *arXiv preprint arXiv:1707.06347* .

Shukla, N., Kolbeinsson, A., Otwell, K., Marla, L. and Yellepeddi, K. 2019. Dynamic pricing for airline ancillaries with customer context, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM.
URL: *https://doi.org/10.1145/3292500.3330746*

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D. 2016. Mastering the game of go with deep neural networks and tree search, *Nature* **529**(7587): 484–489.
URL: *https://doi.org/10.1038/nature16961*

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

Sutton, R. S., McAllester, D., Singh, S. and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation, *Advances in neural information processing systems* **12**.

Talluri, K. T. and Ryzin, G. J. V. 2004. *The Theory and Practice of Revenue Management*, Springer US.
URL: *https://doi.org/10.1007/b139000*

Wang, K., Zhang, A. and Zhang, Y. 2018. Key determinants of airline pricing and air travel demand in china and india: Policy, ownership, and LCC competition, *Transport Policy* **63**: 80–89.
URL: *https://doi.org/10.1016/j.tranpol.2017.12.018*

Wittman, M. D. and Belobaba, P. P. 2018. Dynamic pricing mechanisms for the airline industry: a definitional framework, *Journal of Revenue and Pricing Management* **18**(2): 100–106.
URL: *https://doi.org/10.1057/s41272-018-00162-6*

Zankadi, H., Hilal, I., Idrissi, A. and Daoudi, N. 2022. A social profile ontology to enhance learner experience in moocs, *International Journal of Emerging Technologies in Learning (iJET)* **17**(4): 148–170.

Zankadi, H., Idrissi, A., Daoudi, N. and Hilal, I. 2023. Identifying learners' topical interests from social media content to enrich their course preferences in moocs using topic modeling and nlp techniques, *Education and Information Technologies* **28**(5): 5567–5584.

Zegrari, F. and Idrissi, A. 2020. Modeling of a dynamic and intelligent simulator at the infrastructure level of cloud services, *Journal of Automation Mobile Robotics and Intelligent Systems* **14**(3): 65–70.

Zegrari, F., Idrissi, A. and Rehioui, H. 2016. Resource allocation with efficient load balancing in cloud environment, *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, pp. 1–7.

Zhang, W. 2021. Dynamic airline network pricing, *SSRN Electronic Journal* .
URL: *https://doi.org/10.2139/ssrn.3833013*