TU Delft

Delft University of Technology

Fleet planning under demand and fuel price uncertainty using actor–critic reinforcement learning

Geursen, Isaak L.; Santos, Bruno F.; Yorke-Smith, N.

DOI
10.1016/j.jairtraman.2023.102397

Publication date
2023

Document Version
Final published version

Published in
Journal of Air Transport Management

Citation (APA)
Geursen, I. L., Santos, B. F., & Yorke-Smith, N. (2023). Fleet planning under demand and fuel price uncertainty using actor–critic reinforcement learning. *Journal of Air Transport Management*, *109*, Article 102397. https://doi.org/10.1016/j.jairtraman.2023.102397

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Journal of Air Transport Management 109 (2023) 102397

Contents lists available at ScienceDirect

Journal of Air Transport Management

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

Fleet planning under demand and fuel price uncertainty using actor–critic reinforcement learning

Izaak L. Geursen [a], Bruno F. Santos [b,*], Neil Yorke-Smith [c]

[a] ORTEC B.V., Netherlands, Houtsingel 5, 2719 EA Zoetermeer, The Netherlands
[b] Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands
[c] Algorithmics group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

ARTICLE INFO

Keywords:
Airline fleet planning
Stochastic optimisation
Reinforcement learning
Advantage Actor–Critic
Fuel price uncertainty

ABSTRACT

Current state-of-the-art airline planning models face computational limitations, restricting the operational applicability to problems of representative sizes. This is particularly the case when considering the uncertainty necessarily associated with the long-term plan of an aircraft fleet. Considering the growing interest in the application of machine learning techniques to operations research problems, this article investigates the applicability of these techniques for airline planning. Specifically, an Advantage Actor–Critic (A2C) reinforcement learning algorithm is developed for the airline fleet planning problem. The increased computational efficiency of using an A2C agent allows us to consider real-world-sized problems and account for highly-volatile uncertainty in demand and fuel price. The result is a multi-stage probabilistic fleet plan describing the evolution of the fleet according to a large set of future scenarios. The A2C algorithm is found to outperform a deterministic model and a deep Q-network algorithm. The relative performance of the A2C increases as more complexity is added to the problem. Further, the A2C algorithm can compute a multi-stage fleet planning solution within a few seconds.

1. Introduction

Fleet planning concerns the long-term fleet composition decisions of transportation companies. A company seeks to develop its fleet in order to optimally supply a demand network with the forecast amount of transportation units while respecting operational constraints. The fleet planning process generally consists of two questions: (1) what units are needed, and (2) when to acquire them (Sa et al., 2019).

The deregulation of the airline industry had a major influence on the economic model of airlines. The resulting economic incentive has spiked the interest in optimisation approaches to optimise airline processes. The airline planning process is a sequential one, with each decision imposing constraints on the next. The first step of the airline planning process is fleet planning. This has a major influence on the financial position of an airline, as it defines the bounds for sequential decisions (Belobaba et al., 2009).

In order to optimise this fleet planning process, various modelling techniques have been established, ranging in complexity as well as the computational effort required. Early modelling of airline fleet planning started with the use of deterministic models to optimise long-term fleet planning (Dantzig and Fulkerson, 1954; Wyatt, 1961; Gould, 1969; Shube and Stroup, 1975; Schick and Stroup, 1981). In contrast to the single-stage approach followed in previous works (Shube and Stroup, 1975) were the first study to create a multi-stage fleet plan, capturing the acquisition and retirement of vehicles over a period of 10 years. These deterministic models assume a perfectly predictable future without the use of stochastics. This assumption is an over-simplification of reality, necessarily resulting in a sub-optimal solution and limiting the models' applicability. Despite this, some modern deterministic fleet planning models have been developed, often to test new methods or to find a basis for further research and analysis (e.g., Sayarshad and Ghoseiri, 2009; Bazargan and Hartman, 2012).

The various uncertainties encountered in airline planning have caused the development of stochastic models to solve the fleet planning problem. The importance of incorporating uncertainty in passenger demand into airline planning models has been highlighted in the literature, with several studies proposing stochastic models to address the problem (e.g., Listes and Dekker, 2005; Khoo and Teoh, 2014; Sa et al., 2019). A popular implementation of stochastic models is the two-stage model, with implementations to solve the airline fleet planning in, e.g., Oum et al. (2000), List et al. (2003), Listes and Dekker (2005) and Carreira et al. (2017). Naturally, several authors have extended the two-stage approaches to multi-period models, capturing multiple

* Corresponding author.
E-mail addresses: izaak.geursen@ortec.com (I.L. Geursen), b.f.santos@tudelft.nl (B.F. Santos), n.yorke-smith@tudelft.nl (N. Yorke-Smith).

https://doi.org/10.1016/j.jairtraman.2023.102397
Received 12 July 2022; Received in revised form 7 February 2023; Accepted 5 March 2023
Available online 20 March 2023
0969-6997/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

decision periods in the definition of the airline fleet plan (e.g., Schick and Stroup, 1981; Hsu et al., 2011; Repko and Santos, 2017; Sa et al., 2019).

The complexity of the problem increases when considering the intrinsic uncertainties associated with long-term planning. Several sources of uncertainty need to be considered when modelling uncertainty associated with long-term planning. However, previous studies only addressed uncertainties associated with passenger demand. No other sources of uncertainties were considered. In particular, fuel prices are among the highest costs of an airline, reaching values of up to 60% of an airline's direct operating cost (Clarke and Smith, 2004; Belobaba et al., 2009). They are also critical for assessing the value of renewing a fleet of aircraft. The major influence of fuel price, combined with the extreme volatility in fuel prices (U.S. Energy Information Administration, 2020), compromises the capability to accurately assess the profitability of a fleet plan if fuel price uncertainty is not considered (Naumann and Suhl, 2013).

When attempting to extend the fleet planning problem to consider multiple sources of uncertainty, the challenge is to address the computational limitations associated with the traditional stochastic modelling techniques. Including additional factors of uncertainty into a fleet planning model will, therefore, either reduce the size of the problem to which a solution can be found or limit the operational applicability of this solution due to the sub-optimality of the implemented method.

In this article, we propose an Approximate Dynamic Programming or Reinforcement Learning (RL) approach to address the airline fleet planning problem under multiple sources of uncertainty. Dynamic programming (DP) approaches were already proposed in the past to solve large-scale planning problems (Lam et al., 2007; Cristobal et al., 2009; Novoa and Storer, 2009), including some applications in the airline industry (Hsu et al., 2011; Khoo and Teoh, 2014). DP allows a sequential decision problem to be subdivided into sub-problems which are solved recursively (Bertsekas, 2005; Shapiro et al., 2014). However, solving these large problems backward results in computational difficulties when solving large use cases, as is seen in the literature (Pantuso et al., 2015, 2016). This again requires the models to either be limited in size or complexity to be solvable: the *curse of dimensionality* (Powell, 2011). Reinforcement learning (RL) brings another angle to cope with the computational difficulties associated with large DP problems (Sutton and Barto, 2018). Following this technique, the problem is solved by determining the optimal decision at every time step, and learning the expected value by analysing previous interactions with the environment. RL algorithms are a popular research topic and have been implemented in many application fields (Lam et al., 2007; Novoa and Storer, 2009; Simão et al., 2010; Powell et al., 2014; Mnih et al., 2015; Tong et al., 2020; Chen et al., 2020). Practical applications of RL techniques in aviation problems can be found in Balakrishna et al. (2010), Shihab and Wei (2021) and Andrade et al. (2021). However, it must be noted that cases of RL models in the strategic airline literature are very limited. To the authors' best knowledge, the only exceptions are the academic works from Requeno García (2017) and de Koning (2020). The most recent work, de Koning (2020), described the application of a Deep Q-Network to a fleet planning model under demand uncertainty, where an optimisation algorithm evaluates and learns optimal fleet decisions. The algorithm was applied to an illustrative use case, and it was shown that the method could improve its prediction values over the learning period and produce near-optimal fleet solutions showing comparable results to its deterministic equivalent. However, de Koning (2020) concluded that the calculation of the reward function was very computationally extensive. To calculate the reward, the solution computed in each episode had to be compared to the optimal solution, which was computed using a MILP model, suffering from the same computational issues as other approaches developed to solve the fleet planning problem under uncertainty. This limits the size and operational applicability of the method proposed.

In fact, despite the many application examples in the literature, classical RL techniques, such as Q-learning, commonly suffer from complications when applied to real-world problems because of two reasons: (1) problems possess complex data samples, where even simple implementations might require enormous amounts of data, and (2) tuning parameters such as learning rates and exploration constants are difficult to optimise and unstable (Haarnoja et al., 2018). These factors cause slow convergence and expensive implementation procedures. A possible method to reduce computational issues is using actor–critic (AC) methods (Grondman et al., 2012). AC methods can achieve a more rapid convergence than classical RL techniques, using simultaneous value and policy iteration to converge and train an agent more efficiently. Arguably the most popular AC algorithm is the *Advantage Actor–Critic* method (A2C) (Wang et al., 2017), which we adopt in this article.

For this reason, in this paper, we propose the actor–critic (AC) method to solve our stochastic multi-stage problem. In particular, we propose to use the *Advantage Actor–Critic* algorithm (A2C) (Clemente et al., 2017), adapted from the asynchronous variant of Mnih et al. (2016). This has proven to be a promising algorithm in other fields of research, such as robotics (Grondman et al., 2012). Additionally, the major advantage of this method, compared to classical RL approaches as Deep Q-Networks (DQN) (Arulkumaran et al., 2017), is that an implementation of a less complicated reward function is possible.

In summarising, the contributions of this article to the state-of-the-art are:

1. The first approach includes multiple sources of uncertainty in the airline fleet planning optimisation process, and the first implementation of fuel price as a stochastic variable.
2. A novel application of state-of-the-art reinforcement learning technique to optimise this multi-stage stochastic problem.
3. The first application of an actor–critic method within the airline planning domain providing a practical application example of this algorithmic approach for supporting decision-makers.

The result of the proposed approach is a multi-stage probabilistic fleet plan. That is, it is a plan over time representing the probability of different compositions being the best fleet when considering future demand and fuel price evolution scenarios. We believe that our approach, including two sources of uncertainty for the first time and capable of considering a large set of scenarios when producing the fleet plans, is relevant for airlines to support their strategic fleet decisions. The probabilistic assessment of how the fleet should evolve will help decision-makers to:

- know which action to take now to respond to short-term demand and market opportunities and without compromising future decisions' scope.
- foresee how the fleet plan should evolve in the long term considering possible future scenarios, helping to prepare the fleet evolution. For this, the decision-makers do not need to know exactly what the fleet composition should be in, e.g., ten years but would benefit from knowing the scale of the future investments needed.

The remainder of the article is structured as follows. Section 2 specifies the problem addressed and develops a mathematical model. Section 3 explains the RL approach, and Section 4 the model training. Results from computational experiments are given in Section 5. Section 6 summarises future directions.

2. Problem formulation

The airline fleet planning problem is a multi-period optimisation problem that concerns the fleet sizing decisions over a planning horizon. This is a strategic problem that consists of defining the number of aircraft of different types to acquire or retire at multiple decision

periods. The goal is to maximise the estimated profit, given the estimated demand, fuel prices, fleet composition, and allocation of aircraft to potential routes in the network. In this sequential decision process, the profitability of the fleet in future periods will be influenced by the planning decisions in the initial periods.

In this section, we provide the formulation that we will follow to address this problem using a reinforcement learning algorithm. The formulation is presented as a *Mixed Integer Linear Programming* (MILP) model, considering uncertainty related to demand and fuel prices. It is assumed that these stochastic input factors have different values in each decision period and that they follow a trend observed in historical data. Decisions are made in discrete time steps $t = 0, 1, \dots, T$, with each time step representing one or multiple years between decision periods. Further, it is assumed that the profitability of the fleet per period can be estimated by allocating the fleet to the airline network and using the estimated demand for a standard week of the decision period. No seasonality effects are considered.

The following sets are considered in the formulation of the problem:

$\mathcal{T}$ A set of discrete time steps corresponding to the planning horizon $T = [0, \dots .T]$

$\mathcal{K}$ The set of available aircraft types (size K)

$\mathcal{N}$ The set of airports included in the network (size N)

2.1. State variables

The state variables for this fleet planning problem capture the composition of the fleet and the estimated value of the stochastic variables. The state variable can then be described using the following:

$$s_t = \{ac^{k,t}, q_{ij}^t, \phi^t\} \in S$$

where,

t	The current time step
$ac^{k,t}$	The array of the number of aircraft in the fleet for each type k, for the current time step t
q_{ij}^t	The array of the passenger demand for the route between airports i and j, for the current time step t
ϕ^t	The fuel price, for the current time step t
S	Is the *state space*.

2.2. Decision variables

The decision variables can be divided into two sets, the fleet evolution decisions, representing the decisions regarding the evolution of the fleet, and the fleet allocation decisions per time period t, defining the optimal allocation of the fleet to routes in order to maximise the expected profit. The fleet evolution decisions modify the fleet composition between decision periods either by acquiring new aircraft or disposing units of the fleet. These decision variables can be formulated as follows:

$ac_{acq}^{k,t}$	The amount of acquired aircraft of type k, in time step t
$ac_{dis}^{k,t}$	The amount of disposed aircraft of type k, in time step t

in which,

$$ac_{acq}^{k,t} \in \mathbb{Z}^+, ac_{dis}^{k,t} \in \mathbb{Z}^+.$$

Regarding the fleet allocation decisions, they reflect the frequency defined and the flow of passengers in each route. The frequency is defined by considering the fleet available and the demand estimate for each time period. The flow of passengers is divided into direct and indirect flows, the latter referring to passengers connecting at a hub airport, in the case of a network carrier.

$z_{ij}^{k,t}$	Weekly flight frequency from airport i to airport j operated by aircraft type k, in time step t
x_{ij}^t	Transported passengers from airport i to airport j, in time step t
w_{ij}^t	Transported passengers from airport i to airport j through the hub, in time step t

in which,

$$x_{ij}^t \in \mathbb{R}^+, w_{ij}^t \in \mathbb{R}^+, z_{ij}^{k,t} \in \mathbb{Z}^+.$$

For simplicity, and given the strategic scope of the problem, the two passenger flow decision variables are assumed to be continuous.

We can then define the decision variables of our problem as the tuple of all the previous variables,

$$a_t = \{ac_{acq}^{k,t}, ac_{dis}^{k,t}, x_{ij}^t, w_{ij}^t, z_{ij}^{k,t}\} \in \mathcal{A}$$

where $\mathcal{A}$ is the action space. We let $X^\pi(s_t)$ be the policy that maps a state s_t to an optimal value of these decision variables.

2.3. Exogenous information

The exogenous information is revealed in each decision period t and arises from a stochastic process. This information describes the demand per origin–destination pair considered in the network and the fuel price. We model these stochastic variables using the following:

$\hat{q}_{ij}^t$	Demand between airport i and j in time step t [pax]
$\hat{\phi}^t$	The fuel price in time step t [$/gallon]

We then write our exogenous information W_{t+1} as

$$W_{t+1} = \{\hat{q}_{ij}^{t+1}, \hat{\phi}^{t+1}\}.$$

It is useful to think that this information is revealed to the decision-maker in the time between the decision period t and $t+1$, just before a decision is made. This information can be estimated using stochastic models developed based on historical data.

2.4. Transition function

The transition function captures the evolution of the fleet over the decision periods, according to the fleet evolution decisions in each period. We represent these transitions using the two sets of equations below.

$$ac^{k,t} + ac_{dis}^{k,t} - ac_{acq}^{k,t} = ac^{k,t-1} \qquad \forall t \in (1, \dots, T), k \in \mathcal{K} \tag{1}$$

$$ac^{k,0} + ac_{dis}^{k,0} - ac_{acq}^{k,0} = F^k \qquad \forall k \in \mathcal{K} \tag{2}$$

where F^k is the initial fleet of aircraft of type k. The two functions can be combined, together with the forecasts from the exogenous information, into what we call the *transition function* that we write as

$$S_{t+1} = S^M(s_t, X^\pi(s_t), W_{t+1}).$$

2.5. Objective function

The objective of the fleet planning model is to maximise the profit generated from the allocation to the network of the fleet owned at the multiple time stages of the problem. To formulate the objective function we use the following:

$fare_{ij}$	The average fare per passenger transport in a route between airport i and j [$/passenger]
c^k_{DOC}	Operating cost of an aircraft of type k [$/mile/seat]
c^k_{own}	Cost of owning an aircraft of type k [$]
c^k_{dis}	Cost of disposing of an aircraft of type k [$]
$c^k_{DOC'}$	The fraction of the direct operating costs of the aircraft type k excluding fuel costs [$/mile/seat]
d_{ij}	Distance between airport i and j [miles]
η^k	The fuel efficiency of an aircraft of type k [gallon/mile/seat]
n_{week}	Number of operating weeks in a year
n_{year}	Number of years in an interval, i.e. the interval length
$seat^k$	Number of seats in an aircraft of type k
q^t_{ij}	Demand between airport i and j in period t [pax]
BT^k	Maximum amount of utilisation hours or 'block time' of an aircraft of type k [h/week]
TAT^k	Turnaround time for an aircraft of type k [h]
v^k	The average speed of an aircraft of type k [miles/hour]
R^k	Range of an aircraft of type k [miles]
LF	The Load Factor, the percentage of which the aircraft seats are filled
g_i	Binary variable, $g = 0$ if airport i is the hub airport, else $g = 1$.

The profit of the airline can be divided into three components, being the revenue generated from transporting passengers, the direct operating costs associated with these flights, and the cost associated with having a fleet of aircraft. The revenues are computed by estimating the number of direct and indirect passengers transported between each pair of airports, while the operating costs are computed based on the weekly frequency to be offered in each route of the network. Based on Repko and Santos (2017), we assume that there are no costs associated with acquiring an aircraft. The fleet costs are captured by considering ownership costs and disposal costs. The first represents the costs associated with the depreciation of the aircraft value, eventual aircraft financing costs, insurance, and maintenance costs. The latter disposal costs may refer to the charge that will be applied when a leased aircraft is returned before the end of the leasing contract, or to the value lost in the case the airline sells an aircraft owned by the airline at a lower value than its 'book' value. This way, the total profit over the planning horizon ($t = 0, \ldots, T$) is then given by the following function:

$$\begin{aligned} maxProfit = & \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\left[fare_{ij}\cdot\left(x^t_{ij}+w^t_{ij}\right)\right] \\ & -\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{K}}\left[\left(c^k_{DOC'}+\phi^t\times\eta^k\right)\cdot d_{ij}\cdot seat^k\cdot z^{k,t}_{ij}\right] \\ & -\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}\left[c^k_{own}\cdot ac^{k,t}_{own}+c^k_{dis}\cdot ac^{k,t}_{dis}\right] \end{aligned} \tag{3}$$

where the three terms in the profit are respectively the revenue, the operating costs, and the ownership costs. In order to incorporate the influence of fuel prices in the direct operating costs, we have separated the costs associated with fuel costs from other operating costs. We considered the fuel efficiency of each aircraft type η^k, expressed by the average amount of fuel consumed per each seat-mile travelled, which is assumed to be computed at the maximum occupancy and 'maximum range at maximum payload' of each aircraft type.

2.6. Constraints at time t

The optimal frequencies per route and flows of passengers in the network at each time step t must respect a set of constraints. These constraints should reflect the relation between passengers demand and network flows, the availability of aircraft to offer the capacity needed to transport those passengers, the continuity of the aircraft and passenger flow in the network and the range limitations of each aircraft type. We use the following additional nomenclature to formulate these constraints:

The constraints to this problem can be formulated as follows:

$$x^t_{ij}+w^t_{ij}\le q^t_{ij} \qquad \forall i,j\in\mathcal{N} \tag{4}$$

$$w^t_{ij}\le q^t_{ij}\cdot g_i\cdot g_j \qquad \forall i,j\in\mathcal{N} \tag{5}$$

$$\begin{aligned} & x^t_{ij}+\textstyle\sum_{m\in\mathcal{N}} w^t_{im}\cdot(1-g_j) \\ & +\textstyle\sum_{m\in\mathcal{N}} w^t_{mj}\cdot(1-g_i) \\ & \le\textstyle\sum_{k\in\mathcal{K}} z^{k,t}_{ij}\cdot seat^k\cdot LF \end{aligned} \qquad \forall i,j\in\mathcal{N} \tag{6}$$

$$\sum_{j\in\mathcal{N}} z^{k,t}_{ij}=\sum_{j\in\mathcal{N}} z^{k,t}_{ji} \qquad \forall i\in\mathcal{N},k\in\mathcal{K} \tag{7}$$

$$\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\left(\frac{d_{ij}}{v^k}+TAT^k\right)\cdot z^{t,k}_{ij}\le BT^k\cdot ac^{k,t} \qquad \forall k\in\mathcal{K} \tag{8}$$

$$z^{k,t}_{ij}\le aux^k_{ij} \quad \text{with: } aux^k_{ij}=\begin{cases}10{,}000 & if\ d_{ij}\le R^k\\ 0 & otherwise\end{cases} \qquad \forall i,j\in\mathcal{N},k\in\mathcal{K} \tag{9}$$

where constraints (4) limit the number of direct and indirect passengers transported between any pair of airports to the demand estimated between these two airports. Constraints (5) consider the existence of passengers connecting at a hub airport. In the formulation of our problem, we considered airlines with either one single hub airport or purely point-to-point operations. In the case of a single hub airport, constraints (5) force that there are no indirect passengers between airport i and j, connecting at the hub airport if either of the airports i and j are the hub airport. In the case of a point-to-point airline, these constraints cancel the possibility of using the matrix of indirect passengers between any pair of airports i and j. Constraints (6) guarantee that the transported passengers on any route cannot exceed the number of available seats offered. To compute the number of seats offered in this strategic problem, we considered an average load factor to capture the fact that most aircraft will not fly completely full. Constraints (7) enforce that there is a balance between the number of aircraft arriving at and departing from an airport in the considered time step. Constraints (8) limit the utilisation of the aircraft to a fraction of the time of the fleet of aircraft of each type. That is, it is assumed that aircraft do not fly 24 h per day. Due to the turn-around times at each airport and the moments aircraft spend time on the ground for maintenance, the aircraft availability is limited to an average number of block hours per week. Finally, constraints (9) guarantee that aircraft of a given type can only be allocated to routes within their range.

The network optimisation model resulting from constraints (4)–(9) and objective function (3) simulates an airline network operating a hub-and-spoke network with a single hub. However, it also considers the existence of flights between spokes, i.e., between i and j that are not the hub. Given this, the model can be used to also model purely hub-and-spoke operations by eliminating routes between spokes or point-to-point networks by removing decision variables w^t_{ij}, constraints (5) and adapting constraints (6).

3. Approach

In this section, we elaborate on the reinforcement learning algorithm developed to solve the multi-period fleet planning problem with demand and fuel price uncertainty. We start by introducing the actor–critic method adopted, followed by the methodology considered to model the stochastic process associated with the demand and fuel price estimations. The components of the environment model are discussed in Section 3.3, followed by the elaboration of the reward computation. We conclude this section with an overview of the training setup used to train the A2C agent.

3.1. Actor–critic reinforcement learning

An RL approach supposes the existence of an agent that interacts with an environment. The agent aims to find optimal actions following a *Markov Decision Process* (MDP). When the agent takes actions in the environment, a *reward* is generated. This process is repeated and the agent is trained to find the optimal decisions (Sutton and Barto, 2018).

We recall the standard terminology of RL. At each time step t the agent receives the corresponding *state*, $s_t \in S$, the set of possible states with size S, of the environment. Based on this state the agent executes an *action*, a_t, which is one of the possible actions in the *action space*, $a_t \in \mathcal{A}$. Which action to take is determined by the MDP *policy*, π. The policy is the transfer of the input state to an action, $\pi(a|s)$. After the action the environment transitions to the new state, s_{t+1}, and the agent receives a reward, r_{t+1}. This reward corresponds to the action and modification of the environment resulting in a new state. This loop is an *episode*, e. This process is continued until the terminal state or the amount of episodes are completed. The agent pursues a maximisation of accumulated rewards by discovering the optimal policy. The objective of the agent is to maximise the expected accumulated future reward, G_t. These expected future rewards of an action are taken into consideration when determining the policy.

We design the RL agent to use a function estimation method to learn the optimal policy. Specifically, we use a neural network (NN) (Goodfellow et al., 2016).

When the NN model is trained, the weights of the functions are modified in order to minimise an *error loss*. The setup of a neural network, being the different amount of layers and neurons, will hereafter be referenced to as the model *configuration*.

As the RL training algorithm, we adopt a *actor–critic* method. This method finds the optimal policy by simultaneously executing value and policy iteration. The executed policy is denoted by the *actor*, and the *critic* refers to the approximated value function (Grondman et al., 2012). The actor executes an action, which the critic then evaluates.

As mentioned in the introduction, the *Advantage Actor–Critic (A2C)* algorithm proposed in Mnih et al. (2016) has proven successful in other domains. In this A2C algorithm, the value estimation metric is called the *Advantage*, $A(s_t, a_t)$. This metric describes the expected reward of an action given the state, compared to the prediction of future values when in this state (Sutton and Barto, 2018, Section 13.4). A visualisation of the A2C architecture is given in Fig. 1.

The actor and the critic are implemented as two separate neural networks. The value function used to calculate the advantage and the policy gradient are approximated using two different neural networks and have different function parameters. The parameters of the policy (actor) network are denoted by θ and the parameters of the value function (critic) by θ_v. In order to evaluate the decisions and update the parameters, AC algorithms use *losses*, L, defined separately for the actor and the critic. The *actor loss*, L_π, is defined as the sum of expected advantages. The output of the NN is a probability distribution of the different actions. Multiplication with the corresponding advantages for these actions yields the actor loss. The *critic loss* L_v is defined as the mean squared error between the reward as returned by the environment and the expected reward. The actor loss L_π and the critic loss L_v are combined into the *Total Loss* L. An entropy term $H(\pi)$ is included for the total loss, which encourages the agent to explore different states. The entropy term penalises visiting known states. As the advantage is included in the actor loss, the two neural networks are intertwined and simultaneously trained at each iteration. Pseudocode for the A2C training algorithm is presented in Algorithm 1.

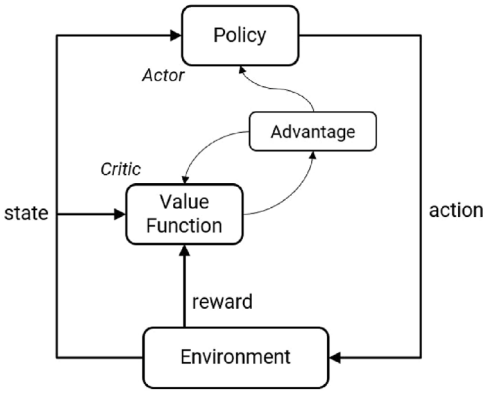


Fig. 1. The A2C Architecture.
Source: Adapted from Sutton and Barto (2018).

Algorithm 1: Advantage Actor–Critic

```
Initialise agent
for each episode do
    start episode
    initialise environment
    get initial state s_t
    while not done do
        perform action a_t according to policy, π_θ(a_t|s_t)
        calculate action–value, Q(s_t, a_t)
        transition to state, s_{t+1}
        receive reward, r_{t+1}
        calculate advantage, A(s_t, a_t)
    end
    calculate actor loss, L_π
    calculate critic loss, L_v
    update actor network with the policy parameters, θ
    update critic network with the value parameters, θ_v
end
```

3.2. Modelling uncertainty

Section 2 identified two key elements on uncertainty in the problem: uncertainty in the demand and in the fuel prices. In this subsection we present the methodology followed to model both sources of uncertainty.

3.2.1. Demand generation model

The airline market is influenced by yearly growth and variations, and in addition, uncertain events have a high impact on demand and other elements of the airline planning process. To capture this stochastic and volatile nature of forecasting uncertainties in airline planning, the *mean-reverting Ornstein–Uhlenbeck (OU)* process was used (Uhlenbeck and Ornstein, 1930). The OU process is based on a Brownian Motion, based on the underlying principle that after a disruptive event, a process will eventually return to its mean function (Ibe, 2013). The process describes the particle following a tendency and a shocking term, and it is characterised by the trend to drift towards the mean

tendency (i.e., mean-reverting). The attraction to the average tendency is as larger as the particle trajectory moves further away from the average trajectory. The magnitude of the shock term is dependent on the volatility of historical data of the process being modelled. OU is a popular forecasting method within the fields of financial forecasting, economics and econometrics because of the stochastic nature of those processes (Barndorff-Nielsen and Shepard, 2001).

For the demand forecasting technique used in modelling the fleet planning process, the method used in Sa et al. (2019) is applied. The authors justified the applicability of OU to forecast air travel demand and its use in airline planning because of the strong correlation between air travel demand and GDP. The mean-reverting process is a stochastic differential equation of the growth of the air-travel demand q^t, which can be discretised to estimate the next air travel demand growth q^{t+1}:

$$q^{t+1} = q^t + \lambda(\mu - q^t) + \sigma \times dW^t \tag{10}$$

The term $\lambda\left(\mu - q^t\right)$ describes the mean reversion process often called the *drift term*. λ is 'the speed of the mean reversion' describing how fast deviation reverts back to the mean. The μ parameter, the 'long term mean growth rate', can be interpreted as the mean air-travel demand growth which the model will approach in the long term. Finally, the process becomes stochastic by including $dW^t = W^{t+1} - W^t$ which is assumed to follow the Normal$(0,1)$ distribution, and is referred to as the 'shock term'. The term σ influences the impact of the disruptions and can be interpreted as the volatility of the change in the growth of demand.

The μ, λ, σ parameters are referred to OU parameters and are deducted from historic data by approximation of the linear relationship between growth in demand x_t and the change of the growth in demand y_t with linear regression fitting. The linear regression of the historic data x_t, y_t reveals the regression coefficients for the slope b and the intercept a of the fitted data to calculate λ, μ, and σ (Chaiyapo and Phewchean, 2017).

In the research of Sa et al. (2019), the air travel demand trajectories for multiple routes are sampled independently from each other. Those authors used historical data from each route in the network to compute the OU parameters for each individual route. However, we consider that this is an unrealistic representation of reality. The demand growth is usually shared by most or several routes, since this growth is commonly influenced by economic growth, fuel prices, aviation market trends, or social factors. Therefore, in this article, we propose an adapted OU demand forecast model in which we consider an inter-dependency between the demand of each single route, referred to *ij*, and the demand of all routes together or a set of routes that compose a market, referred to *m*. More specifically, we first compute the network parameters, λ_m, μ_m, and σ_m, using the Least Squares Regression for the cumulative demand from all routes, and we computed the OU parameter per route, λ_{ij}, μ_{ij}, and σ_{ij}, using the same method but considering the historic demand per each route. To sample the demand, firstly, the market growth, δ_m^{t+1} is sampled according to:

$$\delta_m^{t+1} = \delta_m^t + \lambda_m(\mu'_m - \delta_m^t) + \sigma_m \times dW_n^t \tag{11}$$

with μ'_m being the mean market growth, sampled according to $\mu'_m \sim \mathcal{N}(\mu_m, \sigma_m^2)$. The market growth is used to compute the route growth, δ_{ij}^{t+1}, which is assumed to be an average of both the market and the individual route growth:

$$\delta_{ij}^{t+1} = \frac{1}{2}(\delta_m^{t+1} + \delta_{ij}^t + \lambda_{ij}\left(\mu'_{ij} - \delta_{ij}^t\right) + \sigma_{ij} dW_{ij}^t) \tag{12}$$

The mean growth of a specific route is again sampled according to $\mu'_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$ defined for each different routes. For both the market and the route growth sampling, the Wiener process included with the shock term is sampled according to $dW_{ij}^t \sim \mathcal{N}(0,1)$. With the growth computed for every market, for each time step, the estimated demand for the next period, q_{ij}^{t+1} is:

$$q_{ij}^{t+1} = q_{ij}^t \times (1 + \delta_{ij}^{t+1}) \tag{13}$$

Fig. 2 presents an example for the results from 50 different samples computed with this approach for one of the routes of our case study, the SFO–ORD (San Francisco–Chicago).

3.2.2. Fuel price generation model

Unforeseen events have a significant influence on fuel prices. This extreme volatility is expected to be difficult to capture accurately in a forecasting technique. However, OU has previously been used to model both gas prices (Frikha and Lemaire, 2018) and crude oil spot prices (Ogbogbo, 2018). Aucott and Hall (2014) conclude, based on U.S. data between 1950 and 2013, that fuel prices and GDP are interdependent as well, similar to the relation explored (Sa et al., 2019) to estimate the demand forecast. For these reasons, we also used the OU process to forecast fuel prices.

The OU parameters, λ_f, μ_f, and σ_f, now with a subscript f to represent fuel, can be computed using the Least Squares Regression and historical fuel price values. The input for the OU process is the average yearly values of the fuel prices, which in our study is assumed to be constant over the year. For the fuel price forecast, the price ϕ^t is generated directly by the OU process following:

$$\begin{aligned}\phi^{t+1} &= \phi^t + \lambda_f(\mu_f - \phi^t) + \sigma_f dW_t \\ &= \phi^t + \lambda_f(\mu_f - \phi^t) + \sigma_f \cdot N(0,1)\end{aligned} \tag{14}$$

An example of 100 fuel prices generated with this approach is depicted in Fig. 3. The high volatility encountered in historic fuel prices influences the shock term σ_f, resulting in a wide range of fuel forecasts.

3.3. Environment model

The environment model is captured using the state variables, the decision variables, the exogenous information and transition function, as described in Section 2. In this section, we further elaborate on the formulation of the state space and action space. These are defined in a way that it became efficient for the agent to learn an optimal policy to solve the multi-stage fleet planning problem.

3.3.1. State space

As introduced in Section 2.1, the state variables can be capture using the tuple:

$$s_t = \{ac^t, q_{ij}^t, \phi^t\} \in S$$

However, for the definition of the state space for the A2C algorithm, we decided to consider the demand variation, with respect to the demand in the previous time step, $q^t - q^{t-1}$, instead of the demand at the beginning of the interval. This allows the A2C agent to optimally utilise the difference in demand when making fleet decisions, increasing the learning capabilities of the agent. Consequently, the resulting state vector is given by Eq. (15).

$$s_t = \begin{bmatrix} ac^t \\ q_{ij}^t - q_{ij}^{t-1} \\ \phi^t \end{bmatrix} \tag{15}$$

The size of the state space is defined by the number of aircraft types used as well as the number of airports in the network. For each city pair, one market is considered for the demand. The size of the state space can be calculated following Eq. (16).

$$\begin{aligned} &|S| = n_{ac} + n_{markets} + 1 \\ \text{with:}\quad &n_{markets} = \frac{1}{2} \cdot n_{airports} \cdot (n_{airports} - 1) \end{aligned} \tag{16}$$

It can be inferred that with an increase in the number of airports used in the problem, the state space increases exponentially. Extending the problem in size is therefore expected to have a significant influence on problem metrics such as the training time. Nevertheless, it is assumed that this computation complexity will not grow as much as it would for an equivalent multi-period linear programming approach. This hypothesis is examined in Section 5.

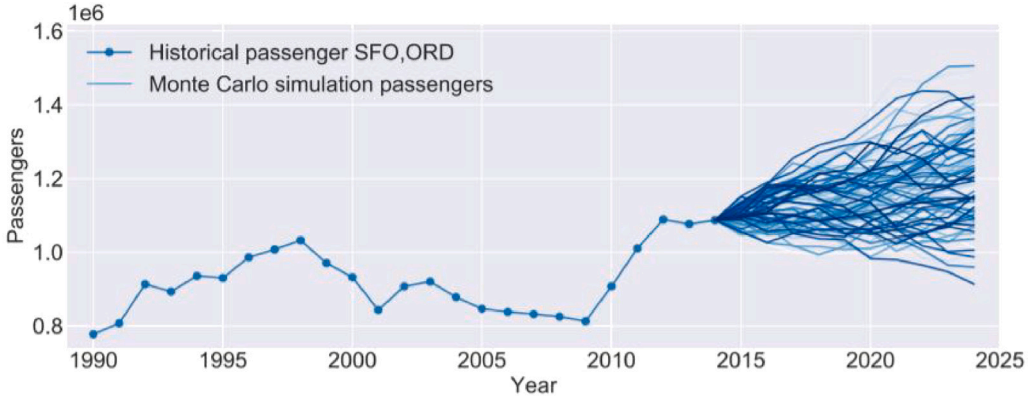


Fig. 2. 50 samples of the demand on route SFO–ORD, using both the network and market parameters.

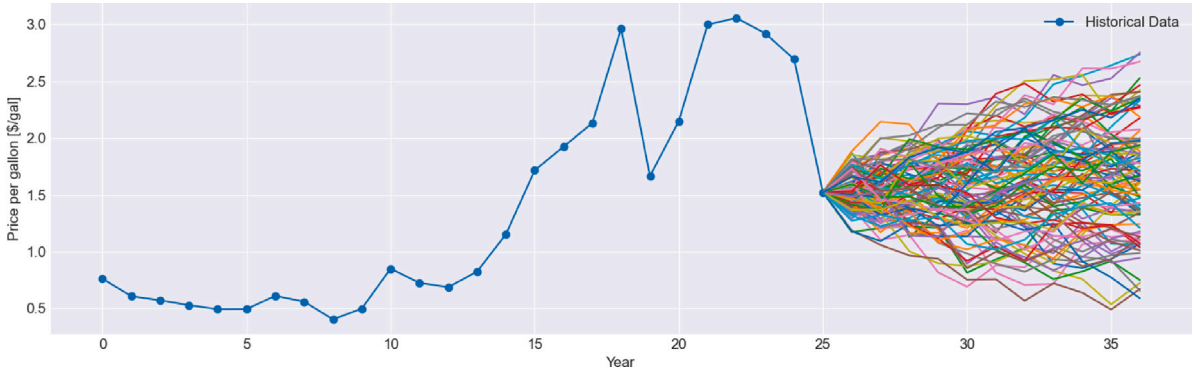


Fig. 3. An example of 100 different fuel price forecasts.

3.3.2. Action space

The action space for the A2C algorithm is defined using the decision variables introduced in Section 2.2. As explained there, the variables can be divided into fleet evolution and fleet allocation decisions. In the formulation of the environment, it is assumed that the agent controls the fleet evolution variables and that the fleet allocation decisions are part of the reward calculation. That is, the weekly flight frequency and the flow of direct and indirect passengers are estimated as a result of the actions decided by the agent with regard to the acquisition and disposal of aircraft. Therefore, only the fleet evolution decision variables were considered in the action space for the agent.

Assuming that an airline can buy and sell any number of aircraft at any decision moment of our problem raises dimensionality issues in the formulation of the action space of our problem. For this reason, we considered that the airline will only execute fleet modifications of one aircraft type, and will not mix acquisitions and disposals in each decision stage. Further, it is also assumed that the airline will only consider a maximum number of aircraft to acquire or dispose at any stage of the problem (i.e., f_{max}). This number can be defined as a function of the problem size. For a problem with a smaller network, the range in optimal fleet planning decisions will be of a smaller magnitude than for a larger network.

Following these assumptions, the action space of our problem can be reformulated as follows.

$$\mathcal{A} = \left[\left[F^k\right]_{k\in\mathcal{K}}, 0 \right] \quad \text{with} \quad F^k = [-f_{max}, \dots, f_{max}] \quad 0 \notin F^k \tag{17}$$

where F^k denotes the collection of possible modifications for an aircraft type $k \in \mathcal{K}$, which is bounded by the maximum modifications allowed, f_{max}; $\mathcal{K}$ is the collection of available aircraft types with size K. The corresponding size is given in Eq. (18).

$$|\mathcal{A}| = (f_{max} \cdot 2) \cdot K + 1 \tag{18}$$

An example of the action space, for a simulation with $K = 2$ aircraft types and a maximum number of modifications of $f_{max} = 3$ is given in Eq. (19). It can be observed that the agent can make modifications to one type in the fleet, or do nothing.

$$\mathcal{A} = \left[\underbrace{-3,-2,-1,+1,+2,+3}_{\text{Type 1}}, \underbrace{-3,-2,-1,+1,+2,+3}_{\text{Type 2}}, 0 \right], \quad \text{with } A = 13 \tag{19}$$

3.4. Reward calculation

Along with the state and action spaces, an appropriate design of the reward function is essential to achieve proper learning and convergence properties (Sutton and Barto, 2018). The reward function is responsible for distinguishing between a good and bad action, giving a reward to the A2C agent for taking good actions and punishing the agent when taking poor actions. The agent's goal is to maximise the total reward it receives. The *reward*, r_{t+1}, is computed using the reward function $R(a_t, s_t, s_{t+1}) \in \mathcal{R}$, with $\mathcal{R}$ being the set of possible rewards.

For our multi-stage fleet planning problem, the maximum reward the agent can receive during an episode should be given by selecting the actions that maximise the objective function described in Eq. (20), respecting constraints (4)–(9). For each time step t, this could be approximated by computing an optimal solution for the equivalent single-stage fleet planning problem. In this case, the contribution value resulting from an action a_t at time step t could be computed as follows:

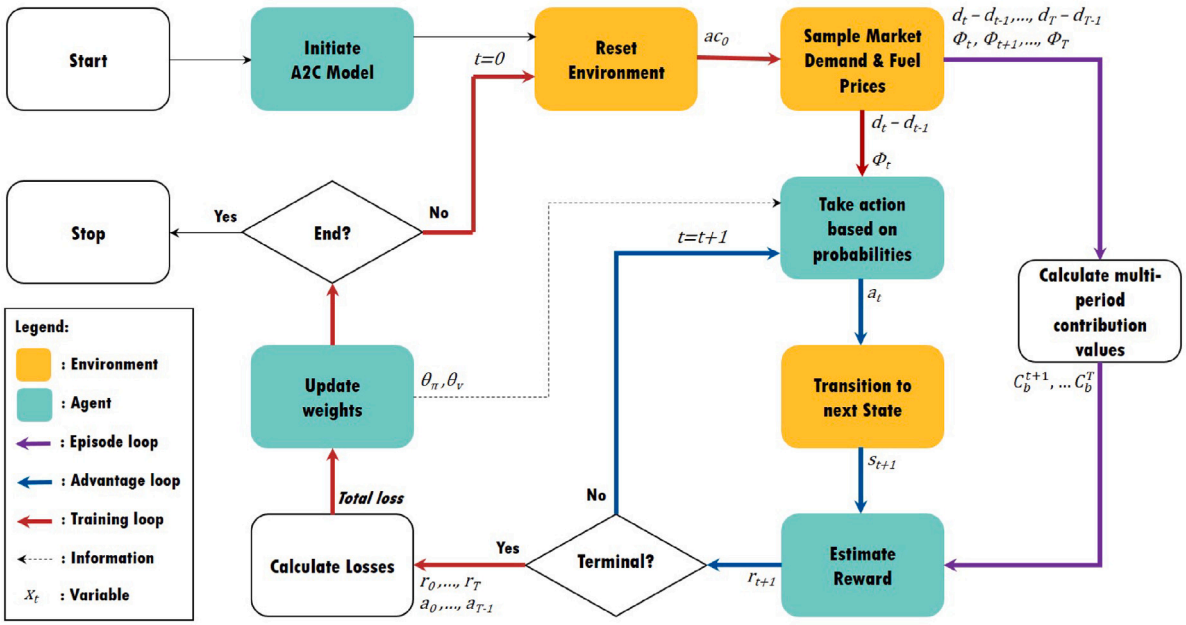


Fig. 4. Visualisation of the interaction between the A2C agent and the environment during training. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$C^{t+1}(s_{t+1} \mid a_t, s_t) = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \left[fare_{ij} \cdot \left(x_{ij}^t + w_{ij}^t\right)\right] - \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{k\in\mathcal{K}} \left[\left(C_{DOC'}^k + \phi^t \times \eta^k\right) \cdot d_{ij} \cdot s^k \cdot z_{ij}^{k,t}\right] - \sum_{k\in\mathcal{K}} \left[C_{own}^k \cdot ac_{own}^{k,t} + C_{dis}^k \cdot ac_{dis}^{k,t}\right] \tag{20}$$

However, using this contribution value as a reward to train the reinforcement learning agent will result in contribution values that depend on the realisation of uncertainty. In contrast, high demand and low fuel prices allow for more profit to be generated regardless of the actions taken. To solve this problem, we propose a reward function that scales the contribution value with the value associated with a 'doing nothing' action. That is, we cancel the exogeneity effects of uncertainty realisation by considering a *baseline solution* as a scaling element. The idea is that the A2C agent should be rewarded when it performs better than the baseline solution and it should be penalised when its action results in a lower contribution than 'doing nothing'. The proposed reward function is given by Eq. (21), in which C_0^{t+1} is the contribution value of the baseline solution computed with the single-stage version of the fleet planning model and C_b^{t+1} is the contribution value of the action taken by the agent.

$$r_{t+1} = \frac{C_b^{t+1} - C_0^{t+1}}{C_0^{t+1}} \tag{21}$$

With this expression, the agent's actions directly influence the reward, which is fed back to the agent to evaluate the action and learn the best policy. The challenge is to compute the contribution value of the actor's action. Executing the single-stage fleet planning problem can be computationally expensive. Given that a reward has to be computed for thousands of training episodes, having the necessity to run it at every episode has therefore a significant influence on the convergence time.

The A2C algorithm addresses this challenge. In fact, what we propose with this algorithm is to replace the computation of the optimisation problem with an estimation of the contribution value by using a NN model. This can be trained to approximate the computation of the reward in a matter of seconds. The overall approach can be summarised in Fig. 4. There we can see three loops. The first loop, represented by the blue arrows, is the computation of a single episode, over the discrete-time steps of the planning horizon. In this cycle, the actor defines the actions for multiple periods, according to the approximation of the rewards for each action and the best policy. The second loop, represented in purple, is used to represent the critic and estimates the reward, or the so-called advantage. This loop is only run for part of the episodes from the training set, enough to train the critic NN model. We use the multi-stage fleet planning model to compute the contribution value for the optimal decisions at each time stage. The third loop, represented by the red arrows, is the training cycle. This cycle is used to calculate the losses associated with both the decisions taken in the first loop (*actor loss*) and the approximations made in the second loop (*critic loss*).

4. Case study and model training

Having defined the A2C algorithm, this section reports how the agent was trained to solve the multi-stage fleet planning problem. We use a proof-of-concept problem representing the fleet planning process of an airline operating on an airport network in the United States (U.S.). The case study used to represent the problem is introduced in the following sub-section, followed by the discussion of the model configuration and the analysis of the model convergence behaviour.

4.1. Case study setup

The case study used to train and later assess the A2C agent comprises ten major U.S. airports in a network, including all 90 possible routes between these nodes. This can be considered a small-size network, equivalent in size to the one operated by Hawaiian Airlines (US) and Lineas Aereas Azteca (Mexico). Two aircraft types are considered in the case study, a medium-sized aircraft (AC Type 1) and a large-sized aircraft (AC Type 2). The aircraft parameters needed for the simulation were obtained from two narrow-bodies aircraft commonly used by airlines. The parameters of the aircraft types are presented in Table 1. We considered an initial fleet of 20 units of AC Type 1 and 10 of AC Type 2. The average fares per route and the historical

Table 1
Parameters of the aircraft types used in the simulation.

Type	s^k [-]	v^k [km/hour]	R^k [miles]	BT^k [h/week]	TAT^k [h]	C^k_{DOC} [$/mile/seat]	$C^k_{DOC'}$ [$/mile/seat]	η^k [gallon/mile]	C^k_{own} [$/year]	C^k_{dis} [$]
AC Type 1	162	543	3582	77	1	0.13	0.65	0.03932	3.05e6	3.05e6
AC Type 2	243	530	3377	80	1.5	0.135	0.0675	0.03878	2.4e6	2.4e6

Table 2
Environment parameters for the initial problem.

Parameter	Abbreviation	Value
Number of Airports	N	10
Number of Routes	$(i,j) \in \mathcal{N}$	90
Aircraft Types	$\mathcal{K}$	AC Type 1, AC Type 2
Maximum Load Factor	LF	0.85
Planning Period [years]	–	10
Interval Length [years]	–	2
Episode Length (time steps)	–	5
Start Year	–	2015
Maximum Fleet Choice	f_{max}	5
Size of the Action Space	$\|A\|$	21
Size of the State Space	$\|S\|$	47

demand data are extracted from the Bureau of Transportation Statistics public data set. In particular, the T-100 Domestic Market (U.S. Carriers) database is used to obtain the historical monthly market data of all U.S. airlines. This dataset contains information on all domestic flights starting from 1990. The information includes specifics about the origin/destination, the operating airline, passenger numbers and distance of a flight. The total amount of passengers between the different origin–destination pairs is used to generate demand realisations for the future. In addition, the fuel price forecast is done using data from the U.S. Energy Information Administration (2020), containing kerosene prices from 1990 onward. As explained previously, the Ornstein–Uhlenbeck process is applied to the historical data to generate the demand/fuel price scenarios for both the training and the testing of the model. This is done to expose the model to the different realisations of uncertainties and evaluate the performance for different situations. The ten airports used in the case study are a selection of some of the busiest airports in the U.S., being: *San Francisco, Atalanta, Chicago-O'Hare, Las Vegas-McCarran, Phoenix, JFK, Boston-Logan, Dallas/Fort Worth, Seattle*, and *Denver*. For the extended case study discussed later, five more airports are added, being: *Los Angeles, Baltimore–Washington, Orlando, Minneapolis–Saint Paul*, and *Miami*.

A planning horizon of 10 years was assumed with a fleet decision every 2 year, resulting in 5 time periods for the A2C loop. To consider the fact that aircraft will not always fly full, it is assumed that all flights have a maximum load factor of $LF = 85\%$. Table 2 summarises the parameters used in the case study.

4.2. Model configuration and tuning

Model configuration refers to the setup of the NN used to represent the actor and the critic in our algorithm: the NN architecture in terms of layers and amount of neurons, as well as the specific hyper-parameters that realise an optimal convergence and testing performance. When tuning a RL agent, the goal is to achieve a rapid and good learning performance, while mitigating the risk of over-fitting. During this research, limited literature was found that described how to do so optimally. Therefore we devised a strategy to tune the agent, including a grid-search approach for the parameterisation of the agent model and an empirical analysis for the agent configuration and training process.

The best configuration of the neural networks was obtained following a grid-search approach. For different parameters combinations, the agent was trained and tested for different sets of episodes to avoid over-fitting. The resulting hyper-parameters are shown in Table 3. The amount of hidden layers n_h is 2, with 100 neurons for each layer. These

Table 3
Parameters used in the 'optimal' configuration.

Parameter	Abbreviation	Value
Learning rate	α	0.0001
Discount factor	γ	0.95
Number of hidden layers	n_h	2
Number of neurons	n_n	100
Number of training episodes	E	15 000

hidden layers are combined with the input of state s_t, and when passed through the different neural networks, result in the policy $\pi_\theta(a|s)$ for the actor and value estimation $V(s)$ for the critic.

4.3. Model convergence

The A2C agent must be evaluated separately from the training, because of the stochastic exploration. The training progress of the agent can be observed by observing the *Exponential Moving Average (EMA)* of the rewards r_t during training. This gives an indication of whether the agent is converging towards a desirable reward, and at which point the agent is showing conversion. Closely monitoring the evolution of the EMA over time is also important for evaluating whether the model is over-fitting and whether to adjust the number of training episodes. The formulation of the EMA calculation is given in Eq. (22).

$$EMA_t = (Y_t - EMA_{t-1}) \cdot k + EMA_{t-1} \tag{22}$$

with $Y_t = r_t$

$$k = \frac{2}{1+n}$$

$n = 50$

The corresponding EMA evolution for the original problem over time during training is plotted in Fig. 5. Because of the stochastic exploration used by the A2C algorithm, the values of the EMA are quite irregular. To be able to observe the trend a smoothed line is plotted. A smoothing plot of the EMA shows an increase of the EMA over time, indicating proper learning. The number of episodes, E, used for training in order to achieve optimal convergence and testing properties, was found to be 15 000.

5. A2C algorithm performance analysis

The performance of the A2C algorithm was analysed by considering a set of scenarios, with different sizes and complexity, and by benchmarking it with two other modelling approaches. In the section, we start by discussing the setup of the scenario, followed by the benchmark techniques with which we compared the A2C algorithm. In Section 5.3 we present the evaluation metrics used to compare the multiple techniques. The performance of the techniques is assessed in Section 5.4, followed by a sensitivity of the results to analyse the correlation between some problem variables and the performance of the techniques.

5.1. Scenarios

The performance of the A2C algorithm was tested in four different scenarios (Table 4). The scenarios are the multiple combinations of two problem sizes and the inclusion or not of the Fuel Price Uncertainty

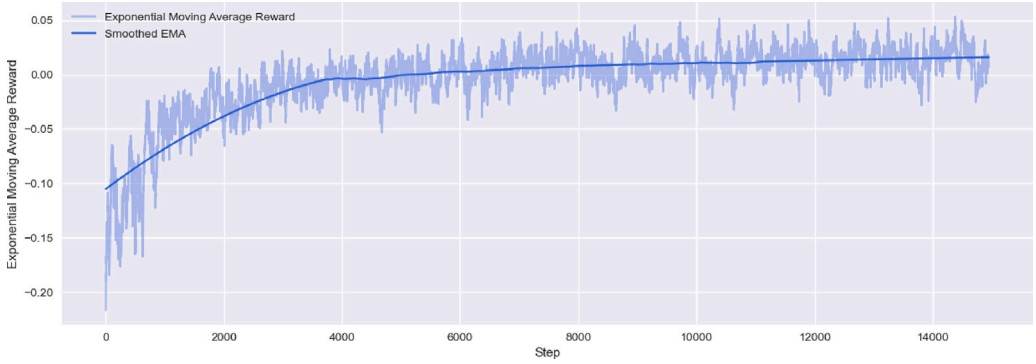


Fig. 5. EMA of the rewards during training for the original problem.

Table 4
Different scenarios used in the model validation.

Scenario	1	2	3	4
N	10	15	10	15
FPU	No	No	Yes	Yes

(FPU). The original problem size consists of 10 airports, and to test the impact of increasing the problem size, five additional airports are added in two of these scenarios, resulting in a total network of 210 possible routes. This is equivalent to a medium-sized network, with an equivalent number of routes to the ones operated by, e.g., Thai AirAsia (Thailand) and Jetstar Airways (Australia). We also increased the initial fleet size — for this medium-size network, we considered the initial fleet to have 40 units of AC Type 1 and 30 of AC Type 2.. To test the impact of adding another source of uncertainty, both network sizes are also evaluated with and without fuel price uncertainty.

The case study described in Section 4.1 was used to create the environment for these scenarios.

5.2. Techniques to compare

The performance of the proposed A2C algorithm was compared with two reference modelling techniques: namely, a deterministic multi-stage fleet plan model and an alternative state-of-the-art RL algorithm.

The first of these was the **Deterministic Dynamic (DD)** model, in which we considered the single-stage fleet planning model formulated by the objective function (20), constraints (4)–(9), and by the following additional set of constraints:

$$\sum_{k\in\mathcal{K}}\left(\min\left(ac_{acq}^{k,t},1\right)+\min\left(ac_{dis}^{k,t},1\right)\right)\leq 1 \qquad \forall t\in\mathcal{T} \tag{23}$$

$$\sum_{k\in\mathcal{K}}\left(ac_{acq}^{k,t}+ac_{dis}^{k,t}\right)\leq f_{max} \qquad \forall t\in\mathcal{T} \tag{24}$$

$$ac^{k,0}+ac_{dis}^{k,0}-ac_{acq}^{k,0}=F^{k} \qquad \forall k\in\mathcal{K} \tag{25}$$

$$ac^{k,t}+ac_{dis}^{k,t}-ac_{acq}^{k,t}=ac^{k,t-1} \qquad \forall t\in(1,\ldots,T),k\in\mathcal{K} \tag{26}$$

Constraints (23) and (24) limit the fleet decision to the action space defined for the A2C algorithm. The first set of constraints guarantees that only the fleet of one type can be changed per time period, while the second set limits the amount of aircraft that can be added to or removed from the fleet in a single time period. The last two sets of constraints concern the fleet continuity over the planning horizon, given the decisions made in a specific time period. Constraints (25) set the initial time period, in which F^k is the initial fleet for aircraft of type k, while constraints (26) repeat the transition for the subsequent time periods. This model was solved at every time step for the remaining time steps within the time horizon (i.e., $[t;t+1;\ldots;T]$) and a modification of the fleet was computed accordingly. Future demand and fuel uncertainty were estimated based on a simple linear regression forecast of the expected future values. At every time step t the stochastic inputs for that time step were revealed, and a simple linear regression forecast is used to estimate future values. The optimal solution is computed using the commercial solver *Gurobi*. This model allowed us to assess the impact of not considering uncertainty in the planning process and compare the computation effort associated with both methods.

For the second comparison model, we used the **Deep Q-Network (DQN)** algorithm proposed in de Koning (2020) to solve this multi-stage fleet planning problem under demand uncertainty. This is a recurrent RL technique used to solve large control and optimisation problems. Comparing our A2C algorithm with the DQN algorithm gave us an idea of the value of using the actor–critic concept, both in terms of solution quality and computational times.

Furthermore, all these modelling techniques, the A2C algorithm, the DD model, and the DQN algorithm, were also compared with the optimal solution of the problem computed executing the multi-stage fleet planning model together with constraints (23)–(26). The solution was computed in a post-processing step, considering the realisations of the uncertainties in the multiple periods. This optimal solution was denoted as C_{FPM}.

5.3. Evaluation metrics

To compare all these modelling techniques, in terms of resulting profit (i.e., *effectiveness*) and computational time (i.e., *efficiency*), we computed four metrics. Two metrics were related to the performance and two other to assess the efficiency of the techniques.

The first effectiveness metric was the ***Testing Score***, TS. The Testing Score assesses how close the solution from a modelling technique was from the profit generated by the optimal solution. Being a comparison with the optimal solution, this method allowed for an equal comparison of the three techniques. Given that the profit by the entire fleet can be a large value and the difference between the profit obtained by the different techniques is relatively close to the optimal solution value, we decided to escalate this score by only considering the interval between the optimal solution value at a given time step (C_{FPM}^t) and a percentage of this optimal solution value, computed by considering a *lower bound* coefficient, lb (Fig. 6). That is, if a technique X generates a profit value lower than $lb * C_{FPM}^t$ the resulting Testing Score TS_X^t is assumed to be 0. A maximum score of 1 was given in the case the technique solution profit matches the optimal solution value and value between 0 and 1 if the profit was between $lb * C_{FPM}^t$ and C_{FPM}^t (red diagonal line). The TS for a technique X was computed according to expression (27).

$$TS_X^t=\begin{cases}\dfrac{C_X-lb\cdot C_{FPM}^t}{(1-lb)\cdot C_{FPM}^t} & if\ C_X>lb\cdot C_{FPM}^t\\ 0 & otherwise\end{cases} \tag{27}$$

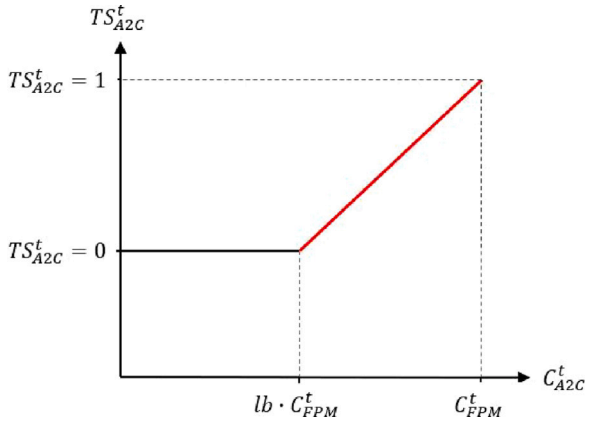


Fig. 6. Visualisation of the Testing Score applied to the A2C algorithm.

The second effectiveness metric was the ***Relative Score***, RS_X, which compares both RL algorithms with the DD model. The mathematical expression for this score is:

$$RS_X = \frac{TS_X}{TS_{DD}} \tag{28}$$

Finally, we used the ***conversion time***, Δt, and the ***computational time***, δt, to assess the algorithms efficiency. The conversion time denotes the total time needed to complete running all ***training episodes***, E. This metric was only relevant for the RL algorithms. It refers to the time needed to properly train the agents. The second metric was used to compare the time it took the three techniques to compute the fleet solution for a single episode.

5.4. Performance analysis

The analysis of the performance of the three modelling techniques was divided into two parts. In the first sub-section, we compared the three techniques for Scenario 1 (Table 4), given that the DQN algorithm proposed by de Koning (2020) has a long training time and was not adapted to consider other sources of uncertainty besides demand uncertainty. In the second sub-section, we compared the performance of the DD model and of the A2C algorithm for the four scenarios.

All metrics presented in this analysis were computed based on a validation batch with a sample size of $n = 1000$ episodes.

5.4.1. Scenario 1

An overview of the different Testing Scores, their corresponding variance and Relative Scores for the different techniques is given in Table 5. The results indicate that the A2C algorithm outperforms the DD model and the DQN algorithm, both in terms of effectiveness and efficiency. Although the A2C only provided a solution with a profit value 0.4 per cent better than the solution from the DD model, the DQN algorithm could not do better than the DD. This is in line with the results reported by de Koning (2020). In terms of computational time, δt, the A2C takes, on average, 3.28 s to compute the solution for a single episode. This is 21 and 24 per cent of the time taken by the DD model and by the DQN algorithm, respectively. It is important to recall that the DQN presented de Koning (2020) has the necessity of continuously calculating an equivalent DD solution in order to compute the reward. This caused the agent to have an inefficient computational time, limiting the model's applicability. A major advantage of the proposed A2C algorithm is the efficient reward function, removing the necessity of calculating the FPM at every time step. The agent requires less computational effort to complete an episode but needs more training episodes (E). In total, the A2C agent needed 15 000 training episodes to converge, against 5 000 for the DQN agent. However, because of the reduced time needed to complete a training episode, the total training time was significantly less. Completing all training times took 871 min for the A2C agent, and 1130 min for the DQN agent.

Table 5
Techniques performance and efficiency for Scenario 1.

	A2C	DQN	DD
TS_X	85.90	83.88	85.56
$Var_{TS,X}$	1.950	0.691	1.710
RS_X [%]	**100.40**	98.04	–
Δt_X [min]	870.54	1130.32	–
E	15 000	5 000	–
δt_X [s]	**3.28**	13.56	15.29

5.4.2. Multiple scenarios

After concluding that the A2C agent is learning satisfactorily and outperforms the DQN agent, the analysis is expanded in this subsection to consider the four scenarios introduced in Section 5.1. We restrict our attention now to the DD model and the A2C algorithm.

The metrics from both techniques are given in Table 6. The A2C algorithm was able to outperform the DD method in the other three scenarios too, both in terms of effectiveness and efficiency. The Related Score increased when the problem size increased and when the fuel price uncertainty (FPU) was considered. In fact, the A2C algorithm is 32.5% better than the DD model in the most complex case, with 15 airports and FPU included. For the two scenarios, including FPU, the effectiveness of both the DD and the A2C techniques deteriorates when compared with the optimal profit value, with Testing Scores of 50.4 and 60.1, respectively. However, the A2C seemed to cope better with the increase of complexity. Further, the A2C obtained better Testing Scores for the scenarios with FPU compared with the homologous scenarios without FPU, suggesting the technique's suitability to deal with multiple sources of uncertainty. When considering the 95% intervals in Table 6, it can be concluded with a statistical significance that the A2C consistently outperforms the DD model for scenarios 2 and 4.

As expected, the average computation time per episode, δt, increased with the increase of the size of the problem and the additional source of uncertainty. The results suggest that the relative time increase of the A2C agent is higher than for the DD equivalent. The computation times for the DD model increased around 50 to 60 per cent when considering five additional airports and do not change when including FPU, given that the model is computed after the revelation of the stochastic value and it is computed as a deterministic problem. On the other hand, the A2C seemed to be more sensitive to the increase in the problem size. The computation time increased by about 60 to 100 per cent when increasing the network size and slightly increases when introducing the FPU. Still, this computational time of the A2C does not compromise the practicability of the proposed technique. Even for the case with 210 possible routes and including both demand and fuel price uncertainty, the trained A2C algorithm can produce a solution to this multi-stage stochastic problem in a matter of seconds.

Table 6
DD and A2C techniques performance for the four scenarios. On the right-hand side, the 95% confidence intervals for the Relative Scores.

Scenario	1	2	3	4
N	10	15	10	15
FPU	No	No	Yes	Yes
TS_{A2C}	85.90	87.29	50.40	60.13
Var_{A2C}	1.95	1.69	5.95	7.74
TS_{DD}	85.59	84.65	48.39	45.39
Var_{DD}	1.71	1.85	6.64	6.72
RS_{A2C} [%]	100.04	103.12	104.15	132.47
δt_{A2C} [s]	3.284	7.278	4.772	7.939
δt_{DD} [s]	15.29	24.68	15.75	24.78

N	FPU	lb_{RS}	RS_{A2C}	ub_{RS}
10	No	98.45	100.04	102.37
15	No	101.16	103.12	104.99
10	Yes	97.55	104.15	111.26
15	Yes	124.28	132.47	141.27

Table 7
Probabilistic fleet plan — A2C solution for the AC Type 2 fleet for Scenario 4.

Time step 1		Time step 2		Time step 3		Time step 4		Time step 5	
Fleet size	%	Fleet size	%	Fleet size	%	Fleet size	%	Fleet size	%
30	6,1	[30, 38[	9,7	[30, 38[	3,2	[30, 44[	11,4	[30, 46[	6,8
34	93,9	38	68,9	[38, 42[	10,3	[44, 46[	5,5	[46, 48[	5,5
		39	21,3	42	48,6	46	32,4	[48, 50[	6,1
				43	22,8	47	19,4	50	23,8
				44	15	48	18,3	51	13,4
						49	13	52	15,9
								53	16,5
								54	12

5.5. Analysis and discussion

In this section, we focus on Scenario 4. First we will analyse the resulting probabilistic fleet plan obtained and discuss it can be used in practice. Then we analyse the influence of the fuel prices generated and the number of fleet modifications on the performance of the DD model and A2C algorithm.

5.5.1. Fleet planning analysis

The fleet plan obtained by the A2C agent for Scenario 4 suggests that the number of aircraft of *AC Type 1* should be kept constant and equal to 40 aircraft. The demand growth expected for most of the markets should be captured by increasing the size of the *AC Type 2* fleet. The fleet for this fleet is summarised in Table 7. The probabilities were computed by considering the fleet sizes suggested by the A2C agent for the 1000 episodes of future demand and fuel price values considered.

This probabilistic fleet plan can help the decision-maker to understand that, for instance:

- most likely, four new *AC Type 2* aircraft need to be acquired in the first time step (this was the best solution for 93.9 per cent of the episodes considered);
- for the following time step, there are 68.9 per cent of chances that the ideal fleet will be composed of 38 *AC Type 2* aircraft, and for 21,3 per cent of the episodes, the fleet should have 39 aircraft of that type;
- in the long term, there is an 81.6 per cent probability that in the coming 10 years the fleet has to increase from the current 30 aircraft to a fleet of 50 to 54 aircraft.

5.5.2. Fuel price relation

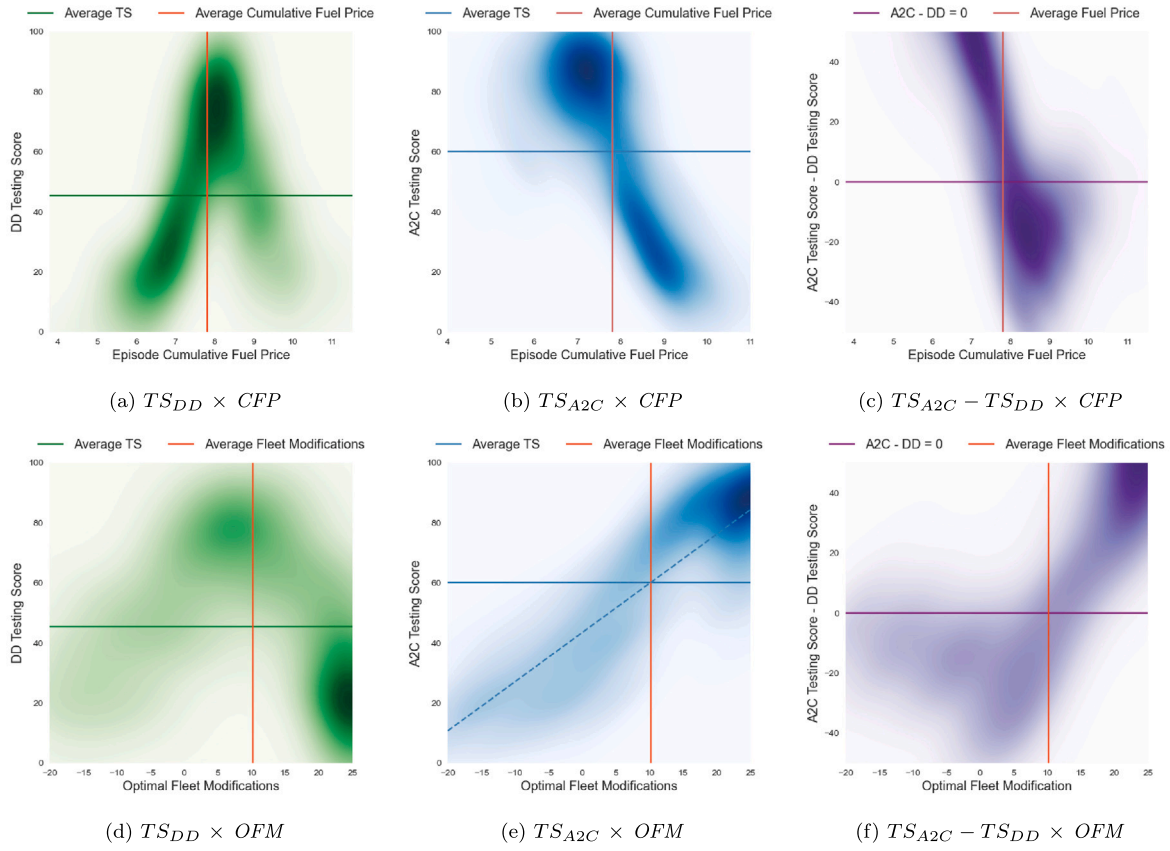
The relationship between the fuel price and the Testing Score of episodes solved by the algorithms is analysed using *density maps*. In Fig. 7, the Testing Scores of the DD model (Fig. 7(a)) and of the A2C algorithm (Fig. 7(b)) are plotted against the Cumulative Fuel Price (CFP) generated in the $n = 1000$ testing episodes. For reference, the average score obtained by the two techniques is represented with the horizontal line, while the vertical line indicates the average CFP generated. The intensity of the colour indicates the number of episodes observed in a specific area of the map, with a darker colour indicating a higher number of observations. The density plot of the fuel price values is shown in Fig. 8(a). The observations approximately followed a normal curve with an average value slightly lower than 1.6 units per time step in each episode.

The DD model performed well, i.e., it obtained higher Testing Scores, for these episodes with an average CFP. More specifically, we observed that the best performance of the DD model was for cases in which the generated CFP was slightly higher than the mean. As expected, the DD model lacks the flexibility to adapt and under-performed for episodes the CFP deviated from the average. On the other hand, the A2C algorithm performed very well for all episodes in which the CFP value was lower or equal to the average value, and performed worst for higher CFP. This suggests that the A2C agent was more adaptable given that it was able to discover an optimal policy for a higher range of CFP values and, in particular, for the cases with high expectancy.

The direct comparison between the two methods is presented in Fig. 7(c). A2C algorithm outperformed the DD model when we have a positive value (upper half of the map), and DD outperformed A2C when we have a negative value. The figure shows that the deterministic model was unable to manage the added uncertainty properly. It used a deterministic forecast of the fuel prices, which resulted in poor performance due to the high volatility. By contrast, the A2C agent was able to utilise the available information and knowledge about the fuel prices evolution better, showing high performances for situations with a high frequency. This resulted in more satisfactory performance, suggesting that the A2C agent was capable of properly adapting to the volatile situation and learning the actions that maximise the expected profit.

5.5.3. Fleet modifications

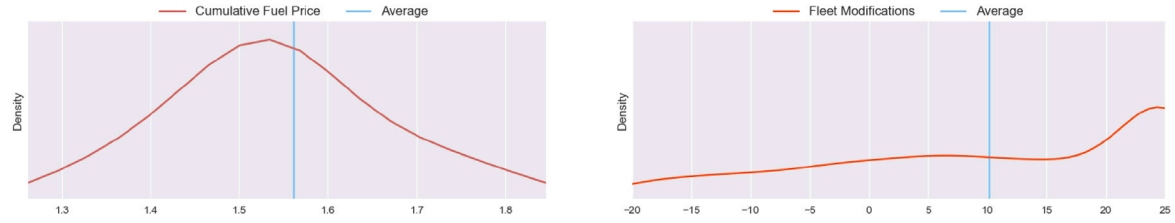
The relationship between the Testing Score and the corresponding optimal amount of Optimal Fleet Modifications (OFM) was also investigated (Figs. 7(d)–7(f)). Positive modifications indicate acquisitions and negative modifications indicate the disposal of aircraft. The cases where the OFM diverge from the average are of particular interest, as these may indicate the robustness of the technique's solutions. The density plot of the optimal number of OFM is visualised in Fig. 8(b). It can be observed that the OFM per episode was more volatile than the CFP, and that there was an accumulation around a high positive amount of fleet modifications.



(a) $TS_{DD} \times CFP$ (b) $TS_{A2C} \times CFP$ (c) $TS_{A2C} - TS_{DD} \times CFP$

(d) $TS_{DD} \times OFM$ (e) $TS_{A2C} \times OFM$ (f) $TS_{A2C} - TS_{DD} \times OFM$

Fig. 7. Density maps of the Testing Scores against the Cumulative Fuel Prices (CFP) and the Optimal Fleet Modifications (OFM) for Scenario 4. Vertical and horizontal lines represented the average values.



(a) CFP (b) OFM

Fig. 8. Density plots of the average Cumulative Fuel Prices (CFP) per time step and the Optimal Fleet Modifications (OFM) for the 15 airport network.

From Fig. 7(d) it can be observed that the DD model performed well for moderate OFM and showed a deteriorating performance whenever the optimal decisions deviated from the average value. This is particularly the case for cases in which the optimal number of fleet modifications was higher than 15. For these frequent cases, the DD model had a poor performance. For the A2C algorithm (Fig. 7(e)), there was almost a linear relationship between the OFM and the performance of the algorithm. Once again, the agent was capable of finding a policy to deal with the most frequent cases, without deteriorating too much the performance for less frequent cases. It performed particularly well for the cases with a high number of fleet acquisitions (with modifications above 15), and it was not much worst than the DD model for a few cases for which the OFM should involve the disposal of aircraft. This shows that the A2C agent was able to adapt properly to the frequent occurrence of a high amount of OFM, and act accordingly. The DD model, because of the limited forecasting capabilities, returned sub-optimal solutions for most of the cases.

6. Conclusion

This article presented the first application of an Advantage Actor–Critic (A2C) reinforcement learning agent within the airline planning domain. Further, for the first time, we considered more than one source of uncertainty when solving the multi-stage fleet planning problem. Besides the traditional demand uncertainty, we implemented the highly-volatile fuel price uncertainty also as a stochastic variable.

For a case study with ten airports and only demand uncertainty, the A2C approach outperformed a deterministic (DD) model and a Deep Q-Network (DQN) RL approach. A2C provided a solution with a profit value better than DQN and slightly better than DD in a baseline scenario. Moreover, A2C required less than 25% of the computation time of DQN, although the use of MILP in the reward function of DQN gave the latter some advantages inconsistently, but not in a statistically significant way.

The advantages of using the A2C when compared to the DD model are even more evident from the results for problems we extended the

size of the network and introduced fuel price uncertainty. A2C was consistently capable of finding better solutions and has shown high adaptability to the unpredictability of the fleet planning process. With respect to computational efficiency, the A2C agent requires a significant amount of time to be trained for bigger and more complex models. This increase in conversion time is relatively higher than the change in solving time for the DD model but it is not comparable with the time of solving the stochastic multi-stage fleet planning problem. Furthermore, after training, the A2C agent generates solutions nearly instantly.

This article has shown the potential of the A2C algorithm to solve the multi-stage fleet planning problem under uncertainty. In particular, we have shown how the resulting multi-stage probabilistic fleet plan can be used to support the strategic fleet decisions of an airline. We believe that this probabilistic assessment is the best way to help the decision-maker to take short-term decisions while foreseeing the general long-term evolution of the fleet considering a large set of future scenarios. The potential of our approach can be further explored in future research. For example, the A2C was proven to be suitable to handle more than one source of uncertainty. Given its adaptability and computation efficiency, this approach may be used to consider other sources of uncertainty, such as the presence of competition and its effect on demand. Another aspect to consider is the correlation between these sources of uncertainty, including the impact of fuel prices on demand.

To reduce the size of the action space, we assumed that at each time step the airline only considers modifications for one aircraft type. While being a common situation in reality, it limits the solution options to solve the problem. The RL approach proposed should be further extended to consider larger action spaces without compromising the training time and algorithm effectiveness. Nevertheless, it should be noted that the adaptivity of the agent to more extreme optimal fleet modifications was higher than that of the deterministic model. Considering that the frequency of these extremes increases when the action space grows, the relatively good performance of the agent could increase accordingly.

An increased action space would require additional exploration by the agent and a more demanding training process. Therefore, some effort should also be put into optimising the convergence of the RL agent by studying the causes of over-fitting (Section 4.3). Identifying these causes in an early stage can have a significant influence on the efficiency of hyper-parameter tuning. Further, employing hyper-parameter search methods (e.g., as discussed in Bergstra et al., 2013) to A2C applications such as that addressed in this article can be studied.

Acknowledgements

Thanks to M. de Koning for the discussions. This research was partially supported by TAILOR, funded by the EU Horizon 2020 programme under grant 952215, and by Epistemic AI, funded by the EU Horizon 2020 programme under grant 964505.

References

Andrade, Pedro, Silva, Catarina, Ribeiro, Bernardete, Santos, Bruno F., 2021. Aircraft maintenance check scheduling using reinforcement learning. Aerospace 8 (4), http://dx.doi.org/10.3390/aerospace8040113.

Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A., 2017. Deep reinforcement learning: A brief survey. IEEE Signal Process. Mag. 34 (6), 26–38. http://dx.doi.org/10.1109/MSP.2017.2743240.

Aucott, Michael, Hall, Charles, 2014. Does a change in price of fuel affect GDP growth? An examination of the U.S. data from 1950–2013. Energies 7 (10), 6558–6570. http://dx.doi.org/10.3390/en7106558.

Balakrishna, Poornima, Ganesan, Rajesh, Sherry, Lance, 2010. Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa Bay departures. Transp. Res. C 18 (6), 950–962. http://dx.doi.org/10.1016/j.trc.2010.03.003.

Barndorff-Nielsen, Ole E., Shepard, Neil, 2001. Non-gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics. J. R. Stat. Soc. Ser. B Stat. Methodol. 63 (2), 167–241. http://dx.doi.org/10.1111/1467-9868.00282.

Bazargan, Massoud, Hartman, Joseph, 2012. Aircraft replacement strategy: Model and analysis. J. Air Transp. Manag. 25, 26–29. http://dx.doi.org/10.1016/j.jairtraman.2012.05.001.

Belobaba, Peter, Odoni, Amedeo, Barnhart, Cynthia, 2009. The Global Airline Industry. John Wiley & Sons, Ltd, ISBN: 978-0-470-74077-4.

Bergstra, James, Yamins, Daniel, Cox, David D., 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proceedings of the 30th International Conference on Machine Learning. ICML 2013, Atlanta, GA, USA, 16-21 June 2013, In: JMLR Workshop and Conference Proceedings, vol. 28, JMLR.org, pp. 115–123, URL http://proceedings.mlr.press/v28/bergstra13.html.

Bertsekas, Dimitri, 2005. Dynamic Programming and Optimal Control. Athena Scientific, Nashua, NH, USA.

Carreira, Joana S., Lulli, Guglielmo, Antunes, António P., 2017. The airline long-haul fleet planning problem: The case of TAP service to/from Brazil. European J. Oper. Res. (ISSN: 03772217) 263 (2), 639–651. http://dx.doi.org/10.1016/j.ejor.2017.05.015.

Chaiyapo, Nattiya, Phewchean, Nattakorn, 2017. An application of Ornstein-Uhlenbeck process to commodity pricing in Thailand. Adv. Difference Equ. 2017 (1), 179.

Chen, Bing, Bai, Ruibin, Li, Jiawei, Liu, Yueni, Xue, Ning, Ren, Jianfeng, 2020. A multiobjective single bus corridor scheduling using machine learning-based predictive models. Int. J. Prod. Res. 1–16. http://dx.doi.org/10.1080/00207543.2020.1766716.

Clarke, Michael, Smith, Barry, 2004. Impact of operations research on the evolution of the airline industry. J. Aircr. 41 (1), 62–72, URL http://arc.aiaa.org.

Clemente, Alfredo V., Castejón, Humberto N., Chandra, Arjun, 2017. Efficient parallel methods for deep reinforcement learning. URL http://arxiv.org/abs/1705.04862.

Cristobal, M. Pilar, Escudero, Laureano F., Monge, Juan F., 2009. On stochastic dynamic programming for solving large-scale planning problems under uncertainty. Comput. Oper. Res. 36 (8), 2418–2428. http://dx.doi.org/10.1016/j.cor.2008.09.009.

Dantzig, George B., Fulkerson, Delbert R., 1954. Minimizing the number of tankers to meet a fixed schedule. Nav. Res. Logist. Q. 1 (3), 217–222. http://dx.doi.org/10.1002/nav.3800010309.

de Koning, Mathias, 2020. Fleet Planning Under Demand Uncertainty: A Reinforcement Learing Approach. Technical Report, Delft University of Technology, URL http://resolver.tudelft.nl/uuid:67125be4-e9d3-46c6-9983-300d71b3511f.

Frikha, Noufel, Lemaire, Vincent, 2018. Joint modelling of gas and electricity spot prices. Appl. Math. Finance 20 (1), 69–93. http://dx.doi.org/10.1080/1350486X.2012.658220.

Goodfellow, Ian, Bengio, Yoshua, Courville, Aaron, 2016. Deep Learning. The MIT Press, Cambridge, MA, USA.

Gould, J., 1969. The size and composition of a road transport fleet. Oper. Res. Q. 20 (1), 81. http://dx.doi.org/10.2307/3008537.

Grondman, Ivo, Busoniu, Lucian, Lopes, Gabriel A.D., Babuška, Robert, 2012. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. IEEE Trans. Syst. Man Cybern. C 42 (6), 1291–1307. http://dx.doi.org/10.1109/TSMCC.2012.2218595.

Haarnoja, Tuomas, Zhou, Aurick, Abbeel, Pieter, Levine, Sergey, 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proc. 35th International Conference on Machine Learning. ICML 2018, ISBN: 9781510867963, pp. 2976–2989.

Hsu, Chaug Ing, Li, Hui Chieh, Liu, Su Miao, Chao, Ching Cheng, 2011. Aircraft replacement scheduling: A dynamic programming approach. Transp. Res. E 47 (1), 41–60. http://dx.doi.org/10.1016/j.tre.2010.07.006.

Ibe, Oliver C., 2013. Brownian motion. In: Markov Processes for Stochastic Modeling, second ed. Elsevier, ISBN: 9780124077959, pp. 263–293. http://dx.doi.org/10.1016/b978-0-12-407795-9.00009-8.

Khoo, Hooi Ling, Teoh, Lay Eng, 2014. An optimal aircraft fleet management decision model under uncertainty. J. Adv. Transp. 48, 798–820. http://dx.doi.org/10.1002/atr.

Lam, Shao Wei, Lee, Loo Hay, Tang, Loon Ching, 2007. An approximate dynamic programming approach for the empty container allocation problem. Transp. Res. C (ISSN: 0968090X) 15 (4), 265–277. http://dx.doi.org/10.1016/j.trc.2007.04.005.

List, George F., Wood, Bryan, Nozick, Linda K., Turnquist, Mark A., Jones, Dean A., Kjeldgaard, Edwin A., Lawton, Craig R., 2003. Robust optimization for fleet planning under uncertainty. Transp. Res. E 39 (3), 209–227. http://dx.doi.org/10.1016/S1366-5545(02)00026-1.

Listes, Ovidiu, Dekker, Rommert, 2005. A scenario aggregation-based approach for determining a robust airline fleet composition for dynamic capacity allocation. Transp. Sci. 39 (3), 367–382. http://dx.doi.org/10.1287/trsc.1040.0097.

Mnih, Volodymyr, Badia, Adria Puigdomenech, Mirza, Lehdi, Graves, Alex, Harley, Tim, Lillicrap, Timothy P., Silver, David, Kavukcuoglu, Koray, 2016. Asynchronous methods for deep reinforcement learning. In: 33rd International Conference on Machine Learning, Vol. 4. ICML 2016, 4, ISBN: 9781510829008, pp. 2850–2869.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K., Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, Hassabis, Demis, 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533. http://dx.doi.org/10.1038/nature14236.

Naumann, Marc, Suhl, Leena, 2013. How does fuel price uncertainty affect strategic airline planning? Oper. Res. 13 (3), 343–362. http://dx.doi.org/10.1007/s12351-012-0131-0.

Novoa, Clara, Storer, Robert, 2009. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. European J. Oper. Res. 196 (2), 509–515. http://dx.doi.org/10.1016/j.ejor.2008.03.023.

Ogbogbo, Chisara Peace, 2018. Modeling crude oil spot price as an Ornstein - Uhlenbeck process. Int. J. Math. Anal. Optim. Theory Appl. 2018, 261–275.

Oum, Tae Hoon, Zhang, Anming, Zhang, Yimin, 2000. Optimal demand for operating lease of aircraft. Transp. Res. B 34 (1), 17–29. http://dx.doi.org/10.1016/S0191-2615(99)00010-7.

Pantuso, Giovanni, Fagerholt, Kjetil, Wallace, Stein W., 2015. Solving hierarchical stochastic programs: Application to the maritime fleet renewal problem. INFORMS J. Comput. 27 (1), 89–102. http://dx.doi.org/10.1287/ijoc.2014.0612.

Pantuso, Giovanni, Fagerholt, Kjetil, Wallace, Stein W., 2016. Uncertainty in fleet renewal: A case from maritime transportation. Transp. Sci. 50 (2), 390–407. http://dx.doi.org/10.1287/trsc.2014.0566.

Powell, Warren B., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality: Second Edition. ISBN: 9781118029176, http://dx.doi.org/10.1002/9781118029176.

Powell, Warren B., Bouzaiene-Ayari, Belgacem, Lawrence, Coleman, Cheng, Clark, Das, Sourav, Fiorillo, Ricardo, 2014. Locomotive planning at Norfolk Southern: An optimizing simulator using approximate dynamic programming. Interfaces 44 (6), 567–578. http://dx.doi.org/10.1287/inte.2014.0741.

Repko, Martijn G.J., Santos, Bruno F., 2017. Scenario tree airline fleet planning for demand uncertainty. J. Air Transp. Manag. (ISSN: 09696997) 65, 198–208. http://dx.doi.org/10.1016/j.jairtraman.2017.06.010.

Requeno García, Laura, 2017. Multi-Period Adaptive Fleet Planning Problem with Approximate Dynamic Programming. Technical Report, Delft University of Technology, URL http://resolver.tudelft.nl/uuid:228626b4-129f-492c-b1d4-8eb859941bbc.

Sa, Constantijn A.A., Santos, Bruno F., Clarke, John Paul B., 2019. Portfolio-based airline fleet planning under stochastic demand. Omega http://dx.doi.org/10.1016/j.omega.2019.08.008.

Sayarshad, Hamid Reza, Ghoseiri, Keivan, 2009. A simulated annealing approach for the multi-periodic rail-car fleet sizing problem. Comput. Oper. Res. 36 (6), 1789–1799. http://dx.doi.org/10.1016/j.cor.2008.05.004.

Schick, G.J., Stroup, J.W., 1981. Experience with a multi-year fleet planning model. J. Manage. Sci. 9 (4), 389–396. http://dx.doi.org/10.1016/0305-0483(81)90083-9.

Shapiro, Alexander, Dentcheva, Darinka, Ruszczyński, Andrzej, 2014. Lectures on Stochastic Programming. SIAM, http://dx.doi.org/10.1137/1.9780898718751.

Shihab, Syed A.M., Wei, Peng, 2021. A deep reinforcement learning approach to seat inventory control for airline revenue management. J. Revenue Pricing Manag. http://dx.doi.org/10.1057/s41272-021-00281-7.

Shube, D.P., Stroup, J.W., 1975. Fleet planning model. In: Winter Computer Simulation Conference Proceedings. pp. 45–50, URL https://informs-sim.org/wsc75papers/1975_0009.pdf.

Simão, Hugo P., George, Abraham, Powell, Warren B., Gifford, Ted, Nienow, John, Day, Jeff, 2010. Approximate dynamic programming captures fleet operations for Schneider National. Interfaces 40 (5), 342–352. http://dx.doi.org/10.1287/inte.1100.0510.

Sutton, Richard S., Barto, Andrew G., 2018. Reinforcement learning: An introduction, second edition. The Lancet. (ISSN: 01406736) ISBN: 0262193981, http://dx.doi.org/10.1016/S0140-6736(51)92942-X.

Tong, Zhao, Deng, Xiaomei, Ye, Feng, Basodi, Sunitha, Xiao, Xueli, Pan, Yi, 2020. Adaptive computation offloading and resource allocation strategy in a mobile edge computing environment. Inform. Sci. 537, 116–131. http://dx.doi.org/10.1016/j.ins.2020.05.057.

Uhlenbeck, George E., Ornstein, Leonard S., 1930. On the theory of the Brownian motion. Phys. Rev. (ISSN: 00319015) 36 (1), 823–841,

U.S. Energy Information Administration, 2020. U.S. Gulf Coast kerosene-type jet fuel spot price FOB. URL https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=pet&s=eer_epjk_pf4_rgc_dpg&f=m.

Wang, Jane X, Kurth-Nelson, Zeb, Tirumala, Dhruva, Soyer, Hubert, Leibo, Joel Z, Munos, Remi, Blundell, Charles, Kumaran, Dharshan, Botvinick, Matt, 2017. Learning to reinforcement learn.

Wyatt, J.K., 1961. Optimal fleet size. Oper. Res. Q. (ISSN: 14732858) 12 (3), 186. http://dx.doi.org/10.2307/3006775.