Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

AIRCRAFT LINE MAINTENANCE SCHEDULING USING SIMULATION AND REINFORCEMENT LEARNING

Simon Widmer
Syed Shaukat
Cheng-Lung Wu

School of Aviation
UNSW Sydney
Kensington, NSW 2052, AUSTRALIA

ABSTRACT

This paper presents a reinforcement learning (RL) algorithm prototype to solve the aircraft line maintenance scheduling problem. The Line Maintenance Scheduling Problem (LMSP) is concerned with scheduling a set of maintenance tasks during an aircraft's ground time. To address this problem, we introduce a novel LMSP method combining a hybrid simulation model and reinforcement learning to schedule maintenance tasks at multiple airports. Initially, this paper briefly reviews the existing literature on optimization-based and AI-enhanced aircraft maintenance scheduling. Secondly, the novel reinforcement learning LMSP method is introduced, evaluated using industry data, and compared with optimization-based LMSP solutions. Our experiments demonstrate that the LMSP method using reinforcement learning is capable of identifying near-optimal policies for scheduling line maintenance jobs when compared to the exact and heuristics-based methods. The proposed model provides an excellent foundation for future studies on AI-enhanced scheduling problems.

1 INTRODUCTION

Aircraft maintenance challenges airlines in terms of costs and planning. According to IATA (2021), aircraft maintenance expenses represent around 11.2% of the costs associated with the operation of fleets of a typical airline. The global maintenance, repair, and overhaul (MRO) spending in 2022 amounted to \$78.5 billion. Due to shifts in airline fleet configurations following the COVID-19 pandemic, coupled with an increased focus on achieving long-term sustainability goals and supply chain complexities within the aviation industry, it is expected that the global demand for MRO services will surge to \$126.6 billion by 2032 (IATA 2021; OliverWyman 2022). Therefore, it is of high interest for airlines to explore ways to reduce maintenance costs while guaranteeing the safety of their passengers and the airworthiness of their fleet.

Airline scheduling is a complex task that includes a multitude of factors, such as the coordination of thousands of daily flights, management of hundreds of aircraft, crew flight time limitations and fatigue, air traffic control, and airport operational curfews (Barnhart and Cohn 2004; Belobaba et al. 2009). Traditionally, the airline planning process is divided into four sequential sub-problems as shown by Bazargan (2016): schedule design problem (SDP), fleet assignment problem (FAP), aircraft routing problem (ARP), and crew scheduling problem (CSP). Historically, these airline schedule sub-problems have been tackled sequentially, due to the high complexity within the four planning phases. Various efforts have been made toward achieving a more integrated approach to planning airline schedules. However, Cadarso and de Celis (2017) emphasize that it has been infeasible so far to produce a fully functional and optimal schedule using an approach that integrates the four sub-problems and finds the trade-off for critical aspects of airline operations, including unpredictable operational constraints and stochastic demand.

Within the airline planning process, aircraft maintenance influences the phases of fleet assignment and aircraft routing. While an airline's maintenance and engineering capabilities may influence the fleet choice for operating a route, aircraft maintenance is primarily incorporated in the aircraft routing problem (ARP) which specifies which specific aircraft to fly each flight segment assigned to the fleet (Belobaba et al. 2009; Zhou et al. 2020). Solving the aircraft routing problem includes assessing the feasibility of the routing plan to satisfy both operational and maintenance criteria. Commercial aircraft are subject to regular maintenance checks mandated by aircraft manufacturers and air transportation regulatory bodies. These checks are triggered on various metrics, including the number of performed flight hours, take-offs and landings (cycles), and elapsed time since the previous maintenance check. Adherence to safety and maintenance standards is strictly enforced and aircraft failing to meet minimum requirements are subject to grounding, as outlined by Barnhart and Smith (2012).

Line maintenance refers to minor maintenance activities performed on aircraft during the turn-around time between two flights to ensure aircraft airworthiness. This may include inspections, minor repairs, and modifications (Gupta et al. 2003). Real-time data collected through sensors on aircraft has advanced significantly, allowing businesses to make data-driven decisions and optimize maintenance strategies (Černỳ 2019). Airlines can now adopt a condition-based maintenance approach based on the aircraft's health status and operational history, which is more efficient and cost-effective than predetermined maintenance schedules. As a result, task-driven maintenance jobs can often be executed during line maintenance, reducing the need for cyclic letter checks (Ruther et al. 2017).

The literature on the LMSP topic is scarce. Senturk and Ozkol (2018) underline that aircraft utilization can be improved if individual jobs can be executed when an aircraft is on the ground, compared to the traditional scheduling of periodic checks. This can improve capacity utilization in line maintenance and balance the workload at maintenance bases, as highlighted by Hughes (2006). However, Senturk and Ozkol (2018) do not provide a mathematical formulation of the underlying problem, and current LMSP solutions in the literature are limited to simple heuristics.

Aircraft maintenance optimization strategies using reinforcement learning are proposed in Hu et al. (2021) and Andrade et al. (2021). These studies highlight the benefits of using reinforcement learning in aircraft maintenance planning, resulting in maintenance schedules that: maximize the flight hour usage of aircraft between checks, reduce the number of scheduled checks throughout an aircraft's lifespan, and enable airlines to generate more revenue while reducing maintenance expenses. However, none of these studies includes the planning of short-term line maintenance jobs.

In the context of a dynamic operating environment characterized by uncertainties and disruptions, heuristics have proven to be an effective alternative to an optimization approach for solving the LMSP efficiently. Previous research conducted by Shaukat et al. (2020) has demonstrated the effectiveness of scheduling heuristics, which have yielded highly satisfactory scheduling results. In this paper, we build on the work of Shaukat et al. (2020) and adopt an RL-enhanced scheduling method to solve the aircraft line maintenance scheduling problem (LMSP).

This paper is organized, as follows: Section 2 provides relevant background on optimization-based and RL-enhanced scheduling methods. Section 3 presents the research data and relevant model constraints including the model implementation approach. Section 4 describes the experimental setup including calibration, verification, and experimental results of the developed model. Section 5 summarizes the results and addresses directions for future research on this topic.

2 LITERATURE REVIEW

2.1 Optimization

The airline industry has a long history of using optimization methods and operations research to solve airline planning problems (Barnhart et al. 2003). A linear integer programming approach to solve the airline fleet scheduling and aircraft routing problem was introduced by Levin (1971). The study applied exact optimization methods incorporating the branch and bound algorithm to find optimal solutions to real-world problems. Research conducted by Feo and Bard (1989) applied a column generation method to generate a flight schedule that met the cyclic constraints of aircraft maintenance requirements. Gopalan and Talluri (1998) developed a fast and simple method using a polynomial-time algorithm that solves the aircraft routing problem. The model satisfied both the three-day maintenance (A-Check) and balance-check (quarter C-checks) requirements. The algorithm was designed for a dynamic finite-horizon model and a static infinite-horizon model. Consequently, realistic airline maintenance constraints were included such as only performing maintenance tasks during night shifts. Sriram and Haghani (2003) introduced a new formulation for aircraft re-assignment and maintenance scheduling by applying a hybrid heuristic method to solve large-scale integer programming problems. The heuristic combined random search and depth-first search which produced results within a 5% gap of the global optimal solutions. The heterogeneity of the fleet and several cyclic constraints made the problem more complex compared to traditional flight scheduling and crew scheduling problems. Although limited in the scope of domestic services and planning A- and B-checks only, the study is seen as an extension to the research by Feo and Bard (1989).

Papakostas et al. (2010) introduced an operational optimisation strategy for scheduling line maintenance as a decision-support tool (DSS). The framework accurately modeled an airline's maintenance policy using criteria weights and analyzed several operational factors such as aircraft health data, maintenance records, available resources, and maintenance costs. The DSS produced a set of feasible maintenance plan alternatives for the decision maker based on objectives such as cost, flight delays, and remaining useful life. This work aimed to tackle the inherently reactive approach in airline operations and focused on resolving unscheduled maintenance activities. Another real-time decision tool that incorporated data from dynamic work progress monitoring was presented by Callewaert et al. (2018). The tool generated alternative plans using heuristics when one or more maintenance tasks deviated from their due dates. According to the findings of the case study, the framework optimized maintenance planning by decreasing maintenance costs from 45 to 90%.

Lagos et al. (2020) introduced a dynamic take on the LMSP by integrating the aircraft maintenance scheduling problem with tail assignments. The model was based on a dynamic Markov Decision Process (MDP) which selects a daily set of aircraft and jobs to be performed and aims to minimize the expected costs of expired maintenance jobs in the long term. The framework scheduled tasks to consume fewer resources in order to utilize the available maintenance capacity. The model further allowed for flexible processing times by adapting resource consumption rates with increased flexibility in scheduling. In addition, the framework took advantage of outsourcing or sharing maintenance resources among multiple airlines and therefore, considered factors enabled by the economies of scale. This approach was, however, simplified to a single maintenance base.

Deng et al. (2020) presented a dynamic programming (DP) approach to optimize the long-term maintenance check schedule for a heterogeneous aircraft fleet by integrating multiple check types into a single schedule solution. The DP-based method aimed to maximize aircraft availability by reducing the number of maintenance checks by approximately 7% over four years. From a financial perspective, this method may save the industry partner between \$1.1M to \$3.4M in maintenance cost, while allowing to generate \$9.8M additional revenue due to increased aircraft availability. However, the study did not take into account the potential for disruptions in aircraft usage or unexpected non-routine work that could affect the duration of maintenance checks. Witteman et al. (2021) proposed a new approach to solve the task allocation problem (TAP) in aircraft maintenance scheduling. The TAP assigned maintenance tasks to maintenance opportunities which is a complex combinatorial problem that maintenance operators need to solve daily. The proposed approach was built on top of the DP approach for maintenance scheduling introduced by Deng et al. (2020). The problem was defined as a time-constrained variable-sized bin packing problem (TC-VS-BPP) and applied a worst-fit decreasing algorithm to solve it efficiently. The approach was tested and validated with maintenance data sourced earlier from a project partner airline. For a heterogeneous fleet of 45 aircraft, the method was found to be more than 30% faster than the previous approach, with an optimality gap below 3% in most cases.

A DP-based scheduling method and task allocation approach was integrated into a decision support system (DSS) by Deng et al. (2021). The solution allowed maintenance planners to initially plan an optimized long-term aircraft maintenance check schedule. This was followed by allocating specific maintenance tasks to each maintenance check with the help of the developed task allocation method. Lastly, the DSS allowed efficient shift planning according to the allocated task sequence. The presented DSS provided a solution that combined the traditionally separated problem-solving of maintenance check planning, task allocation, and workforce planning. As a result, it offered significant advantages for both scientific research and industry application.

2.2 RL-based Scheduling Methods

Reinforcement learning (RL) involves an agent engaging with its surroundings and utilizing trial and error to learn an optimal policy for taking sequential decisions. The neural network structure of the RL model is designed to employ rewards to train it, which involves taking different actions in a specific environment following guided policies. The Markov-Decision Process (MDP) governs the decision-making process of an RL model, as each action may result in different outcomes in the study environment, further referred to as rewards. Therefore, training an RL model requires a balance between exploiting current knowledge and exploring uncharted territory to discover the best policy for taking actions that yield maximum benefits (Sutton and Barto 2018).

Andrade et al. (2021) incorporated a reinforcement learning approach on top of the dynamic programming approach for long-term maintenance check scheduling previously presented by Deng et al. (2020). The goal was to examine whether a reinforcement learning approach can maximize the flight hour usage of aircraft between scheduled A-, C- and D-checks. The total number of scheduled maintenance checks proposed by the RL-method was compared with the scheduling results achieved using the DP approach. The reinforcement learning method applied the deep Q-learning algorithm with a double Q-learning variant, and achieved better results in scheduling C-checks.

Hu et al. (2021) developed a novel maintenance planning strategy incorporating reinforcement learning with an Extreme Learning Machine (ELM)-based Q-learning algorithm. Traditional aircraft maintenance strategies included scheduled, corrective, prognostic, and health management methods. The applied ELM-Q-learning algorithm considered the uncertainties and complexities of the current operational environment and adapted current flight schedules based on aircraft health, failure rates, repair costs, and mission needs. Therefore, the ELM method provided more dynamic decision support centered around the current state of the maintenance environment, which the authors proposed to potentially replace traditional maintenance strategies. However, the method developed by Hu et al. (2021) was tailored to one aircraft only. Therefore, it would be beneficial to scale the model for multiple aircraft of a heterogeneous fleet and address challenges related to flight scheduling, fleet assignment, crew assignment, and maintenance routing problems.

In contrast to Hu et al. (2021) and Andrade et al. (2021), Lee et al. (2022) presented a RL framework using Q-learning and Double Q-learning algorithms to tackle the aircraft recovery problem during operational disruptions. The RL approach allows for the re-timing of specific flights and swapping of aircraft without duplicating flight arcs and can flexibly adapt to various objectives by modifying reward functions. The RL method was tested using a South Korean airline's domestic flight schedule and delivered promising results, outperforming other methods in computation time. However, the method is prone to cause unnecessary operational disturbance and higher non-operational costs due to increased crew duty swaps.

Reinforcement learning is widely applied, including in board games. TD-Gammon, developed by Gerald Tesauro, achieved a near-grandmaster level in backgammon with minimal prior knowledge (Tesauro et al. 1995). AlphaGo and its successor, AlphaGo Zero, demonstrated the power of reinforcement learning by mastering the game of Go and surpassing human performance. While AlphaGo utilized reinforcement and supervised learning from expert human moves, AlphaGo Zero solely relied on reinforcement learning without human data or guidance beyond the game rules (Silver et al. 2017).

Robotics is a critical platform to adopt new technologies for the development of AI solutions and a typical application of reinforcement learning (Li, Yuxi 2018). Kober et al. (2013) describes several applications such as learning helicopter flight, driving a radio-controlled vehicle, and learning a jumping behavior for a robot dog. Moreover, RL is a crucial part of the development of natural language processing. OpenAI's large language model (LLM) ChatGPT was optimized for dialogue using RL with Human Feedback (RLHF). This method involves using human demonstrations and preference comparisons as supervised fine-tuning to guide the model toward desired behavior, as outlined by OpenAI (2023).

RL has found success in the field of intelligent transportation. Sallab et al. (2016) used deep reinforcement learning (DRL) to create lane-keeping assistance methods for autonomous driving, training the policy using raw sensory data as input. Applying DRL to autonomous driving poses challenges such as addressing the gap between simulation and reality for accurate agent training and integrating safety considerations into RL systems for autonomous agents (Kiran et al. 2022).

3 SCHEDULING ALGORITHMS AND CASE STUDY DATA

3.1 Line Maintenance Scheduling Algorithm

The Line Maintenance Scheduling Problem (LMSP) involves decision-making for two types of scheduling problems: job assignments and timetable design. For job assignments, the LMSP assigns maintenance jobs to maintenance opportunities provided by aircraft schedules. The job assignment problem aims to allocate maintenance jobs to available maintenance opportunities with the objective of minimizing the total deviation from the ideal due time for each job. The timetabling problem defines the start time for each job within its designated maintenance time window. Maintenance jobs have due times which indicate the ideal service time for maintenance jobs to be finished. As resource capacity at airports is limited, the timetabling decisions must ensure that the limited resource capacity at each airport is not exceeded. A more detailed description of the model constraints including a mathematical formulation of the LMSP as a mixed-integer problem is available in Shaukat et al. (2020).

It is noted that the timetabling algorithm in this paper uses a bin structure allowing micro-timetabling of maintenance jobs to ensure resource capacity is not exceeded within predetermined time intervals for each maintenance opportunity. In this paper, the bin structure interval is designed by ten bins per hour, i.e. 6 mins for each bin. The micro-timetabling structure aims to set the starting time of a maintenance job as early as feasible within its allocated time slot. From this perspective, the optimisation formulation in Shaukat et al. (2020) was more restricted in terms of job timetabling than the newly implemented bin structure. We will further discuss the difference in the Discussion section of the paper.

The LMSP heuristic was created to replicate the manual procedure used by maintenance schedulers of a study airline. This procedure involves prioritizing jobs based on deadlines and importance, allocating them in a specific order, and estimating resource capacity to create feasible solutions. The first phase of the process produces an initial solution, and the second phase involves manual fine-tuning to ensure resource availability and accommodate unscheduled jobs. However, the manual procedure is not always guaranteed to find a feasible solution. The LMSP heuristic aims to optimize this process and increase efficiency.

The LMSP algorithm involves two main stages:

1. Job assignment stage, which prioritizes mandatory jobs and assigns them to maintenance opportunities in a greedy manner based on their deadlines, while considering capacity constraints.

2. Timetabling stage, which takes the assigned jobs and searches for a feasible timetable considering capacity consumption within determined time intervals (e.g., within every 6min), at every airport or maintenance base, by detecting resource violations and attempting to reschedule the removed job to restore feasibility. The algorithm tries to reschedule the job first within its current maintenance opportunity at different starting times and then in its closest maintenance opportunity.

The LMSP objective function is defined as a weighted sum of earliness and tardiness of mandatory scheduled jobs, plus the cost of jobs left unscheduled as outlined by Shaukat et al. (2020). A job is recorded as unassigned if no maintenance opportunity subject to the above schedule feasibility criteria can be found. In this case, an unassigned job is matched to a fictional maintenance opportunity designated to hold unassigned jobs.

$$\min \sum_{j\in J} m_j(e_j y_j + t_j z_j) + \sum_{j\in J} c_j\Big(1 - \sum_{m\in M} x_{jm}\Big) \tag{1}$$

where,

e_j: cost per unit of earliness for of job j.
y_j: earliness of job j.
t_j: cost per unit of tardiness of job j.
z_j: tardiness of job j.
m_j: binary, indicates whether job j is mandatory or non-mandatory; $m_j = 1$ defines that job j is mandatory and must be scheduled.
c_j: cost for leaving job j unscheduled.
x_{jm}: binary, indicates if job j is executed during maintenance opportunity $m \in M$. $x_{jm} = 1$ indicates that job j is scheduled.

3.2 Research Dataset

The line maintenance job assignment experiments are carried out using a dataset collected during a previous study from a partner airline. The dataset consists of comprehensive information about flight schedules and the corresponding maintenance tasks for a fleet of four aircraft. The airline operates four maintenance bases, each with a defined maintenance personnel resource capacity (person-hours) available across three maintenance trade categories: airframe, engine, and avionics. Job deadlines and due dates are recorded in three units: flight hours, number of landing cycles, and calendar days. Due to operational considerations, only maintenance opportunities with a minimum ground time of 30 minutes or more were considered.

3.3 Implementation

The LMSP heuristic, including the schedule feasibility constraints as described in section 3.1 is implemented as a hybrid simulation model using the simulation software AnyLogic. The simulation environment replicates the scheduling system of an airline and is implemented in the form of a deterministic Markov Decision Process (DMDP) which provides a forecast schedule, although it does not incorporate factors of uncertainty. The simulation model is connected to the Microsoft Project Bonsai AI platform, which enables the training of RL agents using deep reinforcement learning.

A system overview is illustrated in Figure 1, which describes the interaction between the simulation model and the RL environment. As introduced in Section 3.2, the simulation model uses a deterministic set of maintenance tasks and opportunities as input. The model generates schedule performance metrics as outputs, which are then used as states in the RL environment. These states include the current sorting policy for jobs and opportunities, the count of unscheduled jobs, and the resulting objective function value.

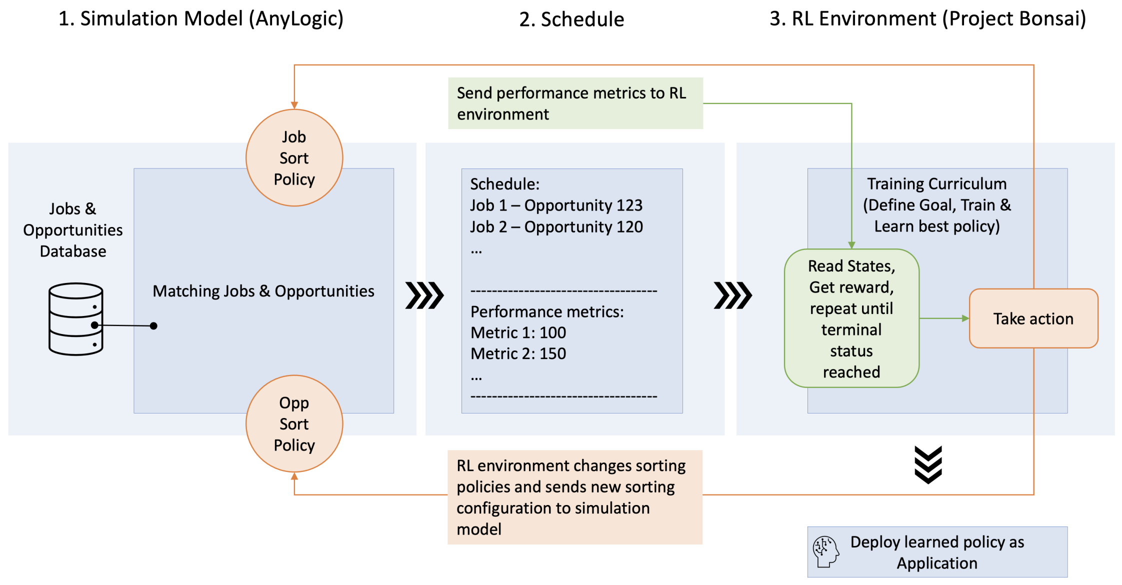
1. Simulation Model (AnyLogic)
2. Schedule
3. RL Environment (Project Bonsai)
Send performance metrics to RL environment
Job Sort Policy
Jobs & Opportunities Database
Matching Jobs & Opportunities
Opp Sort Policy
Schedule:
Job 1 – Opportunity 123
Job 2 – Opportunity 120
...
Performance metrics:
Metric 1: 100
Metric 2: 150
...
Training Curriculum (Define Goal, Train & Learn best policy)
Read States, Get reward, repeat until terminal status reached
Take action
RL environment changes sorting policies and sends new sorting configuration to simulation model
Deploy learned policy as Application

Figure 1: The interaction between the simulation model and the RL training environment.

The RL environment includes a training curriculum specifying the learning objective of the RL process. According to the defined learning curriculum, the RL environment calculates a reward value based on the received policy (state) from the simulation model and takes action by selecting a new policy configuration.

This new scheduling policy is sent to the simulation model, and this concludes one state-action iteration. This process is repeated until the RL agent is fully trained according to the training objective defined in the learning curriculum. The objective is evaluated using the goal satisfaction performance metric, which records the average progress in the percentage of training iterations satisfying the objective, compared to the total number of training iterations. A goal satisfaction of 100% indicates that the trained RL 'brain' has successfully completed the objective defined in the learning curriculum by reaching a terminal condition. Bonsai allows the implementation of the RL objective by using custom reward and terminal functions or through a low-code goal statement such as minimizing, maximizing, or avoiding a target value. In this project, a one-step RL approach with a low-code objective is implemented to learn a scheduling policy that minimizes the objective function value associated with the schedule solution.

The implemented RL environment aims to learn the best scheduling policy with the minimum objective function value from a potential combination of 18 first-level job sorting policies and 22 first-level opportunity sorting policies. The job sorting policies involve 9 job sorting priorities in both ascending and descending order, such as job duration, deadline, or resource requirement: 9 x 2 = 18 job sorting policies. Similarly, the maintenance opportunity sorting policies consist of 11 opportunity sorting priorities in ascending and descending order: 11 x 2 = 22 opportunity sorting policies. In example, this includes sorting opportunities by starting time, duration, or deadline. In total, there are 396 (18 x 22) possible policy choices applying first-level sorting, which defines the discrete action space for the RL training environment. Two example policies available to the RL training algorithm are provided as follows:

- Policy A: Sort jobs ascending to job deadline and match to opportunities sorted by ascending opportunity starting time.
- Policy B: Sort jobs descending to job duration and match to opportunities sorted by descending aircraft use measured in flight hours.

The simulation model's schedule output depends on the chosen sorting policy that pre-processes the simulation model's input data. Once the RL algorithm is fully trained using the Microsoft Project Bonsai platform, the trained policy is referred to as a trained "Brain" and may be deployed as a web application. A deployed Brain is then available to schedule line maintenance jobs by applying the learned scheduling policy.

4 MODEL CALIBRATION, VERIFICATION AND EXPERIMENTAL RESULTS

The performance of the RL-enhanced LMSP model is compared to an optimization-based method (ILMSP) and a heuristic-based method (SLMSP) developed in a previous study by Shaukat et al. (2020). The methods are compared using a test set of 29 schedule variations which differ by the number of aircraft (A), the length of the planning horizon (L), and the total number of jobs to be scheduled (J) as available in Table 2.

The available resource capacity at any of the four maintenance bases is three person-hours for airframe (AF), two for engine (EN), and one for avionics (AV). Due times of maintenance jobs are set to 15 hours, 10 cycles or 1 day before the deadline, according to the relevant unit for the job (flying hours, landing cycles or calendar days). The following weighting factors are applied to both earliness and tardiness in the objective function (1): weight factor 1 for flying hours, 2.5 for cycles, and 10 for calendar days. To encourage the scheduling of all jobs during the planning horizon, a large weight factor $c_i = 50$ is used to account for any unscheduled jobs.

Two RL-scheduling brains were trained using two test instances from Table 2, differing in the number of maintenance jobs and the number of aircraft. The Brain 148J was trained based on the test instance (4/4/148) and the Brain 329J was trained on the test instance (9/4/329). Both RL methods were defined by a training curriculum using a goal statement to minimize the objective function value as much as possible within 200,000 iterations. As available in Table 1, both RL-trained methods were fully trained within 1h 30min to 3h 36min, achieving near-optimal goal satisfaction between 98.8% to 100% in selecting the best scheduling policy that minimizes the schedule objective function value at 98,000 training iterations.

Table 1: RL-Brain training metrics.

No. of iterations	Brain 148J		Brain 329J	
	Duration (min)	Goal Satisfaction	Duration (min)	Goal Satisfaction
400	1	81.7%	1	66.2%
48,000	40	81.7%	71	86.2%
73,000	64	97.8%	177	100%
98,000	90	98.8%	281	100%

As shown in Table 2, both RL-trained methods generate feasible schedules with lower objective function values and shorter runtimes for every test scenario in comparison to the exact ILMSP algorithm. As illustrated in Figure 2, the objective function value for Brain 148J and Brain 329J is on average 20% lower compared to the schedule results obtained by the ILMSP method, and 23% lower compared to the SLMSP algorithm. The RL method is based on the approximate SLMSP approach. Nevertheless, both trained RL models scheduled all mandatory and non-mandatory jobs to maintenance opportunities in line with the exact ILMSP algorithm. However, the exact ILMSP algorithm may produce different job assignments compared to approximate methods. The exact method's objective to avoid penalties for unscheduled non-mandatory jobs can lead to the displacement of mandatory jobs from their optimal maintenance slots. This caused the difference in the schedule's objective function value. For instance, in Table 3, the exact method assigns the mandatory job (T00141) to opportunity 327, resulting in a spread value of 1 and an objective function (O.F.) value of 2.5. In contrast, the RL method schedules the job closer to its due time using a neighboring

Table 2: Comparative computational results from ILMSP, SLMSP, Brain 148J, and Brain 329J method.

(A/L/J)	ILMSP O.F.	Time (s)	SLMSP O.F.	Gap	Time (s)	Brain 148J O.F.	Gap	Time (s)	Brain 329J O.F.	Gap	Time (s)
(3/2/57)	66.0	6	66.0	0.0%	3	53.0	-19.7%	3	53.0	-19.7%	2
(3/3/82)	92.0	13	98.0	6.5%	4	77.5	-15.8%	7	77.5	-15.8%	6
(3/4/112)	148.0	5	148.0	0.0%	5	109.5	-26.0%	5	109.5	-26.0%	5
(4/2/77)	58.0	14	58.0	0.0%	3	41.5	-28.4%	4	41.5	-28.4%	3
(4/3/112)	150.0	8	157.0	4.7%	4	122.0	-18.7%	6	122.0	-18.7%	6
(4/4/148)	155.0	10	155.0	0.0%	7	123.0	-20.6%	8	123.0	-20.6%	8
(4/3/157)	211.5	91	214.5	1.4%	7	171.5	-18.9%	8	171.5	-18.9%	8
(4/4/157)	204.5	10	204.5	0.0%	7	169.5	-17.1%	9	169.5	-17.1%	9
(5/2/94)	124.5	5	134.5	8.0%	2	111.0	-10.8%	4	109.0	-12.4%	4
(5/3/150)	193.0	10	197.0	2.1%	5	160.0	-17.1%	8	159.0	-17.6%	8
(5/3/187)	222.5	90	235.5	5.8%	9	166.5	-25.2%	10	166.5	-25.2%	11
(5/4/189)	305.5	42	307.0	0.5%	6	270.5	-11.5%	14	266.0	-12.9%	13
(5/4/202)	194.5	92	195.5	0.5%	10	128.0	-34.2%	16	131.5	-32.4%	15
(6/4/221)	286.0	400	286.0	0.0%	11	231.0	-19.2%	18	231.0	-19.2%	18
(6/3/229)	247.0	209	254.0	2.8%	11	214.0	-13.4%	13	206.5	-16.4%	12
(7/3/243)	309.5	236	309.5	0.0%	11	246.5	-20.4%	14	246.5	-20.4%	15
(7/4/263)	299.0	892	303.5	1.5%	14	232.0	-22.4%	24	231.5	-22.6%	24
(8/3/285)	357.0	685	362.0	1.4%	12	278.5	-22.0%	17	278.5	-22.0%	18
(8/4/298)	369.0	124	385.0	4.3%	15	313.5	-15.0%	33	311.0	-15.7%	34
(9/4/329)	388.0	938	428.5	10.4%	16	298.0	-23.2%	40	296.0	-23.7%	40
(9/3/348)	464.0	1230	477.0	2.8%	16	389.5	-16.1%	25	389.5	-16.1%	23
(10/4/372)	482.0	781	516.0	7.1%	17	391.0	-18.9%	49	394.0	-18.3%	48
(10/3/382)	490.5	3600	520.0	6.0%	18	394.5	-19.6%	32	396.5	-19.2%	31
(11/3/415)	611.0	1791	681.5	11.5%	18	460.5	-24.6%	39	450.0	-26.4%	40
(11/4/424)	434.5	1223	447.5	3.0%	20	337.0	-22.4%	46	339.0	-22.0%	46
(12/3/211)	238.0	123	242.5	1.9%	10	206.0	-13.4%	39	200.5	-15.8%	38
(12/4/337)	449.0	329	475.5	5.9%	13	355.5	-20.8%	64	357.0	-20.5%	62
(13/3/370)	469.0	743	509.5	8.6%	16	390.5	-16.7%	47	385.0	-17.9%	48
(13/4/467)	532.0	1822	554.5	4.2%	22	467.5	-12.1%	79	463.0	-13.0%	78
Average	294.8	535.2	307.7	3.5%	10.8	238.2	-19.5%	24.1	237.1	-19.8%	23.8

opportunity 328, resulting in a spread and O.F. value of 0. Both methods adhere to the available airport resource capacity.

Table 3: Comparison of assignment for mandatory Job T00141 (deadline type: Flight Cycle) in schedule instance (4/4/148): exact ILMSP method vs. RL

Method	Job Id	Opportunity Id	Spread	O.F.
ILMSP	T00141	327	1	2.5
RL	T00141	328	0	0

The runtime to produce feasible solutions vary significantly between the optimization-based method (ILMSP) and the AI-enhanced LMSP method. On average, the ILMSP method produces schedules in 535 sec, while a trained brain solves the LMSP on average within 24 sec. The heuristics-based approach of SLMSP is the most efficient method in terms of model runtime by producing results average within 10.4 sec.

The runtime and objective function values of all scheduling methods increase as the assignment problem becomes larger and more complex with an increasing number of jobs, aircraft, and an extended planning horizon. Nevertheless, the model runtime of the trained brains increases with a lower rate for more complex schedule instances as compared to the exact ILMSP method as illustrated in Figure 3. While the trained RL methods significantly reduce the calculation time for providing a new feasible schedule solution in comparison to the exact ILMSP method, it is important to note that both brains require significant training time of up to 3 and a half hours before being deployed to solve problems. Both trained brains generate similar objective function gaps in comparison to the ILMSP method within 1.1% on average while the model runtime differs up to 0.3 sec between the two brains, which is considered as insignificant.

In terms of model runtime, the heuristics-based approach of the SLMSP performs especially efficiently in complex test instances of more than 200 jobs. In schedule test instances of up to 200 jobs, the SLMSP method records on average 25% less model runtime (avg. runtime $\leq$200 jobs: SLMSP = 5.2s; Brain 148J = 7.2s; Brain 329J = 6.9s). However, in complex test instances of more than 200 jobs, feasible solutions are produced by the SLMSP algorithm in half the calculation time compared to a trained brain (avg. runtime $\geq$ 200 jobs: SLMSP = 14.7s; Brain 148J = 35.0s; Brain 329J = 34.7s).

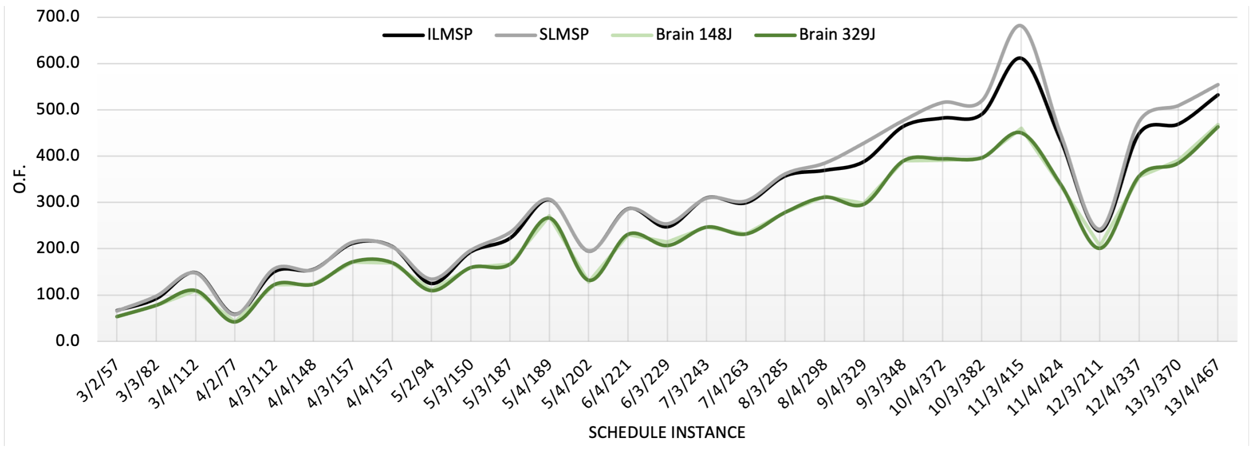

Figure 2: Objective function value comparison for increasing schedule instance complexity.

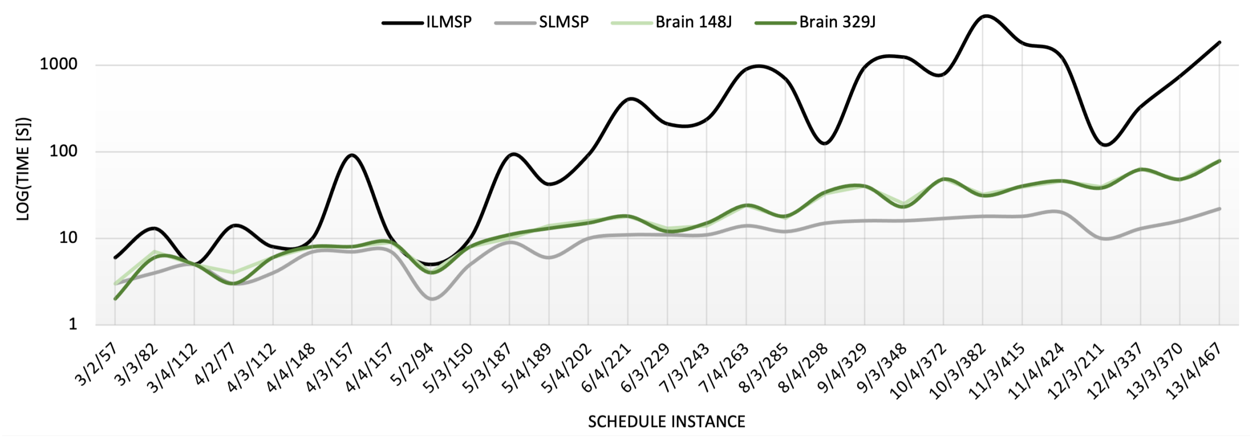

Figure 3: Model runtime comparison for increasing schedule instance complexity.

5 RESULTS AND DISCUSSION

This paper presents an AI-enhanced scheduling method that leverages RL to solve the Line Maintenance Scheduling Problem (LMSP). Two RL-enhanced scheduling solutions were trained, and their performance was compared to exact and heuristic methods developed in a previous study. The experimental results demonstrated that the proposed AI-enhanced scheduling solutions achieved near-optimal schedules for all test instances, reducing the objective function value by 20% on average compared to the exact ILMSP and approximate SLMSP methods. The RL method in this study differs from the exact ILMSP algorithm in two ways. Firstly, it is an approximate approach that aligns with real-world airline practices. Secondly, it includes an adapted timetabling functionality. In contrast, the ILMSP guarantees an exact and feasible solution. We observed that the exact method overemphasizes on scheduling non-mandatory jobs driven by a high penalty cost associated with unscheduled non-mandatory jobs. Table 3 demonstrates that the focus towards achieving an exact solution can lead to the displacement of mandatory jobs from their optimal maintenance slots, thereby affecting the objective function value. In this regard, the approximate RL methods offer a more practical scheduling result from a practical standpoint.

The RL solutions generate feasible schedules in under 24 sec on average, whereas the exact ILMSP method takes approximately 10 min. An airline typically finds a solution time of up to 20 min acceptable for a planning problem. The shorter runtime of the RL method benefits an airline by facilitating rapid adaptation to changing conditions and exploration of multiple scenarios. In contrast to the exact method, the RL models allow for iterative and experimental iterations within the same timeframe, providing added value that the exact ILMSP solution lacks. However, it is noted that the RL-enhanced scheduling solutions require significant training time of up to 3 h 36 min before deployment. This training time can be further reduced by using more cloud computing resources. Further, the scheduling heuristic requires accurate representation by a simulation environment which is a time-consuming development process.

Although the developed RL-brains were trained by specific schedule instances, the deployed LMSP methods demonstrated near-optimal scheduling results across a set of different test instances with varying sizes. A limitation of RL-enhanced methods is identified in complex test instances of more than 200 jobs, in which the heuristics-based method significantly outperforms the trained brains providing feasible solutions in half the calculation time. Future studies could explore integrating multiple schedule variations in the training process of RL-enhanced LMSP solutions to further improve the generalization of results in complex scheduling scenarios. Additionally, including factors of uncertainty in the simulation model would support efforts to increase the robustness of generated schedule results. Further, the current LMSP utilizes a set of first-level sorting policies. However, it is of interest to investigate the potential of integrating multi-stage sorting policies. Despite the higher model training effort expected when increasing the sorting policy complexity, it is anticipated that multi-level sorting policies will enhance the scheduling performance of future LMSP solutions using reinforcement learning.

REFERENCES

Andrade, P., C. Silva, B. Ribeiro, and B. F. Santos. 2021. “Aircraft Maintenance Check Scheduling Using Reinforcement Learning”. *Aerospace* 8(4):113.

Barnhart, C., P. Belobaba, and A. R. Odoni. 2003. “Applications of Operations Research in the Air Transport Industry”. *Transportation Science* 37(4):368–391.

Barnhart, C., and A. Cohn. 2004. “Airline Schedule Planning: Accomplishments and Opportunities”. *Manufacturing and Service Operations Management* 6(1):3–22.

Barnhart, C., and B. Smith. 2012. *Quantitative Problem Solving Methods in the Airline Industry*. 1st ed. Boston, MA: Springer.

Bazargan, M. 2016. *Airline Operations and Scheduling*. 2nd ed. London, UK: Taylor and Francis.

Belobaba, P., A. Odoni, and C. Barnhart. 2009. *The Global Airline Industry*. 2nd ed. Chichester, U.K: Wiley.

Cadarso, L., and R. de Celis. 2017. “Integrated Airline Planning: Robust Update of Scheduling and Fleet Balancing under Demand Uncertainty”. *Transportation Research Part C: Emerging Technologies* 81:227–245.

Callewaert, P., W. J. Verhagen, and R. Curran. 2018. “Integrating Maintenance Work Progress Monitoring into Aircraft Maintenance Planning Decision Support”. *Transportation Research Procedia* 29:58–69.

Černỳ, M. 2019. “Narrow Big Data in a Stream: Computational Limitations and Regression”. *Information Sciences* 486:379–392.

Deng, Q., B. F. Santos, and R. Curran. 2020. “A Practical Dynamic Programming Based Methodology for Aircraft Maintenance Check Scheduling Optimization”. *European Journal of Operational Research* 281(2):256–273.

Deng, Q., B. F. Santos, and W. J. Verhagen. 2021. “A Novel Decision Support System for Optimizing Aircraft Maintenance Check Schedule and Task Allocation”. *Decision Support Systems* 146:113545.

Feo, T. A., and J. F. Bard. 1989. “Flight Scheduling and Maintenance Base Planning”. *Management Science* 35(12):1415–1432.

Gopalan, R., and K. T. Talluri. 1998. “The Aircraft Maintenance Routing Problem”. *Operations Research* 46(2):260–271.

Gupta, P., M. Bazargan, and R. McGrath. 2003. “Simulation Model for Aircraft Line Maintenance Planning”. In *Annual Reliability and Maintainability Symposium, 2003*. January 27th-30th, Tampa, FL, 387-391.

Hu, Y., X. Miao, J. Zhang, J. Liu, and E. Pan. 2021. “Reinforcement Learning-Driven Maintenance Strategy: A Novel Solution for Long-Term Aircraft Maintenance Decision Optimization”. *Computers Industrial Engineering* 153:107056.

Hughes, G. 2006. “Right Place, Right Time: The Art of Short- & Long-Term Planning”. *Aircraft Commerce* Issue 46:53–57.

IATA 2021. “IATA Maintenance Cost Technical Group Report 2021”. https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/fy2021-mctg-report_public.pdf, accessed 13th March 2023.

Kiran, B. R., I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez. 2022. “Deep Reinforcement Learning for Autonomous Driving: A Survey”. *IEEE Transactions on Intelligent Transportation Systems* 23(6):4909–4926.

Kober, J., J. A. Bagnell, and J. Peters. 2013. “Reinforcement Learning in Robotics: A Survey”. *The International Journal of Robotics Research* 32(11):1238–1274.

Lagos, C., F. Delgado, and M. A. Klapp. 2020. “Dynamic Optimization for Airline Maintenance Operations”. *Transportation Science* 54(4):998–1015.

Lee, J., K. Lee, and I. Moon. 2022. “A Reinforcement Learning Approach for Multi-Fleet Aircraft Recovery under Airline Disruption”. *Applied Soft Computing* 129:109556.

Levin, A. 1971. “Scheduling and Fleet Routing Models for Transportation Systems”. *Transportation Science* 5(3):232–255.

Li, Yuxi 2018. “Deep Reinforcement Learning: An Overview”. https://arxiv.org/abs/1701.07274, accessed 13th March 2023.

OliverWyman 2022. “Global Fleet and MRO Market Forecast 2022-2032”. https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2022/feb/MRO-2022-Master-file_v5.pdf, accessed 22nd March 2023.

OpenAI 2023. “ChatGPT General FAQ”. https://help.openai.com/en/articles/6783457-chatgpt-general-faq, accessed 22nd March 2023.

Papakostas, N., P. Papachatzakis, V. Xanthakis, D. Mourtzis, and G. Chryssolouris. 2010. “An Approach to Operational Aircraft Maintenance Planning”. *Decision Support Systems* 48(4):604–612.

Ruther, S., N. Boland, F. G. Engineer, and I. Evans. 2017. “Integrated Aircraft Routing, Crew Pairing, and Tail Assignment: Branch-and-Price with Many Pricing Problems”. *Transportation Science* 51(1):177–195.

Sallab, A. E., M. Abdou, E. Perot, and S. Yogamani. 2016. “End-to-End Deep Reinforcement Learning for Lane Keeping Assist”. *Arxiv*:9 pp.

Senturk, C., and I. Ozkol. 2018. “The Effects of the Use of Single Task-Oriented Maintenance Concept and More Accurate Letter Check Alternatives on the Reduction of Scheduled Maintenance Downtime of Aircraft”. *International Journal of Mechanical Engineering and Robotics Research* 7(2):189–196.

Shaukat, S., M. Katscher, C. L. Wu, F. Delgado, and H. Larrain. 2020. “Aircraft Line Maintenance Scheduling and Optimisation”. *Journal of Air Transport Management* 89:101914.

Silver, D., J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton et al. 2017. “Mastering the Game of Go without Human Knowledge”. *nature* 550(7676):354–359.

Sriram, C., and A. Haghani. 2003. “An Optimization Model for Aircraft Maintenance Scheduling and Re-Assignment”. *Transportation Research Part A: Policy and Practice* 37(1):29–48.

Sutton, R. S., and A. G. Barto. 2018. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT press.

Tesauro, G. et al. 1995. “Temporal Difference Learning And TD-Gammon”. *Communications of the ACM* 38(3):58–68.

Witteman, M., Q. Deng, and B. F. Santos. 2021. “A Bin Packing Approach to Solve the Aircraft Maintenance Task Allocation Problem”. *European Journal of Operational Research* 294(1):365–376.

Zhou, L., Z. Liang, C.-A. Chou, and W. A. Chaovalitwongse. 2020. “Airline Planning and Scheduling: Models and Solution Methodologies”. *Frontiers of Engineering Management* 7(1):1–26.

AUTHOR BIOGRAPHIES

SIMON WIDMER is a visiting research student at UNSW Sydney from the Technical University of Denmark. His email address is widmersim@gmail.com.

SYED SHAUKAT is a PhD student in Aviation Management at UNSW Sydney, Australia. He has extensive professional experience in aircraft maintenance, operations, and scheduling. His email address is s.shaukat@unsw.edu.au.

CHENG-LUNG WU is an Associate Professor at the School of Aviation at UNSW Sydney, Australia. He specializes in solving operations and scheduling problems of airlines and airports. His email address is C.L.Wu@unsw.edu.au.