JST: Smart Systems and Devices
Volume 34, Issue 2, May 2024, 054-065

Minimizing Flight Hours Losses by Utilizing Deep-Q Learning Techniques in Scheduling Aircraft Maintenance

Manh-Tuan Ha*, Minh-Hieu Nguyen, Quang-Hung Pham
Hanoi University of Science and Technology, Ha Noi, Vietnam
**Corresponding author email: tuan.hamanh@hust.edu.vn*

Abstract

Currently, there are insufficiently good or efficient tools or procedures for long-term scheduling of aviation maintenance activities in the world, and Vietnam specifically. Airlines are impacted by a variety of elements, including the quantity of aircraft they operate, their capacity, their personnel resources, their maintenance resources, and unforeseen and urgent occurrences that cause schedule disruptions. There are no fast, automatic solutions to the above-described problems existing in aviation today. The reinforcement learning method as described here could potentially be one answer to these problems. The idea is to plan in years running across a specified period such that the aircraft is brought as close as possible to the inspection deadline. From there, the airworthiness of the aircraft increases while the maintenance inspection decreases, reducing the cost of maintenance. Application optimization of the scheduling plan is done using the Deep Q-learning method. The results achieved by the Q-learning and Deep Q-learning algorithms are better in terms of computation times as compared to the other current techniques. The research results of the checks showed reinforcement learning potential in dealing with this problem, where the fly hours loss of planned inspections was reduced by using data from Vietnam Airlines. Computational experiments show that our methods adapt for different purposes and settings of reality. After teaching the model with these simulated conditions, they show how well a reinforcement learning application quickly arrives at lean repair plans.

Keywords: Aircraft maintenance, maintenance check scheduling, reinforcement learning, Q-learning, deep Q-learning.

1. Introduction

The air transport system stands as one of the most intricate man-made transportation networks globally. Over the past decade, the aviation industry has experienced substantial growth, emerging as a significant contributor to the global GDP surpassing other industries and sectors. Consequently, heightened competition among airlines has ensued, prompting price reductions in airfares, thus resulting in a substantial surge in passenger demand.

A series of sequential operations precedes aircraft departure. Firstly, flight schedules are devised based on the company's strategic blueprint and marketing considerations, including passenger demand. This involves the airline determining destination cities, departure times, and flight frequencies. Secondly, fleet allocation is addressed, with airlines assigning aircraft types to scheduled routes based on factors such as maximum total range, seating capacity, and operational expenses. Subsequently, a sequence of flights ensues, with each aircraft undergoing maintenance checks at designated intervals. Finally, crew management is undertaken, involving the assignment of crew members to each aircraft, taking into account tax implications and collective bargaining agreements.

To ensure the smooth and stable operation of the aviation industry, several critical factors come into play, including operators and services. Among these factors, maintenance emerges as a pivotal aspect contributing to both safety and efficiency within the aviation sector.

Maintenance check intervals are established according to aircraft usage parameters, which are defined by flight hours (FH), flight cycles (FC), and calendar days (DY). When an aircraft reaches its service limit based on any of these three conditions, it becomes eligible for routine maintenance checks. The four primary types of periodic maintenance checks in the aviation industry are the A-check, B-check, C-check, and D-check [1]. An A-check typically involves an internal or external inspection of the aircraft, with certain areas accessible (such as for oil inspection and maintenance, filter replacement, and lubrication). Conducted approximately every two to three months, the A-check is considered a relatively minor inspection, usually completed within a single day. The C-check involves a comprehensive

ISSN 2734-9373
https://doi.org/10.51316/jst.174.etsd.2024.34.2.7
Received: Feb 21, 2024; revised: Mar 14, 2024:
accepted: Mar 15, 2024

examination of individual systems and components to assess their usability and functionality. Typically conducted every 18 to 24 months, this check lasts between 1 to 3 weeks. On the other hand, the D-check focuses on inspecting critical structural elements such as the airframe, support structures, and wings. This intensive inspection occurs every 6 to 10 years and can last from three weeks to two months. While some documents mention the B-check, in practice, many airlines opt to integrate B-check tasks into the A-check process [2]. Similarly, airlines often incorporate numerous tasks from D-checks into C-checks, effectively extending the duration of the C-check beyond its standard timeframe. Due to the substantial impact of these maintenance checks on the aircraft's flight readiness and the significant allocation of resources required for their execution, they are typically scheduled several months in advance, typically within a 3 to 5-year timeframe [2].

Scheduling maintenance checks for an entire fleet of aircraft presents a demanding and intricate challenge. In practice, aircraft maintenance schedules are typically devised based on the expertise of maintenance operators. However, this approach poses significant drawbacks, primarily in terms of time consumption and the potential for suboptimal solutions. Maintenance operators for a large fleet must dedicate several days or even weeks to plan A and C-checks according to each aircraft's inspection cycle and the available maintenance resources of the airline. Any deviations or issues arising during A or C-checks necessitate adjustments to the schedule, leading to continual shifting of inspections to earlier or later timeframes until a viable schedule is established. As this planning method relies on manual processes, the primary focus is often on identifying a feasible maintenance schedule for the fleet rather than an optimal one. Consequently, the traditional manual maintenance planning method inevitably results in reduced aircraft utilization and a higher frequency of A and C-checks in the long term, thereby escalating aircraft maintenance costs.

Presently, airlines are placing growing emphasis on enhancing aircraft availability and optimizing maintenance planning for greater efficiency. Aircraft maintenance stands as one of the primary direct operating costs and holds significant weight on an airline's balance sheet. As reported by [3], approximately 11% of an airline's total costs are allocated to aircraft maintenance. In 2014, this translated to an average expenditure of 295 million USD per year per airline [4]. The long-term economic and operational advantages of adopting a more efficient approach to maintenance are evident. For instance, a typical C-check for the A320 series can incur costs ranging from 150 thousand USD to 350 thousand USD [1], while an A-check typically costs between 10 thousand USD to 15 thousand USD. Moreover, each additional day of operation can yield additional revenue, with commercial revenue ranging from 75 thousand USD to 120 thousand USD, depending on aircraft usage. However, the complexity arises from the fact that A- and C-check planning issues are two interconnected combinatorial problems. The decision to schedule or postpone an A/C check on an aircraft today will influence the future utilization of the aircraft and, consequently, the necessity for A/C checks in the future. Solving such problems presents a considerable challenge, often necessitating the use of heuristics and algorithms to assist in decision-making [5].

Annually, Vietnam attracts a significant influx of 70 - 80 million visitors, including approximately 10 million international travelers. In the initial two months of 2023, both flight frequency and passenger volume surged by over 90% compared to the corresponding period, signaling a notable recovery in the number of international visitors. Since the year's outset, airlines have vigorously promoted domestic demand while seeking opportunities in the gradually rebounding international market. Given the substantial scale of operations, maintenance of each airline's aircraft assumes paramount importance. However, despite their relatively young fleet age compared to global counterparts, the challenge of training engineers, mechanics, and establishing adequate maintenance centers for these airlines persists due to prevailing circumstances.

Presently, in Vietnam, the majority of airlines lack internationally accredited training facilities, relying on traditional, non-automated methods that heavily involve human personnel. Furthermore, the maintenance operations of most airlines in the country are reliant on foreign service providers. Moreover, the software utilized for aviation maintenance tasks is outdated and ill-suited to meet the present requirements and demands of the industry.

Based on the maintenance and service processes of airlines in Vietnam, it is apparent that these carriers rely on manual scheduling for periodic checks. This manual approach is time-consuming, often taking days or even weeks to devise a viable maintenance plan. Consequently, these airlines typically formulate feasible maintenance schedules rather than optimal ones, potentially resulting in suboptimal solutions. Such inefficiencies can adversely impact aircraft performance, leading to financial losses for the airlines. Therefore, there is a pressing need to enhance existing processes and strive for optimized maintenance planning methodologies.

Consequently, the demand for tools to optimize processes and scheduling arrangements within Vietnam's aviation industry has surged. Companies that have already achieved self-reliance are actively seeking such methods, while others in the process of transitioning towards self-reliance will also require these optimal solutions.

The main solution is to apply machine learning methods to replace humans in maintenance planning. This benefits both human resources and the quality of results by optimizing input data and implementation processes.

2. Problem Formulation

Each aircraft undergoes grounding for maintenance checks at predetermined intervals as outlined in the aircraft Maintenance Plan Document (MPD). These intervals are determined by three key metrics that delineate the lifespan of an aircraft: flight hours (FH), flight cycles (FC), and calendar days (DY). Each type of maintenance check usually has its own specified maximum duration, and the associated usage metrics are reset to zero immediately after the completion of the check.

Tolerances, as stipulated in the aircraft's Maintenance Plan Document (MPD), can be employed to schedule maintenance checks beyond the maximum interval. However, this practice should be exercised cautiously, as it necessitates clearance from civil aviation authorities. Additionally, tolerances are deducted from the maximum interval of the subsequent check, with flight hours (FH), flight cycles (FC), and calendar days (DY) serving as the metrics for such deductions.

The purpose of this task is to get the airplane as close to the check deadline as possible before landing to undertake repairs. The three cumulative time parameters indicated above (FH, FC, and DY) are as close to the critical time as feasible, decreasing the number of checks and boosting the aircraft's airworthiness.

When a lighter maintenance check is scheduled to follow a heavier one, airlines often opt to consolidate the two checks. For instance, if both an A-check and a C-check are due at intervals separated by a specified number of days, they are combined. This check merge allows the same aircraft to avoid landing twice in a short period of time without necessarily increasing the flight time. This check merge prevents the same aircraft from landing again in a short period of time without extending the C check time. The consequence of doing so is that it will increase the number of A checks in the long run.

Some airlines also necessitate a minimum interval between the start dates of two identical maintenance checks. This requirement primarily involves the preparation of resources such as labor, tools, aircraft replacement parts, and the hangar facilities.

2.1. Assumptions

The following are the assumptions considered to determine the long-term maintenance inspection planning method:

- The minimum time unit for maintenance planning is 1 DY;
- The cumulative duration of an aircraft can be measured in terms of FH, FC, and DY;
- The cumulative duration of each aircraft can be determined by the data provided by the airline or can be estimated using historical data;
- The time taken for each check may be determined by the airline or may be estimated using historical data or common conventions;
- There are 4 slots per day to perform an A/C-check;
- Each A/C-check uses only one hangar slot during the total check time;
- C-check has a higher priority than A-check;
- A-check can be merged in C-check without increasing the time of C-check or taking up a slot.

Indeed, maintenance times can fluctuate throughout the year due to various factors. During peak commercial seasons, such as the New Year, Lunar New Year, holidays, and summer periods, the volume of maintenance work tends to decrease significantly. In some cases, even heavy maintenance tasks may be deferred during these periods. This adjustment is made to accommodate increased operational demands and to minimize disruptions to flight schedules during peak travel times.

The priority given to C-check planning is underscored by two primary reasons. Firstly, it pertains to the feasibility of merging checks. When scheduling an A-check, the precise start and end dates of the upcoming C-check are considered to determine if merging is viable. Secondly, the duration of the C-check, which typically spans several weeks, significantly influences the scheduling of the A-check. This is because the accumulated parameters for the A-check do not increase while the aircraft undergoes the C-check, as it remains grounded for inspection. Therefore, the timing and duration of the C-check directly impact the scheduling of subsequent A-checks.

Maintenance checks have a cycle of several months or years, while the airline only plans to operate the aircraft for a short time, so the geographical location of the aircraft does not need to be considered.

2.2. Mathematical Formulas and Constraint Conditions

The mathematical formulas for the problem are generated based on the problem determined in Section 2.1. The following symbols are used in mathematical formulas:

- $a_{h,t}^{k}$ is a binary variable to indicate if *k* check can be performed in hangar *h* on day *t,*
- C_{t}^{k} binary variable to indicate if there is enough hangar capacity to schedule *k* check of aircraft *i* on day *t,*
- $DY_{i,t}^{k}$, $FH_{i,t}^{k}$ and $FC_{i,t}^{k}$ correspond to the cumulative usages in DY, FH, and FC, respectively, of aircraft *i* on day *t* for *k* check,
- $\tau_{i,k}^{DY}$, $\tau_{i,k}^{FH}$ and $\tau_{i,k}^{FC}$ correspond to the tolerance used in the last type *k* check of aircraft *i*, with respect to the same three metrics,
- D_{i}^{k} is the next due date for type *k* check of aircraft *i*, that is, the day in which the respective interval is reached,
- u_{i}^{k} is the next due date for type *k* check of aircraft *i* using the tolerance,
- y_{i}^{k} is the ending day of the last type *k* check for aircraft *i,*
- z_{i}^{k} is the scheduled day of the next type *k* check for aircraft *i,*
- $g_{i,t}^{k}$ is a binary variable to indicate if aircraft *i* is grounded on day *t* performing *k* check,
- $R_{i,t}^{k}$ is a binary variable to indicate if there is *k* check starting on day *t* for aircraft *i* ,
- L_{i}^{k} is the duration (in working days) of the next type *k* check for aircraft *i* ,
- δ_{i} is the amount of FH lost in the last scheduled check of aircraft *i* ,
- μ_{k} is minimum number of DY between the start of two type *k* checks ,
- $a_{h,t}^{k}$ is variable to indicate if *k* check can be performed in hangar *h* on day *t* ,
- $I_{i,k}^{DY}$ is interval of type *k* check of aircraft *i* in DY,
- $I_{i,k}^{FH}$ is interval of type *k* check of aircraft *i* in FH,
- $I_{i,k}^{FC}$ is interval of type *k* check of aircraft *i* in FC.

From the considered conditions, we will have the following mathematical formulas. Certain conditions must be determined to select the date to schedule the check.

The first is that the hangar is required for the duration of the inspection. We determine the capacity of a specific hangar as follows:

$$a_{h,t}^{k}=\begin{cases}1, & if\ hangar\ h\ is\ available\ for\\ & \quad type\ k\ on\ day\ t\\ 0, & otherwise\end{cases} \tag{1}$$

This variable is used to calculate whether there are enough aircraft slots for *k* check of aircraft *i* on a particular day *t* and for the duration of the check:

$$C_{t}^{k}=\begin{cases}1, & if\ L_{i}^{k}-\sum_{t'=t}^{t+L_{i}^{k}} a_{t'}^{k}=0\\ 0, & otherwise\end{cases} \tag{2}$$

where *t* is the candidate's scheduled check date.

The minimum number of days between the start dates of two consecutive type *k* checks, μ_k (under the conditions set forth by the airline which is $\mu_k = 3$), is a second requirements. $R_{i,t}^{k}$ can be used to calculate the total number of checks beginning in the range $[t-\mu_k, t+\mu_k]$ (aslo $[t-3, t+3]$), where *t* is the candidate day for scheduling the check. If this sum equals zero, *k* check for aircraft *i* can be scheduled for day *t*:

$$R_{t}^{k}=\begin{cases}1, & if\ \sum_{i}\left(\sum_{t'=t-3}^{t+3} R_{i,t'}^{k}\right)=0\\ 0, & otherwise\end{cases} \tag{3}$$

This spacing condition only applies to C checks.

With (2) and (3), we can choose the date to schedule the next check for type *k* check of aircraft *i*, defined by the variable c_{i}^{k}:

$$c_{i}^{k}=max\begin{Bmatrix}\left(t_{min}R_{t_{min}}^{k}C_{i,t_{min}}^{k}\right),\\ \left((t_{min}+1)R_{t_{min}+1}^{k}C_{i,t_{min}+1}^{k}\right),\dots,\\ \left(t_{max}R_{t_{max}}^{k}C_{i,t_{max}}^{k}\right)\end{Bmatrix} \tag{4}$$

where $t_{min}=y_{i}^{k}$ and $t_{max}=D_{i}^{k}$. We shall, of course, strive to avoid employing tolerances in the scheduling process. So, the goal of (4) is to select a date as near to the due date as possible to reduce the quantity of underutilized FH aboard the airplane.

If the value of $c_{i}^{k}=0$, then there are no possible days to schedule the check before the due date. This can occur when there is insufficient hangar availability and/or when the spacing requirement between checks is not followed. The tolerance can be utilized in this scenario, and the check scheduled day is defined using:

$$c_{i}^{k}=max\begin{Bmatrix}\left((t_{max}+1-t_{min})R_{t_{min}}^{k}C_{i,t_{min}}^{k}\right),\\ \begin{pmatrix}(t_{max}+1-t_{min}+1)\\ R_{t_{min}+1}^{k}C_{i,t_{min}+1}^{k}\end{pmatrix},\dots,\\ \begin{pmatrix}(t_{max}+1-t_{max})\\ R_{t_{max}}^{k}C_{i,t_{max}}^{k}\end{pmatrix}\end{Bmatrix} \tag{5}$$

where $t_{min} = D_i^k$; $t_{max} = u_i^k$; $C_{i,t}^k$ is variable to indicate if there is enough hangar capacity to schedule k check of aircraft i on day t. Similarly, the purpose is to select a day as close to the check due date as feasible, but this time with the goal of lowering the amount of tolerance utilized.

Now that a new type k check for aircraft i has been scheduled, we can schedule y_i^k on the last day of this recently scheduled check:

$$y_i^k = c_i^k + L_i^k \tag{6}$$

The amount of FH that is lost because of this scheduling, defined as δ_i, can be computed with:

$$\delta_i = \left| i_{i,k}^{FH} - FH_{i,t}^k \right| \tag{7}$$

As a result, on day $y_i^k + 1$, the cumulative utilization attributes of aircraft i, $DY_{i,t}^k$ (cumulative DY of aircraft i on day t since its last type k check), $FH_{i,t}^k$ (cumulative FH of aircraft i on day t since its last type k check) and $FC_{i,t}^k$ (cumulative FC of aircraft i on day t since its last type k check) are set to 0. The next check, L_i^k (next type k check duration, in DY, of aircraft i), is updated based on a lookup table containing the duration of future checks set by the airline.

The final stage is determining the new due date for k check. To begin, we must update the amount of tolerance utilized in the previous type k check, which must be subtracted from the following interval:

$$\tau_{i,k}^{DY} = max\left\{0, DY_{i,t}^k - \left(D_i^k - y_i^k\right)\right\} \tag{8}$$

$$\tau_{i,k}^{FH} = max\left\{0, FH_{i,t}^k - \left(D_i^k - y_i^k\right) \times fh_i\right\} \tag{9}$$

$$\tau_{i,k}^{FC} = max\left\{0, FC_{i,t}^k - \left(D_i^k - y_i^k\right) \times fc_i\right\} \tag{10}$$

where fh_i and fc_i correspond to the average daily utilization of aircraft i estimated by the airline in FH and FC, respectively, and $t = c_i^k$. $\tau_{i,k}^{DY}$ is tolerance used in the last type k check of aircraft i in DY; $\tau_{i,k}^{FH}$ is tolerance used in the last type k check of aircraft i in FH; $\tau_{i,k}^{FC}$ is tolerance used in the last type k check of aircraft i in FC; D_i^k is next type k check due date of aircraft i; y_i^k is end day of the last type k check of aircraft i; $R_{i,t}^k$ is variable to indicate if there is k check starting on day t for aircraft i; R_t^k is variable to indicate if there is enough space between type k checks when k check is scheduled on day t.

We must also examine the number of days the aircraft is grounded, and its utilization remains unchanged. For example, if a C-check is planned after the last A-check, the grounded days used to complete the C-check do not count toward calculating the next A-check due date. We define the day when the interval for the next type k check is reached as $d_{i,k}^{DY}$, $d_{i,k}^{FH}$ and $d_{i,k}^{FC}$, for each usage metric, without considering aircraft ground time:

$$d_{i,k}^{DY} = y_i^k + I_{i,k}^{DY} - \tau_{i,k}^{DY} \tag{11}$$

$$d_{i,k}^{FH} = y_i^k + \frac{I_{i,k}^{FH} - \tau_{i,k}^{FH}}{fh_i} \tag{12}$$

$$d_{i,k}^{FC} = y_i^k + \frac{I_{i,k}^{FC} - \tau_{i,k}^{FC}}{fc_i} \tag{13}$$

where $I_{i,k}^{DY}$, $I_{i,k}^{FH}$ and $I_{i,k}^{FC}$ represent the type k check intervals in DY, FH, and FC, respectively.

As a result, the due date of the next type k check is defined with:

$$D_i^k = min\left\{\begin{array}{l} d_{i,k}^{DY} + \sum_{t=y_i^k}^{d_{i,k}^{DY}} \sum_k g_{i,t}^k, \\ d_{i,k}^{FH} + \sum_{t=y_i^k}^{d_{i,k}^{FH}} \sum_k g_{i,t}^k, \\ d_{i,k}^{FC} + \sum_{t=y_i^k}^{d_{i,k}^{FC}} \sum_k g_{i,t}^k \end{array}\right\} \tag{14}$$

This is the due date on which the first usage interval is reached after adding the aircraft ground time since the last type k check.

When the tolerance is employed, the final characteristic to be calculated is the due date for the next k check, u_i^k:

$$u_i^k = min\left\{D_i^k + \theta_{i,k}^{DY}, D_i^k + \frac{\theta_{i,k}^{FH}}{fh_i}, D_i^k + \frac{\theta_{i,k}^{FC}}{fc_i}\right\} \tag{15}$$

where $\theta_{i,k}^{DY}$, $\theta_{i,k}^{FH}$ and $\theta_{i,k}^{FC}$, are the maximum tolerances that are allowed for k check, with respect to DY, FH, and FC, respectively.

2.3. Problem Constraints

There are numerous limits in check planning, some of which have been discussed in previous sections. One of the most critical requirements for this challenge is that the aircraft usage conditions associated with each type of inspection never exceed the maximum permitted level. This maximum represents the overall inspection period with a fixed tolerance, less the amount of tolerance used in the previous inspection. This constraint can be built for each metric using the three equations below:

$$DY_{i,t}^k \leq I_{i,k}^{DY} + \theta_{i,k}^{DY} - \tau_{i,k}^{DY} \tag{16}$$

$$FH_{i,t}^k \leq I_{i,k}^{FH} + \theta_{i,k}^{FH} - \tau_{i,k}^{FH} \tag{17}$$

$$FC_{i,t}^k \leq I_{i,k}^{FC} + \theta_{i,k}^{FC} - \tau_{i,k}^{FC} \tag{18}$$

It is also critical to ensure that the number of checks taking place at the same time each day does not exceed the allotted hangar slots:

$$\sum_{t}\sum_{i} g_{i,t}^{k} \le \sum_{h} a_{h,t}^{k},$$
$$t \in \{t_0, \dots, T\}, \quad i \in \{1, \dots, N\} \qquad (19)$$

where $g_{i,t}^{k}$ and $m_{h,t}^{k}$ are the variables mentioned above.

Finally, we create a constraint based on the minimum number of days between the commencement of two consecutive *k*-type checks (in this case, 3 days for type C-check, with no such condition for A-check):

$$\sum_{i}\sum_{t'=t-3}^{t+3} R_{i,t'}^{k} \le 1,$$
$$t \in \{t_0, \dots, T\}, \quad i \in \{1, \dots, N\} \qquad (20)$$

where $R_{i,t}^{k}$ is a binary variable to indicate whether there was *k* check starting on day *t* for aircraft *i*.

2.4. Objective Function

One of the primary objectives of this paper is cost reduction. Reducing the fleet's underutilized FH is the best approach to achieve this goal in a long-term maintenance scheduling dilemma. In the long term, this leads to fewer planned checks, which can have a big effect because every day the airline is not operating costs a lot of money. We can determine the cost of selecting action x_j on state s_j, $C_j(s_j, x_j)$, at each stage of the scheduling process using the following formula:

$$C_j(s_j, x_j) = \begin{cases} \delta_i, & if\ \delta_i \ge 0 \\ -\delta_i P, & if\ \delta_i < 0 \end{cases} \qquad (21)$$

where *P* is a penalty for using the tolerance, δ_i is the unused FH from the last planned check of aircraft *i*, and *i* is the chosen aircraft.

$$\min_{\pi} \mathrm{E}\left\{\sum_{j=0}^{j} C_j\left(s_j, X^{\pi}(s_j)\right)\right\} \qquad (22)$$

Equation (22) can serve as the foundation for determining the objective function, which aims to minimize costs. In this context, $X^{\pi}(s_j)$ represents the ideal scheduling policy function. This function guides the selection of the optimal action in each state, ultimately leading to a reduction in costs.

3. Problem Solving Methods

3.1. Reinforcement Learning

In the field of Reinforcement Learning (RL), machines acquire task-solving abilities by engaging with the environment, executing actions, and making optimal decisions based on the rewards associated with each action [6]. This process mirrors the way humans learn, as it involves trial and error interactions with the environment to determine the most effective strategies.

The diagram in Fig 1 describes how RL works, but before we get into a more detailed explanation, let's get acquainted with some of the terminology used here:

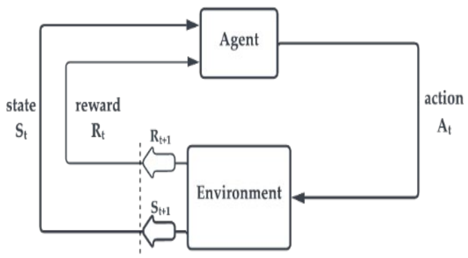

Fig. 1. Interaction of RL components

- Environment is the space where the machine interacts;
- Agent (machine) a subject that interacts with the environment through actions;
- Policy is the policy that the machine uses to take action;
- State describes the current state of the agent;
- Reward reward from the environment corresponding to the action performed;
- Action is what the agent can do.

A complete framework known as Markov Decision Processes (MDPs) is shown in the above graphic. In essence, a mathematical framework for simulating decision-making scenarios is offered by an MDP. In this case, some of the results are random, and the remainder are determined by the decisions made earlier by the agent. The decision maker's reward is contingent upon the course of action selected, considering both the current (S_{t+1}) and previous S_t environmental situations.

The objective is to identify the best course of action that will maximize the expected goal, or R_t. This reward can be calculated using the following formula, which equates to the total reward discounted over time:

$$V(s) = \max_{a}\left(R(s,a) + \gamma V(s')\right) \qquad (23)$$

In which,

- s is A specific state
- a is An action performed by the agent
- s' is Previous state
- γ is Discount factor (we will learn below)
- $R(s, a)$ is Reward function with variable state s and action and returns the value of the reward as the result.
- $V(s)$ is value at a specific state

The max function assists the agent in determining the best course of action, while the discount factor γ helps the agent determine how far away the goal is. The function may still introduce a slight probability of confusion for the agent in situations where multiple options are available for selection. In such cases, the decision-making process is both regulated and stochastic, posing a challenge. The randomness arises from the uncertainty of when the agent might encounter confusion, yet it remains controlled as it adheres to established strategies. However, with minor modifications, this concept can be applied to refine the function:

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')\, V(s') \right) \quad (24)$$

3.2. Q-Learning

The Q-Learning model is like the process mentioned above. However, instead of making decisions about actions based on the value of states *V(s)*, Q-Learning focuses more on evaluating the quality of an action *Q (s, a)*. From above we have the (24):

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')\, V(s') \right) \quad (25)$$

In this formulation we are interested in all states and all possible actions. So when we remove the max function, we will get the equation $R(s,a)+\gamma\sum s'$ $P(s,a,s')V(s')$ and consider it as the value of a created state out for only one possible action. We will take this equation as the action evaluation equation $Q(s,a)$ as follows:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s,a,s')\, V(s') \quad (26)$$

To minimize concurrent calculations and ensure uniformity, further enhancements can be made to refine the equation:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s,a,s') \max_{a'} Q(s',a') \quad (27)$$

The reason we replace $V(s)$ with $maxQ(s,a)$ is that here, we consider the value of a state to be calculated as the largest possible value of $Q(s,a)$. The values calculated from this formula are called Q-values and the agent will learn to calculate Q-values itself and take actions based on these values. Above we have understood how agents make choices based on Q-values, so now let's come to the final part, which is how agents can calculate Q-values themselves.

3.3. Deep Q-Learning

RL is experiencing tremendous growth thanks to the development of new algorithms that allow for solving more complex problems. In this field, complexity is directly related to dimensionality. Classical RL algorithms may not be sufficient to manage large state and action spaces. The dimensionality problem can be solved effectively thanks to the remarkable progress of Deep Learning (DL) algorithms. DL combined with RL creates a new group of algorithms: deep reinforcement learning (DRL). DRL methods can be divided into value-based, function-based, and policy-based methods. The most popular value-function-based method in DRL is Deep Q-Network (DQN). This is a version of Q-Learning where the value function is not calculated in a table but is approximated using an Artificial Neural Network (ANN). The Fig. 2 shows the basic architecture of DQN.

The limitation of Q-learning is that if the state is too large, it will take a lot of space to store the Q-table and slow down the learning time because, during the learning process, the Q-table is accessed continuously. To avoid overfitting, here we will use two neural networks: the main network, called the action-value network, and a secondary network, called the target action-value network. We will create two deep learning networks. We use the first network to derive the Q-values, while the second network contains all the updates during training. After 100,000 updates, we will synchronize. The goal is to keep the interim target Q-values constant, so we won't have any variable targets to follow. Also, parameter changes will not affect the results immediately, and thus, even though the inputs may not be 100% independent and identically distributed, they will not bias the results as mentioned before.

Another problem with DQN is catastrophic forgetting. To reduce this problem, we will use experience replay. We put the last million transformations into a buffer and retrieved a small set of samples of size 32 from this buffer to train the deep learning network. This creates an input data set that is stable enough to train a neural network. When we randomly sample from the playback buffer, the data will be more independent of each other and will ensure independence and equal distribution.

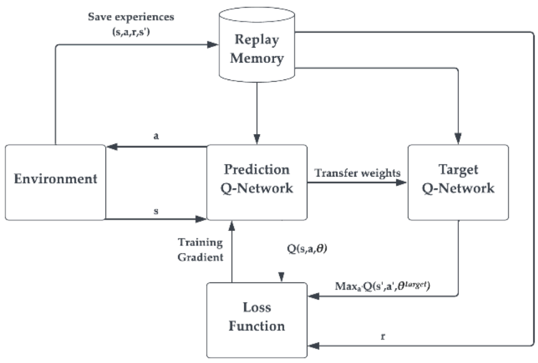

Fig. 2. DQN architecture

This method uses a ‘prediction network’, a ‘target network’ and a ‘memory replay’ to obtain a loss function that will be used to train the prediction network. This function can be defined by the following function:

$$L(\theta) = \left(r + \gamma \max_{a'} Q(s', a', \theta^{target}) - Q(s, a, \theta) \right)^2 \quad (28)$$

4. Experimental Setup

4.1. Fundamental Setup

The created Deep Q-learning algorithm is described in this section, with particular attention paid to the RL components and the algorithm parameters. In addition, it displays the actual maintenance dataset obtained from Vietnam Airlines, which simulates and depicts real-world circumstances.

In this problem, we propose a Deep Q-learning algorithm to optimize the long-term check planning problem. The Fig 3 shows the general workflow of the planning process. At each decision step, the RL agent selects an aircraft i and checks the next type k for that aircraft scheduled on the available date closest to its due date. As mentioned earlier C-check is scheduled first to deal with cases where two types of A and C-check are combined. Then, the due date for the next type k check is calculated and communicated. The necessary aircraft information will be updated.

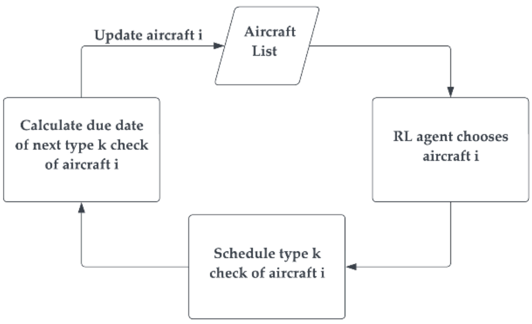

Fig. 3. General workflow of the check scheduling process.

Using experience playback, the agent is trained, and a ε-greedy approach is used to control the amount of exploration. The agent will always give priority to exploration since, in the first episode, ε equals 1. From 0.01 to every episode, this value decays at a linear pace.

Furthermore, the Double Q-learning method, as described in [8], employs a double estimator approach to compute the value of the next state. This method mitigates the overestimation often encountered in action values, constituting its primary advantage. By incorporating an additional network—the target network—in conjunction with the online network to compute action values [9], this approach ensures more accurate estimations. While the online network is responsible for selecting actions and estimating their values, its weights are periodically updated with those of the target network to refine its performance.

Initially, the replay memory is initialized as an empty queue. As detailed in Section 3.3, the agent's experiences are recorded in the replay memory and utilized for model training. Xavier initialization is employed to initialize both the target and online networks [10]. Each episode commences with the current state set to the initial state. Subsequently, actions are selected for each step with a predetermined probability, either randomly or greedily. Each action corresponds to selecting a specific aircraft, as previously mentioned. The selected action is then executed, scheduling the corresponding aircraft's subsequent check. The replay memory computes and stores the reward and the ensuing state, while the current state transitions to the next state. For a minibatch sampled from the replay memory, the target value and associated loss are calculated and utilized to train the online network. The minimum loss is determined as the squared difference between the target value and the predicted value. Finally, the weights of the target network are updated by substituting them with those from the online network.

The amount of FH lost during the check scheduling determines the agent's reward. Equation (29) provides a definition for the reward function:

$$R = \begin{cases} fh_i(z - D), & if\ 0 \le c \le D \\ 10fh_i(D - z), & if\ 0 \le D < c \\ -10^5, & if\ z = -1 \end{cases} \quad (29)$$

where fh_i is the average daily utilization of aircraft i in FH, c is the check's start day, and D is its due date. Three conditions that correspond to three agent penalty levels are present in the function. The FH lost when the check is scheduled without using the tolerance is the condition that corresponds to the smallest penalty, or the first condition. To penalize the agent more for using the tolerance, the second one is equal to the amount of FH that is used multiplied by ten. The last condition, which carries the highest penalty, simulates a situation in which the check was not planned.

Grid search methodology was employed to determine the neural network architecture and hyperparameters for the Deep Q-learning algorithm. Various values were systematically defined and tested individually for each parameter to identify the optimal configuration. The selected network architecture consists of a multilayer perceptron with two fully connected hidden layers, each comprising 200 neurons. The chosen optimizer is the Adam optimizer, and the sigmoid activation function is utilized for both hidden layers [11]. Details of the remaining hyperparameters can be found in Table 1.

Table 1. Hyperparameters for the Deep Q-learning algorithm

Hyperparameter	Value	Description
Episodes	200	Total number of training episodes
Max steps	10000	Maximum number of agent steps per episode
Replay memory size	100000	Size of the queue containing the agent experience
Batch size	032	Number of samples taken from replay memory
Discount factor	0.99	Discount factor of future rewards
Learning rate	0.0001	Learning rate used by the optimizer
Initial ϵ	1	Initial value for exploration
Final ϵ	0.01	Final value for exploration
Target update	10000	Step frequency to update the target network

4.2. Check Cases and Computational Experiments

With 45 aircraft from the Airbus A320 series (A320 and A321), the test case reflects an actual maintenance scenario. Table 2 shows the A/C-check interval and tolerances that are common to all aircraft.

Table 2. A/C-check interval and tolerance for the Airbus A320 and A321.

	Calendar Days	Flight Hours	Flight Cycles
A-check interval	40	250	250
A-check tolerance	4	25	25
C-check interval	730	7500	5000
C-check tolerance	60	500	250

The airline specifies that the duration of C-checks for each aircraft is 15 calendar days, while A-checks are consistently valid for one calendar day. Regarding the minimum interval between the start dates of two consecutive checks of the same type, denoted as μ_k, it is set to three days for C-checks and zero days for A-checks. Additionally, estimates for aircraft usage are provided in the dataset, indicating that an aircraft consumes an average of 10.5 flight hours (FH) and undergoes 7 flight cycles (FC) per day.

In addition to the specified maintenance durations for A/C checks, it's noted that during peak business periods, the C-check operations are temporarily suspended. These high-demand periods include the summer season, spanning from June 1 to September 30. Furthermore, data pertaining to the fleet's initial condition, specifically the duration since each aircraft's previous A and C-check, is provided.

The primary objective in this scenario is to identify the optimal long-term scheduling solution for A/C-checks. The timeframe for analysis spans from January 03, 2020, to January 04, 2024. To evaluate the efficacy of our approach, a comparison will be made between the RL solution and the airline's actual operational data.

5. Results and Discussion

The outcomes of test scenarios are shown in this section. The following are a set of Key Performance Indicators (KPIs) that are defined to assess the calibre of the maintenance plan that is generated:

- the total number of A and C-checks that are planned for the full horizon,
- the average FH between two subsequent checks that are A and C,
- the number of A and C-checks are planned based on the tolerance, and
- the number of A-checks merged into C-checks.

Since they can all affect maintenance costs, which airlines try to keep as low as possible, these KPIs are all important to consider when assessing the maintenance plan. The learning process is one of the most crucial components of the RL algorithm for assessing the agent's performance. Consequently, an effective performance indicator is the total of the prizes that the agent has earned at the conclusion of each training session.

The outcomes are contrasted with airline projections for the same time frame. But since the airline only plans A-checks for the following year, the A-check metrics were not available to them. The approaches are contrasted in Table 3.

Table 3. Comparison of the A and C-check scheduling results between the three approaches under several Key Performance Indicators

KPI	Airline	Our Research
Total C-checks	94	91
C-check Average FH	7236	7348.9
Tolerance Events	0	0
Total A-checks	2750	2743
A-check Average FH	230	238.4
Tolerance Events	-	22
Merged A-checks	-	14
Computation Time	≥ 4 days	954s

According to the findings presented in Table 3, the results derived from the RL method demonstrate notable improvements. Specifically, for C-checks, the RL solution has led to an average increase in flight hours (FH) by 1.6%. This enhancement signifies a boost in daily aircraft operations and overall airworthiness of the fleet. Additionally, the total number of C-checks has been reduced by a factor of 3, indicating a substantial decrease in maintenance frequency. Consequently, this reduction in C-check frequency is anticipated to result in significant cost savings for the airline. Therefore, the utilization of the RL approach for managing C-checks yields superior outcomes compared to traditional manual maintenance planning methods.

The results indicate that the average operating flight hours (FH) for A-checks have improved by approximately 3.6% compared to the airline's baseline. Moreover, the total number of A-checks has been reduced by 7 checks. Additionally, 14 A-checks have been effectively merged with C-checks, while ensuring that the average operating flight hours (FH) remain close to the inspection deadline. This demonstrates the successful achievement of the objective to minimize unused flight hours (FH). Furthermore, the optimization of A-check consolidation with C-checks ensures that the average exploitation remains higher than that of the airline, even after consolidation.

From an economic standpoint, considering that the average airline incurs costs ranging from 70 thousand USD to 350 thousand USD for C-checks and between 10 thousand USD and 15 thousand USD for A-checks [1], the aforementioned results offer significant potential for cost savings. Specifically, the improvements achieved through the RL-based maintenance planning approach can lead to estimated cost savings ranging from 280 thousand USD to 1.2 million USD for the A320/A321 family fleet.

Table 4. Amount of money saved after optimization

	Check reduced	Payment amount/check (USD)	Amount of savings (USD)
C-check	3	70,000 - 350,000	210,000 - 1,050,000
A-check	7	10,000 - 15,000	70,000 - 105,000
Total	10		280,000 - 1,155,000

Table 5. Additional revenue after optimization

	Additional operation days	**Revenue /day (USD)**	**Additional revenue (USD)**
C-check	30 - 45	75,000-120,000	2,250,000-5,400,000
A-check	7	75,000-120,000	525,000-840,000
Total	37 - 52	75,000-120,000	2,775,000-6,240,000

Since it takes approximately 10 to 15 days to complete a C-check and 1 day for an A-check, 3 C-checks and 7 A-checks are reduced to the equivalent of approximately 37 to 52 days for the aircraft to be ready for service. exploit. One day of operations generates revenue from 75 thousand USD to 120 thousand USD [1] and having an additional 37 to 52 days to use for operational purposes means the airline will add revenue from 2.8 million to 6.2 million USD thanks to optimizing maintenance schedules.

The RL technique has reduced the time required to obtain the ideal schedule from 4 days to just 954 seconds when compared to VNA's scheduling time for aircraft maintenance plans. Airlines can therefore save a great deal of time and human resources by implementing machine learning techniques to aircraft maintenance scheduling. This will increase job productivity and result in increased revenue and profits for the business.

The distribution of A and C-check usage levels for the entire fleet is shown in Fig 4 to Fig 7 in bet. From the results we can see the distribution level of the maintenance plan when optimized using the method RL tends to focus more on inspection due times. The number of A-checks tested in the range of 250FH when applying the RL method was 94% higher than that of the VNA side. The number of optimized C-checks is 57% higher.

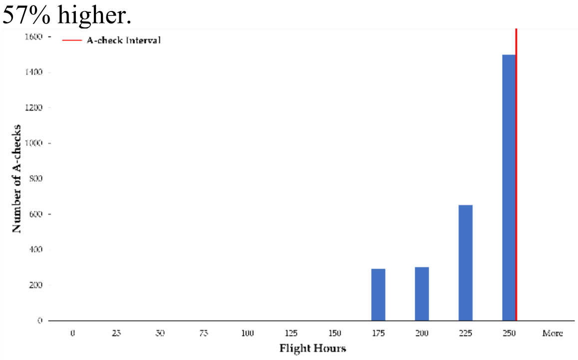

Fig. 4. Distribution of A-check usages of airline

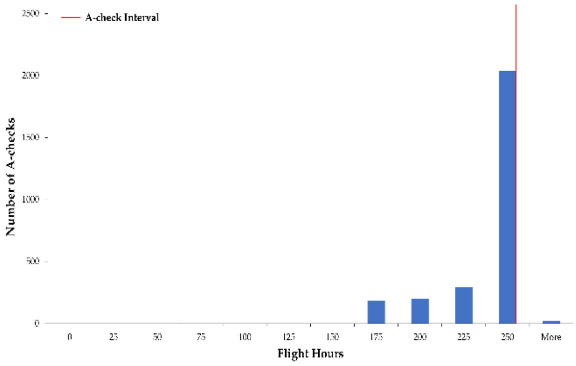

Fig. 5. Distribution of A-check usages optimal

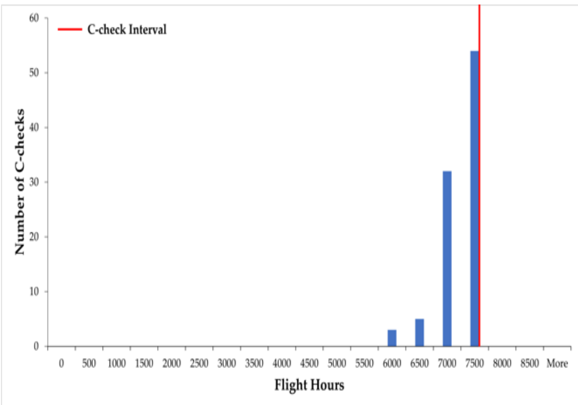

Fig. 6. Distribution of C-check usages of airline

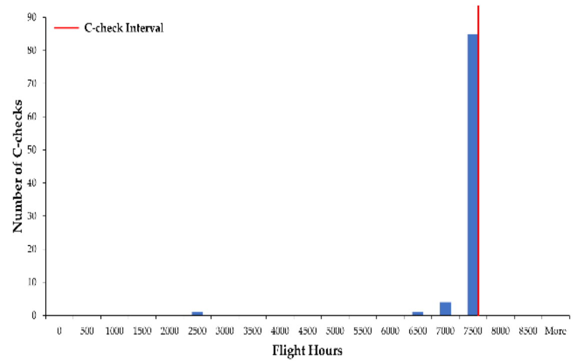

Fig. 7. Distribution of C-check usages optimal

There are some checks that must be done in advance because of the limited number of hangar slots, and the algorithm schedules a C-check several months before the due date due to the peak times we need them. Avoid performing C-checks in the summer, so these checks must be done first to optimize future checks.

From Fig. 4 to Fig. 7, we can see that the application of the RL method has resulted in both A and C-checks approaching the interval, thereby reducing FH loss, and achieving the goal of optimization.

Fig. 8 shows the total reward that the agent receives through each episode. Initially, the agent has just been learned, so there are no optimal strategies to follow, combined with a high discovery rate that leads to low rewards. However, as the agent learns a lot through each episode and has an optimal policy to follow, the reward will increase over time until convergence is reached around episode 155.

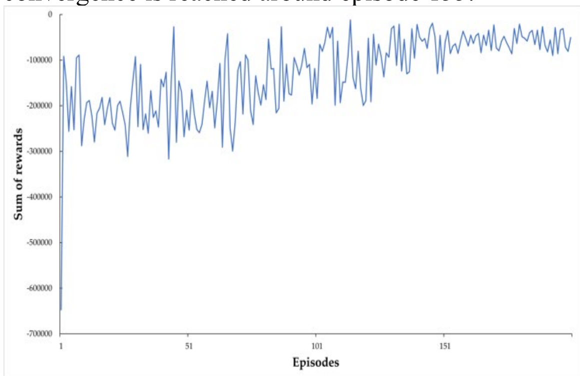

Fig. 8. Total of the agent's training rewards

From the picture, we can see that the reward value increases suddenly in the first episode. This is because when we train the agent to learn, we will schedule C-check first and have a "target model" of C-check. From there, we use the "target model" of C-check to further train the agent when assigned to A-check, to help speed up the learning process of A-check. So, thanks to the previously optimized strategy for C-check, in the first episodes, the reward value increases rapidly. In the following episodes, the agent continues to explore and try new strategies, but the more it tries and learns the optimal arrangement methods, the more the reward value increases, showing that the agent is learning well. Then, as mentioned above, at episode 155, convergence is achieved, which means the agent has found the most optimal strategy, from which the reward value across episodes does not fluctuate a lot.

6. Conclusion

To enhance the long-term scheduling of A and C-checks for a fleet of aircraft, this study introduced a reinforcement learning (RL) approach. The primary objective was to maximize the utilization of flight hours (FH) through optimized maintenance scheduling, thereby increasing aircraft availability. The proposed method utilized a Deep Q-learning algorithm with experience replay. The RL agent was tasked with prioritizing the scheduling of C-checks and selecting an aircraft at each decision stage to schedule its next inspection on the most favorable day available.

With the results and assessments obtained, we see that using the Reinforcement Learning method, in more detail, applying the Deep Q-Learning algorithm to come up with optimal plans for maintenance work. in the engineering industry in general and aviation in particular is increasing. Although still in its early stages, this method has demonstrated great potential in creating optimal maintenance policies and is superior to traditional methods.

Using data from a fleet of 45 aircraft from three sub-fleets (A320, and A321). The RL method yields more efficient maintenance plans as compared to airline estimates for the same period. The airline will benefit financially from reducing the number of unused flight hours (FH) over the entire scheduled period, thus increasing the operating time of 45 of its fleet. In more detail, the method has helped reduce C-check operations by 3 times, increase the average operating flight hours of C-check by 1.6%, reduce A-check operations by 7 times, increase the average operating flight hours of A. -check to 3.6% and merge 14 A-checks into C-check.

Despite its considerable relevance, there aren't many studies on long-term maintenance issues in aviation, which motivates more research on the subject. To further expand the research, other factors affecting aircraft maintenance scheduling, such as resources, tasks to be performed in each type of check, etc., will be researched and added to further optimize the airlines' working processes.

References

[1] S. Ackert, Basics of Aircraft Maintenance Programs for Financiers: Evaluation & Insights of Commercial Aircraft Maintenance, 2010. http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics_of_aircraft_maintenance_programs_for_financiers___v1.pdf

[2] Q. Deng, B. Santos and R. Curran, A practical dynamic programming based methodology for aircraft maintenance check, Eur. J. Oper. Res, p. 256-273, 2019. https://doi.org/10.1016/j.ejor.2019.08.025

[3] P. Horder, Airline operating costs, Presented at Managing Aircraft Maintenance Costs Conference, Brussels, 2003. https://www.dima.univr.it/documenti/Avviso/all/all520253.pdf

[4] IATA, Airline maintenance cost executive commentary, IATA’s Maintenance Cost Task Force, 2015.

[5] A. Steiner, A Heuristic Method for Aircraft Maintenance Scheduling under Various Constraints, Swiss Transport Research Conference at Switzerland, 2006.

[6] R. Sutton and A. Barto, Reinforcement Learning: An Introduction, Cambridge, MA, USA: MIT Press, 2018.

[7] P. Andrade, C. Silva, B. Ribeiro and B. Santos, Aircraft Maintenance Check Scheduling Using Reinforcement Learning, Aerospace, 2021. https://doi.org/10.3390/aerospace8040113

[8] H. van Hasselt, Double Q-Learning, Proceedings of the 24th Annual Conference on Neural Information Processing Systems at Canada, p. 6-9, 2010.

[9] H. van Hasselt, A. Guez and D. Silver, Deep Reinforcement Learning with Double Q-Learning, Proceedings of the 30th AAAI Conference on Artificial Intelligence at USA, p. 12-17, 2016. https://doi.org/10.1609/aaai.v30i1.10295

[10] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Proceedings of the 13th International Conference on Artificial Intelligence and Statistics at Italy, pp. 13-15, 2010.

[11] D. Kingma and J. A. Ba, ADAM: A Method for Stochastic Optimization, ICLR 2015. https://arxiv.org/pdf/1412.6980