Journal of Open Aviation Science (2024), Vol.**2**
doi:10.59490/joas.2024.7891

JOAS

PROCEEDINGS | ***The 12th OpenSky Symposium***

Gradient-based smart predict-then-optimize framework for aircraft arrival scheduling problem

Go Nam Lui [*,1] and Soner Demirel[2]

[1]Management School, Lancaster University, United Kingdom
[2]Erzincan Binali Yildirim University, Turkey
*Corresponding author: g.n.lui@lancaster.ac.uk

(Received: 30 Oct 2024; Revised: 29 Jan 2025 and 8 Apr 2025; Accepted: 9 Apr 2025; Published: 24 Apr 2025)
(Editor: Xavier Olive; Reviewers: David Gianazza, Eri Itoh, Raúl Sáez)

Abstract

This paper introduces a gradient-based Smart Predict-then-Optimize (SPO) framework to solve the aircraft arrival scheduling problem (ASP) in the terminal maneuver area. Traditional approaches to ASP typically separate arrival time prediction from scheduling optimization, potentially leading to incomplete solutions. We address this limitation by developing an end-to-end learning framework that directly integrates prediction with optimization objectives. Our methodology introduces the concept of traffic instances for simultaneous prediction of multiple aircraft arrival times, coupled with a Mixed Integer Programming (MIP) model for scheduling optimization. We evaluated our approach using real-world data from London Gatwick Airport, analyzing 47,452 arrival flights from June to September 2024, organized into 2,404 traffic instances. The framework incorporates comprehensive weather data through the ATMAP algorithm, considering factors such as wind, visibility, precipitation, and dangerous phenomena. Experimental results demonstrate that the MLP+SPO+ framework shows particular effectiveness in adapting to adverse weather conditions, strategically balancing transit times with operational efficiency. While the minimum time interval is required, the MLP+SPO+ will reach around 85.0% and 43.4% lower costs compared with the First-Come-First-Serve (FCFS) cost and optimized true cost, respectively. These findings suggest significant potential for improving arrival scheduling efficiency through integrated SPO approaches.

Keywords: aircraft arrival scheduling problem; smart predict-then-optimize framework; machine learning; mixed integer programming

1. Introduction

Aircraft Arrival Scheduling Problem (ASP) is a crucial challenge in the field of Air Traffic Management (ATM). As global air traffic continues to grow, optimizing the sequence and schedule in which/when aircraft land at airports within Terminal Maneuvering Area (TMA) has become foremost. Efficient arrival scheduling not only reduces fuel consumption and carbon emissions but also significantly improves overall air traffic flow, making it a key focus for both researchers and practitioners in the field. The ASP, classified as an NP-hard problem, has spurred the development of various approaches to tackle its complexity. Traditional methods like First Come First Serve (FCFS) have laid the groundwork, while advanced techniques such as the Trombone [1, 2] and Point Merge System (PMS) [3] leverage geometric principles to further enhance efficiency. These innovations underscore the ongoing importance of solving the ASP to maintain safety, minimize delays, and optimize airport operations in increasingly congested airspace.

Addressing ASP has changed significantly in recent years as a result of increasing access to aeronautical data and rapid advances in machine learning (ML). Researchers have successfully applied diverse ML techniques to predict Estimated Time of Arrival (ETA) and arrival transit times with unprecedented accuracy. These advanced prediction models have not only enhanced our understanding of arrival patterns and potential delays but have also opened up new avenues for optimization. However, a significant gap remains in the field: while ETA prediction has seen substantial progress, the integration of these ML-driven predictions into optimization algorithms for ASP has been relatively unexplored, particularly in terms of optimization performance. Traditional two-stage approaches focus on minimizing prediction errors of certain parameters, typically using metrics such as Mean Square Error (MSE) ($\frac{1}{2}\|c - \hat{c}\|_2^2$) or Mean Absolute Error (MAE) ($\|c - \hat{c}\|_1$). After hyperparameter tuning and a training-validation procedure, the predicted parameter (c^*) is passed to a downstream optimization model. While these approaches have yielded valuable insights, they face significant limitations: 1. the emphasis on prediction error metrics fails to capture the quality of resulting decisions; 2. the disconnect between prediction and optimization stages can lead to feasibility issues.

This study aims to address these limitations by applying the smart predict-then-optimize (SPO) framework to the ASP within TMA. This approach is particularly relevant for the ASP because, even with fixed Standard Terminal Arrival Routes (STARs) and observable weather conditions, aircraft arrival transit times within TMAs can vary significantly due to unexpected factors that may influence decision errors during the landing process. Our work pioneers the application of the gradient-based SPO framework in the air transportation domain. Furthermore, we apply this framework to address a critical challenge in ASP: the incorporation of adverse weather conditions consideration.

The structure of this paper is as follow: Section 2 constructs a literature review for related works, and Section 3 introduces our methodologies. In Section 4, we briefly introduce our case study at London Gatwick airport and the setup of our experiment. Section 5 presents the results and discussion while Section 6 concludes this work.

2. Literature Review

Arrival scheduling is a critical factor in ensuring efficient operations within terminal maneuvering areas (TMAs). A central challenge involves assigning landing times to aircraft while adhering to separation criteria between successive arrivals. Prior studies frame this as an aircraft landing scheduling problem (ASP), where each aircraft must land within a predetermined time window bounded by an earliest and latest time [4]. These temporal constraints reflect operational realities:

- The earliest landing time represents the soonest achievable arrival under ideal conditions (e.g., maximum permissible speed, direct routing), while
- The latest landing time accounts for delay absorption capabilities via speed adjustments, path stretching, or holding patterns, constrained by fuel limits and airspace procedures.

This time window ensures efficient airspace utilization while accommodating uncertainties such as weather or traffic conflicts. Solutions aim to minimize deviations from target times and maintain safe separation, often derived from wake vortex categories or air traffic control (ATC) regulations. While early ASP formulations focused on single-runway allocation [4], extensions to multi-runway systems have become increasingly relevant for high-density airports. There are different approaches to solve this problem in the literature. Some studies focused on exact algorithms and optimization models [4, 5] while some others utilized heuristic and meta-heuristic algorithms to take advantage of reducing solving period [6, 7, 8, 9]. One study was focused on forming an heuristic algorithm to increase scheduling efficiency of arrival aircraft at London Heathrow. The algorithm showed that it

could have the potential to increase the efficiency of the decisions made by air traffic controllers [6]. In order to reduce the workload of air traffic controllers and congestion in airports, a metaheuristic algorithm was applied to a good initial solution to take advantage of its short computing time and the study was carried out in two Italian airports [7]. The use of an Ant Colony algorithm was investigated to focus on the aircraft scheduling problem. The algorithm was based on wake vortex modeling and findings are compared to some methods. This study showed that the algorithm based on wake vortex modeling revealed better results than models such as CPLEX, general ant colony algorithms, and approximation algorithm[8]. A data splitting algorithm was used to solve the aircraft sequencing problem. The model, 0-1 mixed integer programming, was employed with many different realistic constraints. The algorithm had small run times enabling a real-time deployment of the concept[9]. For more details concerning the aircraft scheduling problem, we refer two review studies on this topic [10, 11].

In recent years, the landscape of arrival management research has been transformed by the increasing availability of aviation data, leading to a surge in ML-based approaches for arrival time prediction. The effort that has been spent on predicting arrivals flight time and its contribution to different ATM solutions are important to have more predictable, efficient and greener operations in TMAs [12]. ML has an important role on reaching the goals contributing to providing better air traffic management. In the existing literature, there are different application of its algorithms focusing on Estimated Time of Arrival (ETA) / arrival flight time [13, 14, 15, 16, 17, 18, 19].

Quantile Regression Forests [13], a tree-based ensemble method, was employed for estimation of landing times. A total of 4011 cases were separated 67% and 33% for training and testing respectively. As stated in the research, the model was suitable to predict landing times in real-time applications. Random Forest (RF) [14], a well-known tree-based method, was utilized to improve prediction on ETA. In the application, feature generation and selection was one of the main focus points. As a result of this study, they showed that 78% of total instances have better accuracy within the ML algorithm against Enhanced Traffic Management System in US. Some regression models (Linear, Non-linear and Ensemble) and Recurrent Neural Network [15] were tested to perform prediction of ETA for commercial flights by comparing their model results with EUROCONTROL ETA predictions. One of the main outlines of this study was higher accuracy with smaller standard deviation which made smaller prediction windows of ETA possible. Spatiotemporal Neural Network Model for ETA [17] was proposed with three main stages that were trajectory pattern recognition, trajectory prediction and arrival time prediction. At the conclusion of their research, one of the findings was that the MAE was typically lower with shorter travel times to the destination. A deep learning approach based on Long-Short Term Memory [18] was used to predict ETA by utilizing 4D trajectory of the aircraft and weather data. In addition to the model's result, this research came to the front with its application airport, Madrid Barajas-Adolfo Suárez (Spain). The performed model was superior to RF, Gradient Boosting Machines (GBM) and Adaptive Boosting that were selected as baseline in the study. Ridge Regression (RR) and GBM [16] were selected to predict runway and gate arrival time of flights, based on historical, weather, air traffic control and given data during the data science contest named as GE Flight Quest.

Despite these significant advances in both optimization and prediction domains, several gaps remain in the current literature. Because most researchers handle these problems separately, there exists a disconnect between arrival time prediction and scheduling optimization. While both areas have seen remarkable progress independently, the potential benefits of integrating prediction capabilities into optimization frameworks remain largely unexplored. Few studies have explored this area, but they mostly used the predicted values directly for the downstream optimization [20, 21]. The relationship between prediction accuracy and operational efficiency improvements needs more thorough investigation. Traditional methods also often fail to capture the dynamic nature of the airport envi-

ronment, where predictions and scheduling decisions need to be made and updated continuously in response to changing conditions.

Recent developments in computational frameworks offer promising directions for addressing these limitations. SPO framework [22] provide a structured approach to integrating prediction and optimization, potentially offering a more coherent solution to the arrival scheduling problem. Similarly, learning-to-optimize techniques [23], which directly learn optimization strategies from data, may offer more robust solutions than traditional two-stage approaches. However, while these frameworks show theoretical promise, their practical application in aviation context remains limited. Key challenges include adapting these frameworks to handle the specific constraints and objectives of airport operations and validating their performance under real-world conditions and operational constraints. Given these challenges and opportunities in the existing literature, this research proposes the SPO framework for ASP inside the TMAs. The following section details our proposed approach and its implementation.

3. Methodologies

Fig. 1 presents the general schematic diagram of our proposed method. Starting from the raw flight data, we generate an input dataset $\mathcal{D}$ through a series of data preprocessing, including data trimming, cleaning, and re-alignment. $\mathcal{D}$ consists of K independent traffic instances with the same number of flights, where each instance is represented as a pair $(\mathbf{x}, \mathbf{c})$. For each instance, the input features $\mathbf{x}$ are structured as a vector contains $m \times n_t$ features, where m represents the number of input features for each flight, n_t represents the number of flights in each traffic instance. The corresponding output costs $\mathbf{c}$ are represented as a vector of length n_t, where each element represents the cost associated with each flight in the traffic instance. Therefore, the input dataset can be denoted as $\{(\mathbf{x}_k, \mathbf{c}_k)\}_{k=1,\dots,K}$.

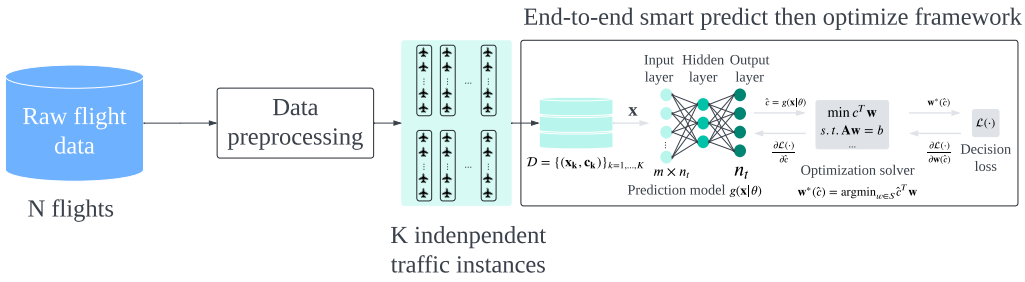


Figure 1. The schematic diagram of end-to-end smart predict-then-optimize framework for aircraft arrival scheduling problem

Based on the dataset $\mathcal{D}$, we can implement the SPO framework [24]. Considering ASP as an integer programming problem, we have several key elements: a feasible region S, an optimal objective value $z^*(c)$ corresponding to objective coefficients c, and an optimal solution $w^*(c)$. Such optimization model will be embedded into a differentiable prediction model $g(\mathbf{x}|\theta)$, such as neural networks, through the decision loss $\mathcal{L}(\cdot)$.

The core function of this framework is the gradient computation and the parameter updates through the backpropagation. For each training instance, the gradient $\frac{\partial \mathcal{L}}{\partial \theta}$ is computed by applying the chain rule. $\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial w^*} \frac{\partial w^*}{\partial \hat{c}} \frac{\partial \hat{c}}{\partial \theta}$. Here, $\frac{\partial \mathcal{L}}{\partial w^*}$ measures how the decision loss changes with respect to the optimal solution, $\frac{\partial w^*}{\partial \hat{c}}$ captures the sensitivity of the optimal solution to changes in the objective coefficients, and $\frac{\partial \hat{c}}{\partial \theta}$ represents how the predicted coefficients vary with the model parameters. Through

this gradient chain, the framework enables end-to-end training where the optimization outcomes directly influence the prediction model's parameter updates.

3.1 Aircraft arrival scheduling problem formulation

In this work, we formulate the ASP as a simple Mixed Integer Programming (MIP) model based on the classical single runway aircraft landing problem proposed by [4]. We assume:

- $\mathcal{A} = \{1, \ldots, n\}$: Set of aircraft, where n is the total number of aircraft
- $i, j \in \mathcal{A}$: Aircraft indices
- T_i: The target (expected) landing time for aircraft i
- E_i: The earliest landing time for aircraft i
- L_i: The latest landing time for aircraft i
- $s_{i,j}$: The required separation time between i & j, where i lands before j
- c_i: Delay costs for aircraft i landing after the expected time T_i
- M: A large constant
- $\hat{T}_i$: The predicted transit time for aircraft i

The decision variables in our models are:

- y_i: Actual landing time of aircraft i
- ω_i: Binary variable indicating if aircraft i lands after its expected time

$$\omega_i = \begin{cases} 1 & \text{if } y_i > T_i \\ 0 & \text{otherwise} \end{cases}$$

- $\delta_{i,j}$: Binary variable for aircraft arrival scheduling

$$\delta_{i,j} = \begin{cases} 1 & \text{if aircraft } i \text{ lands before aircraft } j \\ 0 & \text{otherwise} \end{cases}$$

The objective of this model is to minimize the sum of costs for all delayed aircraft, where:

$$\min \sum_{i \in \mathcal{A}} c_i \omega_i$$

The model formulation is listed as follows:

s.t.

$$E_i \leq y_i \leq L_i \qquad \forall i \in \mathcal{A} \tag{1}$$

$$y_i - T_i \leq M \cdot \omega_i \qquad \forall i \in \mathcal{A} \tag{2}$$

$$y_i - T_i \geq -M \cdot (1 - \omega_i) \qquad \forall i \in \mathcal{A} \tag{3}$$

$$\delta_{i,j} + \delta_{j,i} = 1 \qquad \forall i, j \in \mathcal{A}, i \neq j \tag{4}$$

$$y_j - y_i \geq s_{i,j} - M \cdot \delta_{j,i} \qquad \forall i, j \in \mathcal{A}, i \neq j \tag{5}$$

$$y_i \in \mathbb{R} \qquad \forall i \in \mathcal{A} \tag{6}$$

$$\omega_i \in \{0, 1\} \qquad \forall i \in \mathcal{A} \tag{7}$$

$$\delta_{i,j} \in \{0, 1\} \qquad \forall i, j \in \mathcal{A}, i \neq j \tag{8}$$

Our ASP seeks to minimize delay-related costs. At its core, the mathematical formulation employs a simple objective function that sums the costs across all delayed aircraft. Three decision variables drive the model: continuous variables y_i for landing times, binary indicators ω_i for delays, and ordering variables $\delta_{i,j}$ that establish the sequence of operations between aircraft pairs. These variables work in concert to capture all necessary scheduling decisions.

Constraint (1) ensures that each aircraft i must be scheduled within its feasible time window $[E_i, L_i]$. Constraint (2) and (3) define whether an aircraft is delayed using the big-M method. If the actual arrival time y_i exceeds the expected time T_i, the aircraft is considered delayed ($\omega_i = 1$). The constraints work in pairs to force ω_i to take the appropriate binary value. Constraint (4) refers to the ordering constraint, in which any pair of aircraft (i, j), either i must precede j or j must precede i. Constraint (5) works in conjunction with the ordering constraint (4) to ensure proper separation between any pair of aircraft:

1. When aircraft i lands before j ($\delta_{i,j} = 1$, $\delta_{j,i} = 0$):
   - The constraint becomes: $y_j - y_i \geq s_{i,j}$, this enforces the minimum separation time $s_{i,j}$ between landings.
2. When aircraft j lands before i ($\delta_{i,j} = 0$, $\delta_{j,i} = 1$):
   - The constraint becomes: $y_j - y_i \geq s_{i,j} - M$, the large M term makes this constraint non-binding.
   - Meanwhile, the complementary constraint $y_i - y_j \geq s_{j,i} - M \cdot \delta_{i,j}$ becomes active.
   - This enforces the minimum separation time $s_{j,i}$ between landings.

Thus, the pair of constraints ensures proper separation regardless of landing order, with $s_{i,j}$ applied when i precedes j and $s_{j,i}$ applied when j precedes i. The rests are domain constraints for the decision variables.

The conventional delayed cost definition is $c_i = c_i^* \cdot (\hat{T}_i - T_i)$, where c_i^* denotes the unit time delayed cost for each aircraft type [25], $(\hat{T}_i - T_i)$ refers to the delayed time. For our optimization framework, we can simplify this cost representation due to two key observations. First, the expected arrival time T_i is known before the prediction task begins. Second, the unit delay cost c_i^*, which varies by aircraft type and is typically derived from extensive operational cost studies, is also predetermined. Given these fixed parameters, the delay cost c_i maintains a direct proportional relationship with the predicted arrival time $\hat{T}_i$. This proportional relationship enables us to streamline our cost representation by using $\hat{T}_i$ directly as our cost metric ($c_i \approx \hat{T}_i$). While this simplification might appear to lose some granularity, it preserves the essential mathematical properties needed for optimization while reducing computational complexity.

3.2 Costs prediction via traffic instances

Traditional approaches to ETA prediction focus on individual flight independently. For each flight i, m input features–comprising pre-terminal flight data, meteorological conditions, and historical patterns–to forecast the estimated flight duration $\hat{T}_i$ for each flight. When integrating ML with the optimization framework, we need to reconceptualize the prediction task to align with the objective. In SPO framework, the ML model iteratively attempts to minimize the decision loss—a task that requires optimization for multiple aircraft than individual. To address this issue, we propose *traffic instances*, which refers to a certain air traffic scenario that contains the same amount of flights that needs to be resolved. Instead of mapping m features to a single flight duration, we predict flight times for an entire *traffic instance* simultaneously. Each *traffic instance* contains n_t flights, transforming our input dimension to $n_t \times m$ features and generating outputs that directly correspond to the costs (n_t features) for decision loss computation.

Algorithm 1 Non-Overlapping Traffic Instance Generation

Require:
Flight sequence $F = \{f_1, ..., f_m\}$ ordered by entry time
Instance size N
Maximum time interval ΔT_{max}
Ensure:
Set of **non-overlapping** instances I where:
    1. Each instance contains exactly N flights
    2. All flights in an instance occur within ΔT_{max}
    3. No two instances share any flights
function GENERATEINSTANCES($F, N, \Delta T_{max}$)
    $I \leftarrow \emptyset$ ▷ Initialize empty instance set
    $i \leftarrow 0$ ▷ Start at first flight
    while $i + N \leq |F|$ **do**
        $G \leftarrow \{f_i, ..., f_{i+N-1}\}$ ▷ Next candidate group
        $\Delta T \leftarrow f_{i+N-1}$.time $- f_i$.time
        if $\Delta T \leq \Delta T_{max}$ **then**
            $I \leftarrow I \cup \{G\}$ ▷ Commit valid instance
            $i \leftarrow i + N$ ▷ **Key: Jump ahead by N flights**
        else
            $i \leftarrow i + 1$ ▷ Reject group, check next flight
        end if
    end while
    return I
end function

Algorithm 1 constructs strictly non-overlapping traffic instances from temporally ordered flights using a hybrid windowing strategy. For each candidate group of N consecutive flights, the algorithm commits it as a valid instance *only if* its temporal span satisfies $\Delta T \leq \Delta T_{max}$, then advances the window by N flights to prevent overlap. If rejected (i.e., $\Delta T > \Delta T_{max}$), the window slides forward by 1 flight to explore alternative groupings while preserving temporal density. This ensures: 1) **mutual exclusivity** between instances (no shared flights), 2) **temporal coherence** (all flights within Δ_{max}), and 3) **leakage prevention** through day-stratified splitting, where all instances from a calendar day reside exclusively in either the training or test set.

Based on the traffic instances, we can perform prediction task via ML. The prediction model in this framework has to be differentiable, we here proposed two simple model as our baseline, including Linear Regression (LR: $f(x) = Wx + b$) and Multi-Layer Perceptron (MLP):

$$f(x) = f_2(\text{ReLU}(f_1(x)))$$

where:

$$f_1(x) = W_1 x + b_1 \text{ (first layer)}$$
$$f_2(x) = W_2 x + b_2 \text{ (second layer)}$$
$$\text{ReLU}(x) = \max(0, x) \text{ (activation function)}$$

As mentioned in Section 3.1, the output is the predicted transit times $\hat{T}_i$ for each traffic instances. For the input x, we refer to the common features in previous ETA prediction studies [12, 26, 27], including initial position (latitude, longitude, altitude) and operation (heading, speed, descent rate) state for individual aircrafts enter the terminal area.

3.3 Decision loss

The decision loss in our framework is based on the SPO loss introduced by [22]. This loss measures how well our predicted costs lead to optimal decisions compared to decisions made with true costs. The rigorous unambiguous SPO loss is defined as:

$$\mathcal{L}_{SPO}(\hat{c}, c) = \max_{\omega \in W^*(\hat{c})} (c^T \omega) - z^*(c) \tag{9}$$

where:

- $W^*(\hat{c})$ is the set of optimal solutions using predicted costs $\hat{c}$
- $z^*(c)$ is the optimal objective value using true costs c
- The max operator accounts for multiple optimal solutions that could arise from $\hat{c}$

However, numerical studies in [24] demonstrate that this rigorous form yields similar results to a simplified version known as "regret":

$$\mathcal{L}_{SPO}(\hat{c}, c) = c^T \omega^*(\hat{c}) - z^*(c) \tag{10}$$

where:

- $\omega^*(\hat{c})$ is an optimal solution obtained using predicted costs $\hat{c}$

This measures the gap between the true cost of decisions made by predicted costs $c^T \omega^*(\hat{c})$, and the best possible cost achievable with true costs $z^*(c)$. While we use this regret formulation for evaluation purposes, it isn't directly suitable for training due to its computational intractability. In the following section, we introduce the tractable version of SPO functions that enable gradient-based training while maintaining the spirit of optimizing decision loss.

3.3.1 Smart predict-then-optimize plus (SPO+)

Since the SPO is intractable, Elmachtoub and Grigas [22] derived a surrogate convex upper bound for SPO called SPO+:

$$\mathcal{L}_{SPO+}(\hat{c}, c) = \max_{\omega \in S}(c^T \omega - 2\hat{c}^T \omega) + 2\hat{c}^T \omega^*(c) - z^*(c) \tag{11}$$

The computation of SPO+ involves solving a modified optimization problem with costs $(2\hat{c} - c)$ in the forward pass, where the loss is computed with appropriate sign adjustments for maximization problems [24]. The backward pass then enables end-to-end training by computing gradients based on the difference between true and predicted optimal solutions, scaled by 2 and adjusted for the optimization sense (minimization or maximization).

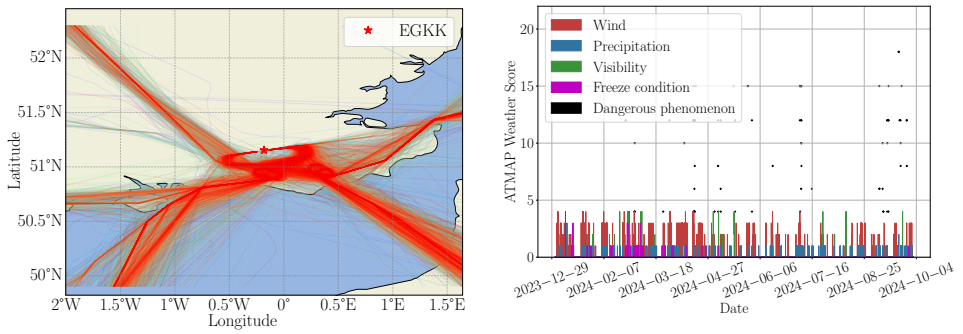
4. Case study at London Gatwick Airport

In this paper, we construct our study in London Gatwick Airport. London Gatwick Airport (ICAO: EGKK) serves as a major international aviation hub in the United Kingdom. Operating with a single runway system—unique among airports of its size and traffic volume—Gatwick stands as London's second-busiest airport and the second-largest single-runway airport globally, located approximately 29.5 miles south of Central London. In 2024 until October, it already handled 203,439 traffic including both arrivals and departures[1].

[1] https://ansperformance.eu/dashboard/stakeholder/airport/db/EGKK.html

4.1 Data description

47,452 of arrival flights (ADS-B data) at EGKK from June 2024 to September 2024 obtained from `OpenSky Network` [28] are used in this study. For the local weather information, we refer to the Meteorological Terminal Aviation Routine Weather Report (METAR) of EGKK in 2024[2]. METAR is a weather report which contains the information for an area enclosed within a 16 km radius around the airport. Raw METAR data offers a series of weather information, such as wind, temperature, visibility, moisture, etc. Based on the raw METAR data, we apply the air traffic management airport performance (ATMAP) weather algorithm [29, 30] to extract the certain scores for each weather component, including wind, visibility, precipitation, freeze condition, and dangerous phenomenon.



(a) 10,000 EGKK arrivals, June-September 2024 **(b)** Weather score distribution of EGKK in 2024 (until October)

Figure 2. Flight trajectories and local weather visualization of EGKK

Fig. 2a illustrates 10,000 sample flights in the scope of our study, capturing the terminal maneuvering area where arriving aircraft perform final approach sequences. The flight trajectories used in this study align with Gatwick Airport's approach procedures. Fig. 2b presents the weather score distribution of EGKK in 2024. As the figure illustrates, wind components are the most significant weather events in EGKK, consistently showing the highest scores throughout the observed period. The wind scores frequently reach values 2.5 on the weather score scale. Precipitation issues also contribute to the overall weather conditions but to a lesser extent. Freeze conditions are more frequent in winter period but less important during summer season. Visibility appears to be relatively minimal, showing lower scores and frequency compared to other weather components. Dangerous phenomena are occasionally recorded but remain relatively rare events in the dataset.

4.2 Experiment setup

Table 1 summarizes the key parameters and configurations of our experimental setup. The study encompasses 2,404 traffic instances from 47,452 arrivals, with each instance involving 15 aircraft within a 45-minute time interval. The area of interest is confined to a 50 Nautical mile radius around EGKK, providing comprehensive coverage of the TMA.

As mentioned in Section 3.2, we implement two ML approaches for our analysis: LR and MLP. For the ASP, the model parameters need to be pre-set and static during the training process, we select typical scenarios from the instances to define the parameters, characterized by maximum weather parameters including wind ($Wind_{max}$), precipitation ($Precipitation_{max}$), visibility ($Visibility_{max}$), and dangerous phenomena ($DangerousPhenomenon_{max}$), along with minimum time interval ($Time_{min}$). Since the scope of our data is from June to September 2024, we do not consider freeze condition in this work. The typical scenario will affect the parameter setting of the optimization model, where

[2] https://www.ogimet.com/metars.phtml.en

Table 1. The key setup for the experiment

Period	June - September 2024
Number of aircraft	47,452
Number of traffic instances	2,404
Number of aircraft per instances	15
Maximum time interval per instance	45 minutes
Area of interest	50 Nautical miles around EGKK
Machine learning models	{Linear regression; Multi-layer perceptron}
Typical scenarios	$\{Wind_{max}, Precipitation_{max}, Visibility_{max}, DangerousPhenomenon_{max}, Time_{min}\}$
Input features	{Latitude, longitude, velocity, heading angle, vertical rate} at entry state
Output feature	Transit time
Loss function	SPO+, Mean Square Error (Two-stage approach)

T_i will be the expected relative transit time to the first entry aircraft within that instance, $E_i = T_i - 60$ and $L_i = T_i + 1800$ refers to an open-source ASP benchmark [9, 11][3]. While this benchmark simplifies aircraft-specific performance, (e.g., it does not dynamically model BADA parameters), it provides a tractable framework for scheduling algorithms. The required separation time $s_{i,j}$ is derived from wake turbulence categories (WTC). Aircraft type codes are mapped to WTC classifications (Light, Medium, Heavy, Jumbo) using the Aircraft Database provided by `OpenSky Network` [28]. The required separation time is then determined based on the WTC of the preceding and succeeding aircraft[4].

The input feature space comprises five key aircraft parameters at the entry state: latitude, longitude, velocity, heading angle, and vertical rate. These parameters capture the essential initial conditions of each aircraft's trajectory. The models are trained to predict the transit time as the output feature. For model optimization, we employ two distinct loss functions: SPO+, and Mean Square Error (MSE) in a two-stage approach. The ratio between training sets and test sets are $8:2$. The batch size is 32 and number of epochs is 20.

5. Results and Discussion

In this section, we will present the results and corresponding discussions. First, Fig. 3 illustrates the learning curves for both loss functions on the training sets using normalized loss values. The SPO+ and two-stage approaches exhibit distinctly different convergence behaviors during training. The SPO+ loss curves show rapid initial decrease and stabilize at very low normalized loss values (below 0.1) across all scenarios by around iteration 250. This consistent convergence pattern appears similar for both Linear Regression and MLP implementations.

The two-stage approach, however, demonstrates markedly different behavior. While the Linear Regression variants show quick initial convergence, MLP implementations maintain relatively high normalized loss values (fluctuating between 0.2 and 0.6) throughout training. The learning curves show considerable oscillation, particularly for the maximum danger scenario, suggesting potential stability issues in the optimization process.

This performance discrepancy suggests that for subsequent analyses, focus should be directed toward three specific configurations: LR + Two-Stage, MLP + SPO+, and LR + SPO+. The MLP +

[3] http://data.recherche.enac.fr/ikli-alp/
[4] https://knowledgebase.vatsim-germany.org/books/separation/page/wake-turbulence-separation

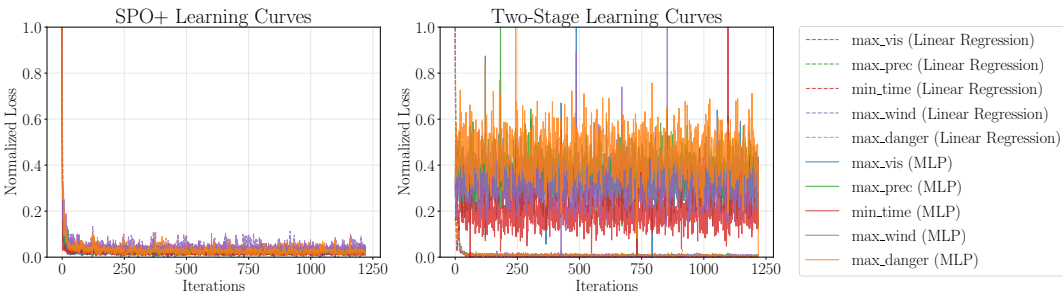


Figure 3. The learning curves for two-stage approach and SPO+ on training sets

Two-Stage configuration can be reasonably excluded from further investigation due to its demonstrated inferior convergence properties.

Following the first analysis on learning curves, Fig. 4 presents the normalized regret distribution during training process for test sets. Our experimental results demonstrate the effectiveness of end-to-end decision-focused learning approaches, particularly when combined with more expressive model architectures. The MLP + SPO+ implementation consistently achieves superior performance across most typical scenarios, exhibiting lower normalized regret compared to both LR + SPO+ and LR + Two-Stage approaches.

To rigorously assess the performance differences between approaches, we employed the Mann-Whitney U test, a non-parametric statistical test that evaluates whether two independent samples come from the same distribution. This test is particularly appropriate for our analysis as it makes no assumptions about the normality of the data and is well-suited for comparing the regret distributions. A lower U-statistic indicates greater separation between the distributions.

Statistical analysis reveals particularly significant differences in the maximum wind scenario, where MLP + SPO+ significantly outperforms the two-stage approach ($U = 130.0$, $p = 0.024$). This advantage is also suggested, though not statistically significant at the $\alpha = 0.05$ level, in the maximum dangerous phenomenon scenario ($U = 147.0$, $p = 0.066$). These findings support the hypothesis that the ability to capture non-linear relationships proves beneficial in complex scenarios.

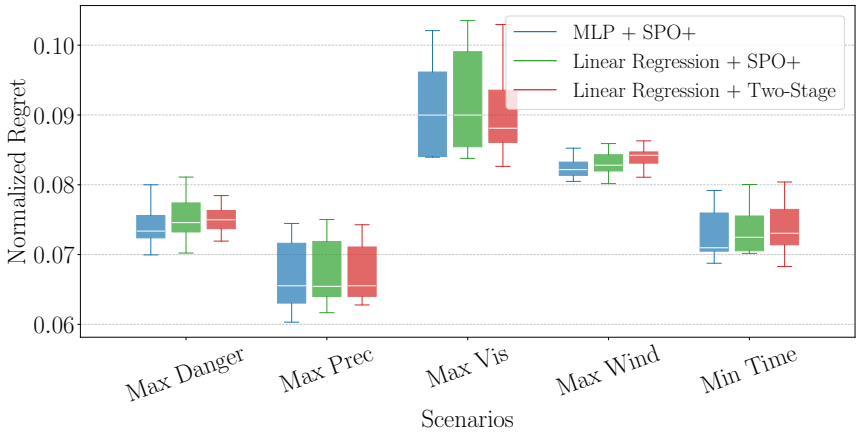


Figure 4. Normalized regret distributions across test sets for different scenarios.

Interestingly, while SPO+ generally shows favorable performance, the statistical tests reveal no sig-

nificant differences between LR + SPO+ and LR + Two-Stage across most scenarios (all $p > 0.05$), with particularly similar performance in maximum precipitation ($U = 208.0$, $p = 0.763$) and maximum visibility ($U = 250.5$, $p = 0.458$) scenarios. We further conducted additional Mann-Whitney U tests on the union of all data subsets (combining all weather scenarios). These aggregated results confirm no statistically significant differences between any of the methods, SPO+ (MLP) vs Two-Stage (LR): $U = 5248.0$, $p = 0.549$. This comprehensive analysis across all weather conditions further nuances our understanding of the relative performance of these approaches, suggesting that the advantages of SPO+ might be more subtle than initially apparent in certain contexts.

The variation in performance across different architectures and optimization frameworks provides valuable insights for practical implementations. The notable success of MLP + SPO+ not only demonstrates the advantage of end-to-end SPO learning but also highlights the importance of model expressiveness in capturing complex weather-related patterns. These findings suggest that while SPO+ generally provides stronger performance, the choice of underlying model architecture significantly influences the overall effectiveness of the optimization framework.

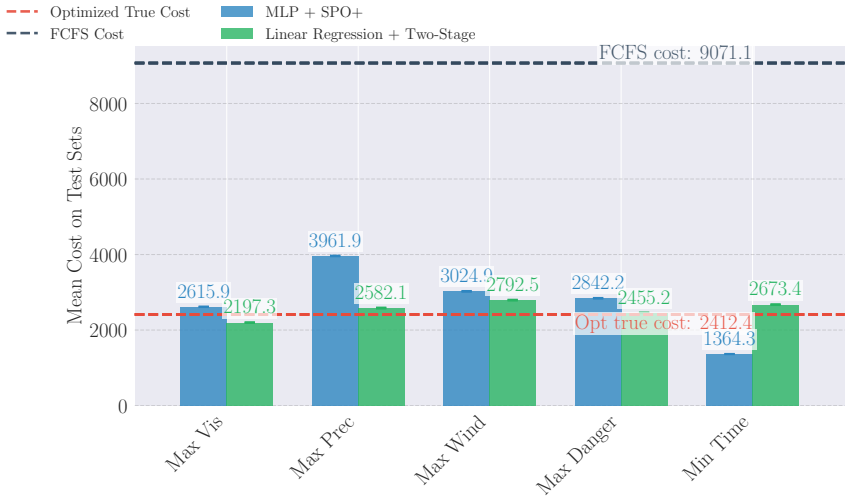


Figure 5. The mean cost comparison between FCFS, optimized true cost, and optimized predicted cost on test sets

The next analysis compares the optimized costs of SPO+ and two-stage approaches based on the trained ML models. We input the features of test sets to predict the costs and use these predictions to optimize each instance via the ASP in the test sets (Fig. 5). With a cost of 2,412.4 when optimizing using true landing times—representing the minimum achievable average cost under the specific scheduling constraints we defined—both MLP+SPO+ and LR+Two-Stage methods significantly outperform the FCFS baseline of 9,071.1. Interestingly, the MLP+SPO+ shows particularly strong performance in scenarios optimized for minimum time interval, achieving a mean cost of 1,364.3 compared to Two-Stage's 2,673.4. This outperformance relative to the "optimal true cost" does not indicate a violation of optimization principles, but rather highlights a key insight: optimization using true landing times isn't necessarily optimal for the complete operational context. The SPO+ approach can discover solutions that account for broader operational dynamics and uncertainty patterns that aren't captured when directly optimizing with true landing times. The most significant insight emerges from examining performance across different weather conditions: while the Two-Stage approach maintains relatively uniform costs across all scenarios, the SPO+ method demonstrates sophisticated adaptation to weather conditions, strategically accepting higher transit times under challenging conditions while finding better overall solutions.

This weather-responsive behavior of MLP+SPO+ represents a crucial advancement in arrival scheduling optimization. The systematically higher costs observed under extreme weather scenarios (ranging from 2,615.9 to 3,961.9) indicate that the model effectively incorporates weather-related risks into its decision-making process, making more conservative prediction when conditions are adverse. In contrast, the Two-Stage approach's more uniform cost distribution suggests a limitation in capturing the complex interplay between weather conditions and optimal routing decisions. These findings indicate that while SPO+ might occasionally suggest higher transit times compared to the optimal true cost, these decisions reflect a trade-off between speed and safety, demonstrating the method's capability to make more nuanced, context-aware aircraft arrival scheduling decisions.

In addition to algorithmic analysis, we perform a delay assignment analysis to evaluate the fairness consideration in this model. We use the transit time difference for each aircraft and the number of shifting for maximum precipitation scenario. This scenario is selected because it has the largest total cost for MLP+SPO+. comparing between MLP+SPO+ and optimization using true cost.

Table 2. Transit time difference and number of position shifting for maximum precipitation scenario.

	Transit time difference			**No.position shifting per instance**
	Mean [s]	Standard deviation [s]	Max [s]	Mean
Optimization with true cost	43.62	236.69	703.54	17
MLP+SPO+	**18.60**	**181.67**	**572.10**	**13**

Table 2 reveals that MLP+SPO+ demonstrates improved fairness compared to optimization with true cost, as evidenced by lower mean transit time differences (18.60s vs. 43.62s), reduced standard deviation (181.67s vs. 236.69s), and fewer position shifts per instance (13 vs. 17) in the maximum precipitation scenario. Consider we have 15 aircraft per instance, MLP+SPO+ can achieve average less than 1 position shifting for each aircraft. However, since neither MLP+SPO+ nor the baseline explicitly incorporates fairness parameters, both methods exhibit high variability in transit time differences, reflected in the large standard deviations. This suggests that while MLP+SPO+ achieves better fairness outcomes implicitly through its learning framework, the absence of fairness-aware optimization leads to inconsistent treatment of individual aircraft. The results highlight the potential for further improvements by integrating fairness constraints directly into the model to reduce disparity and stabilize outcomes.

6. Conclusion

This paper presents an application of the SPO framework to the Aircraft Arrival Scheduling Problem within Terminal Maneuvering Area. We developed an end-to-end learning approach that integrates arrival flight time prediction with scheduling optimization, specifically focusing on London Gatwick Airport operations. Our methodology introduces the concept of traffic instances for simultaneous prediction of multiple aircraft arrival times, coupled with a Mixed Integer Programming model for optimal aircraft arrival scheduling decisions.

The experimental results demonstrate several key findings. First, the MLP+SPO+ implementation consistently outperforms traditional two-stage approaches across most scenarios, particularly with complex weather conditions. The framework shows sophisticated adaptation to varying weather conditions, strategically accepting higher transit times under adverse conditions while maintaining operational efficiency. When the minimum time interval is required, the MLP+SPO+ will suggest around 43.4% lower costs compared with the true cost. Second, our analysis reveals that while simpler LR models with two-stage optimization can sometimes match SPO+ performance in specific scenarios (particularly low visibility conditions), the end-to-end approach generally provides more

robust and adaptable solutions.

A critical consideration for practical implementation is balancing operational efficiency with ATC manageability and fairness to airlines. FCFS scheduling is conventionally favored for its simplicity and perceived fairness. Our proposed framework demonstrates that optimized sequences can achieve significant cost reductions without inherently compromising these priorities. Compared with benchmark optimization, MLP+SPO+ demonstrates enhanced fairness.

However, our study identifies important limitations and areas for refinement. Methodologically, our focus on isolating the SPO+ loss function's impact led us to maintain consistency by using unnormalized inputs and Gradient Descent (GD) optimization across the compared methods (e.g., LR+SPO+ vs. LR+2S). While this consistency aids in evaluating the relative benefit of the SPO+ loss, it presents trade-offs. Using unnormalized inputs might not yield the absolute peak performance, particularly for MLP architectures known to benefit from normalization, although our results still confirmed the SPO+ advantage. Similarly, while GD (or other gradient-based methods) is inherent to optimizing the SPO+ loss, applying it to the LR+2S baseline (instead of standard OLS) ensures optimizer consistency for comparison but deviates from typical standalone LR practices. Furthermore, as our experiments suggested, optimal training, particularly concerning input normalization, appears sensitive to hyperparameter calibration, especially for LR models under GD where we encountered convergence challenges with normalization in our initial trials. Beyond these methodological considerations, a significant constraint remains the current SPO framework's reliance on fixed optimization model structures (beyond objective costs), limiting adaptability to scenarios with varying constraints. Computational efficiency for larger instances and the lack of explicit fairness mechanisms, potentially leading to higher variation in delay assignment, are also key concerns.

Looking ahead, several promising research directions emerge. Extending the SPO framework itself, perhaps incorporating dynamic MIP parameter updates [31] and regret computations, is a key avenue. This could involve exploring diverse neural network architectures for traffic instance cost prediction. Crucially, a systematic investigation into the interplay between input normalization techniques, hyperparameter tuning, and model performance (both SPO+ and baselines) is warranted. This includes exploring individually optimized configurations, potentially using OLS for LR+2S baselines when comparing absolute achievable performance rather than isolating loss function effects. Improving computational efficiency, possibly through optimization problem relaxations, remains vital. The framework's principles could also be extended to related scheduling or routing problems [32, 33], and transfer learning could enhance applicability across different airports. Lastly, systematically addressing fairness is essential. Future work should explicitly incorporate airline equity metrics (e.g., delay distribution thresholds) as constraints or weighted objectives in the optimization model, better aligning the framework with real-world ATC priorities while preserving its efficiency advantages.

These findings and identified future directions contribute to the growing body of research on ML applications in air traffic management, particularly in the critical area of arrival scheduling optimization. The demonstration of end-to-end SPO learning approaches suggests potential for further development and practical implementation in real-world airport operations.

Author contributions

Go Nam Lui: Conceptualization, methodology, formal analysis, data curation, software, resources, writing – original draft, writing – review & editing, visualization. *Soner Demirel:* Conceptualization, data curation, writing – original draft, writing – review & editing.

Funding statement

Go Nam Lui receives funding from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant numbers 10086651 (Lancaster University)]. Opinions expressed in this work reflect the authors views only, and the SESAR 3 JU and UKRI are not responsible for any use that may be made of the information contained herein.

Open data statement

All data analyzed during this study are publicly available in https://zenodo.org/records/14014439.

Reproducibility statement

The source code of this research is stored at https://github.com/harrylui1995/ASP_E2EPO.

References

[1] Kevin R Sprong, Brennan M Haltli, James S DeArmon, Suzanne Bradley, et al. "Improving flight efficiency through terminal area RNAV". In: *6th USA/Europe Air Traffic Management R&D Seminar*. 2005.

[2] Raúl Sáez, Xavier Prats, Tatiana Polishchuk, and Valentin Polishchuk. "Traffic synchronization in terminal airspace to enable continuous descent operations in trombone sequencing and merging procedures: An implementation study for Frankfurt airport". In: *Transportation Research Part C: Emerging Technologies* 121 (2020), p. 102875.

[3] Ludovic Boursier, Bruno Favennec, Eric Hoffman, Aymeric Trzmiel, François Vergne, and Karim Zeghal. "Merging arrival flows without heading instructions". In: *7th USA/Europe air traffic management R&D seminar*. 2007, pp. 1–8.

[4] John E Beasley, Mohan Krishnamoorthy, Yazid M Sharaiha, and David Abramson. "Scheduling aircraft landings—the static case". In: *Transportation science* 34.2 (2000), pp. 180–197.

[5] Maximilian Pohl, Rainer Kolisch, and Maximilian Schiffer. "Runway scheduling during winter operations". In: *Omega* 102 (2021), p. 102325.

[6] John E Beasley, Julia Sonander, and P Havelock. "Scheduling aircraft landings at London Heathrow using a population heuristic". In: *Journal of the Operational Research Society* 52.5 (2001), pp. 483–493.

[7] Marcella Sama, Andrea D'Ariano, Alessandro Toli, Dario Pacciarelli, and Francesco Corman. "A variable neighborhood search for optimal scheduling and routing of take-off and landing aircraft". In: *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE. 2015, pp. 491–498.

[8] Bo Xu. "An efficient ant colony algorithm based on wake-vortex modeling method for aircraft scheduling problem". In: *Journal of Computational and Applied Mathematics* 317 (2017), pp. 157–170.

[9] Rakesh Prakash, Rajesh Piplani, and Jitamitra Desai. "An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput". In: *Transportation Research Part C: Emerging Technologies* 95 (2018), pp. 570–581.

[10] Meriem Ben Messaoud. "A thorough review of aircraft landing operation from practical and theoretical standpoints at an airport which may include a single or multiple runways". In: *Applied Soft Computing* 98 (2021), p. 106853.

[11] Sana Ikli, Catherine Mancel, Marcel Mongeau, Xavier Olive, and Emmanuel Rachelson. "The aircraft runway scheduling problem: A survey". In: *Computers & Operations Research* 132 (2021), p. 105336.

[12] Junfeng Zhang, Zihan Peng, Chunwei Yang, and Bin Wang. "Data-driven flight time prediction for arrival aircraft within the terminal area". In: *IET Intelligent Transport Systems* 16.2 (2022), pp. 263–275.

[13] Yan Glina, Richard Jordan, and Mariya Ishutkina. "A tree-based ensemble method for the prediction and uncertainty quantification of aircraft landing times". In: *American Meteorological Society–10th Conference on Aritificial Intelligence Applications to Environmental Science, New Orleans, LA*. 2012.

[14] Christian Strottmann Kern, Ivo Paixao de Medeiros, and Takashi Yoneyama. "Data-driven aircraft estimated time of arrival prediction". In: *2015 annual IEEE systems conference (syscon) proceedings*. IEEE. 2015, pp. 727–733.

[15] Samet Ayhan, Pablo Costas, and Hanan Samet. "Predicting estimated time of arrival for commercial flights". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 33–42.

[16] Gabor Takacs. "Predicting flight arrival times with a multistage model". In: *2014 IEEE International Conference on Big Data (Big Data)*. IEEE. 2014, pp. 78–84.

[17] Yan Ma, Wenbo Du, Jun Chen, Yu Zhang, Yisheng Lv, and Xianbin Cao. "A spatiotemporal neural network model for estimated-time-of-arrival prediction of flights in a terminal maneuvering area". In: *IEEE Intelligent Transportation Systems Magazine* 15.1 (2022), pp. 285–299.

[18] Jorge Silvestre, Miguel A Martínez-Prieto, Anibal Bregon, and Pedro C Álvarez-Esteban. "A deep learning-based approach for predicting in-flight estimated time of arrival". In: *The Journal of Supercomputing* (2024), pp. 1–35.

[19] Go Nam Lui, Chris HC Nguyen, Ka Yiu Hui, Kai Kwong Hon, and Rhea P Liem. "Enhancing aircraft arrival transit time prediction: A two-stage gradient boosting approach with weather and trajectory features". In: *Journal of the Air Transport Research Society* 4 (2025), p. 100062.

[20] Zhuoming Du, Junfeng Zhang, and Bo Kang. "A Data-Driven Method for Arrival Sequencing and Scheduling Problem". In: *Aerospace* 10.1 (2023), p. 62.

[21] Yutian Pang, Peng Zhao, Jueming Hu, and Yongming Liu. "Machine learning-enhanced aircraft landing scheduling under uncertainties". In: *Transportation Research Part C: Emerging Technologies* 158 (2024), p. 104444.

[22] Adam N Elmachtoub and Paul Grigas. "Smart "predict, then optimize"". In: *Management Science* 68.1 (2022), pp. 9–26.

[23] Ke Li and Jitendra Malik. "Learning to optimize". In: *arXiv preprint arXiv:1606.01885* (2016).

[24] Bo Tang and Elias B Khalil. "PyEPO: a PyTorch-based end-to-end predict-then-optimize library for linear and integer programming". In: *Mathematical Programming Computation* (July 2024). ISSN: 1867-2957. DOI: 10.1007/s12532-024-00255-x.

[25] Andrew J Cook and Graham Tanner. "European airline delay cost reference values". In: (2011).

[26] Zhengyi Wang, Man Liang, and Daniel Delahaye. "A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area". In: *Transportation Research Part C: Emerging Technologies* 95 (2018), pp. 280–294. DOI: https://doi.org/10.1016/j.trc.2018.07.019.

[27] Go Nam Lui, Thierry Klein, and Rhea P Liem. "Data-driven approach for aircraft arrival flow investigation at terminal maneuvering area". In: *AIAA Aviation Forum*. 2020, p. 2869.

[28] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. "Bringing up OpenSky: A large-scale ADS-B sensor network for research". In: *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE. 2014, pp. 83–94.

[29] EUROCONTROL. *Algorithm to describe weather conditions at European airports*. 2011. URL: https://www.eurocontrol.int/sites/default/files/publication/files/algorithm-met-technical-note.pdf.

[30] Go Nam Lui, Kai Kwong Hon, and Rhea P Liem. "Weather impact quantification on airport arrival on-time performance through a Bayesian statistics modeling approach". In: *Transportation Research Part C: Emerging Technologies* 143 (2022), p. 103811.

[31] Xinyi Hu, Jasper CH Lee, and Jimmy HM Lee. "Predict+ Optimize for packing and covering LPs with unknown parameters in constraints". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 4. 2023, pp. 3987–3995.

[32] Ronald Lewis Graham, Eugene Leighton Lawler, Jan Karel Lenstra, and AHG Rinnooy Kan. "Optimization and approximation in deterministic sequencing and scheduling: a survey". In: *Annals of discrete mathematics*. Vol. 5. Elsevier, 1979, pp. 287–326.

[33] Lucio Bianco, Aristide Mingozzi, and Salvatore Ricciardelli. "The traveling salesman problem with cumulative costs". In: *Networks* 23.2 (1993), pp. 81–91.