aerospace

MDPI

Article

A Decision Support Framework for Aircraft Arrival Scheduling and Trajectory Optimization in Terminal Maneuvering Areas

Dongdong Gui [1], Meilong Le [1,*], Zhouchun Huang [2,*] and Andrea D'Ariano [3]

[1] College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[2] College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[3] Department of Civil, Computer Science and Aeronautical Technologies Engineering, Roma Tre University, 00145 Rome, Italy; andrea.dariano@uniroma3.it
* Correspondence: mlle@nuaa.edu.cn (M.L.); zhouchun.huang@nuaa.edu.cn (Z.H.)

Abstract: This study introduces a decision support framework that integrates aircraft trajectory optimization and arrival scheduling to facilitate efficient management of descent operations for arriving aircraft within terminal maneuvering areas. The framework comprises three modules designed to tackle specific challenges in the descent process. The first module formulates and solves a trajectory optimization problem, generating a range of candidate descent trajectories for each arriving aircraft. The options for descent operations include step-down descent operation, Continuous Descent Operation (CDO), and CDO with a lateral path stretching strategy. The second module addresses the assignment of conflict-free trajectories to aircraft, determining precise arrival times at each waypoint. This is achieved by solving an aircraft arrival scheduling problem. To overcome computational complexities, a novel variable neighborhood search algorithm is proposed as the solution approach. This algorithm utilizes three neighborhood structures within an extended relaxing and solving framework, and incorporates a tabu search algorithm to enhance the efficiency of the search process in the solution space. The third module focuses on comparing the total cost incurred from flight delays and fuel consumption across the three descent operations, enabling the selection of the most suitable operation for the descent process. The decision support framework is evaluated using real air traffic data from Guangzhou Baiyun International Airport. Experimental results demonstrate that the framework effectively supports air traffic controllers by scheduling more cost-efficient descent operations for arrival aircraft.

Keywords: decision support framework; descent operations; aircraft arrival scheduling; trajectory optimization; variable neighborhood search

Citation: Gui, D.; Le, M.; Huang, Z.; D'Ariano, A. A Decision Support Framework for Aircraft Arrival Scheduling and Trajectory Optimization in Terminal Maneuvering Areas. *Aerospace* **2024**, *11*, 405. https://doi.org/10.3390/aerospace11050405

Academic Editor: Eri Itoh

Received: 2 April 2024
Revised: 14 May 2024
Accepted: 15 May 2024
Published: 16 May 2024

1. Introduction

Air transport demand has been growing rapidly in recent decades. Despite the significant impact of the COVID-19 outbreak on air transportation, high levels of air traffic are still projected in the near future [1,2]. It is estimated that the demand will surge by 50% within the next 17 years, accompanied by a corresponding increase in delay frequencies estimated to be 15 times greater than the current levels [3]. Short-term infrastructure development for capacity expansion is often limited by financial, geopolitical, and ecological factors. Therefore, optimizing the efficient use of existing infrastructure and managing air traffic within airport Terminal Maneuvering Areas (TMAs) play a significant role in mitigating airport congestion. Additionally, fuel efficiency is a top priority for airlines as fuel consumption costs constitute a major expense for them [4]. Furthermore, the environmental and health impacts of air traffic have prompted increased scrutiny from both the government and the public, primarily due to concerns related to noise and emissions. Consequently, aviation authorities continuously set ambitious targets to address various aspects of Air Traffic Management (ATM), including increasing airport capacity, minimizing

aircraft delays, and reducing the environmental impact of air traffic while adhering to ATM safety regulations [5,6].

However, it is worth noting that some of these targets may conflict with each other when put into practice. One such example is the objective of minimizing aircraft delays, which may require increasing aircraft speed and cruising altitude to avoid flight conflicts in congested air segments. Unfortunately, this can result in higher fuel consumption and increased emissions. Furthermore, Air Traffic Controllers (ATCs) currently have limited support from automated systems, which hinders their ability to make well-informed decisions [7]. At present, ATCs primarily rely on their experience and tactical instructions, such as heading vectors, speed adjustments, and path stretching, to ensure safe separation between aircraft and maximize the throughput within the TMA. This situation is primarily attributed to the lack of granularity and efficiency in aircraft trajectory management and scheduling techniques.

Trajectory Based Operations (TBO) is an advanced concept in ATM that has been extensively explored by initiatives such as SESAR [8] and NextGen [9]. The primary objective of TBO is to transition the current air traffic control system into a collaborative one that enables precise coordination of aircraft operations and achieves higher accuracy in all dimensions. By efficiently negotiating trajectories within TBO, both airborne and ground equipment can be synchronized, allowing for the establishment of a descent plan with prescribed landing times before an aircraft enters the TMA. This approach ensures equitable and advantageous trajectories for aircraft. The management of aircraft landings within the TMA using TBO involves several stages. Firstly, the Flight Management System (FMS) on board the aircraft calculates a set of trajectories based on its descent operation, including the travel time for each air segment on the assigned landing route. This information is then transmitted to the ground system. Subsequently, the ground system sequences the aircraft and examines for any potential conflicts between trajectories. Finally, ATCs plan the descent operation, taking into account the appropriate decision criteria to ensure a satisfactory outcome.

Within the framework of TBO, the Continuous Descent Operation (CDO) has gained recognition as a vital operational technique. This procedure involves guiding arrival aircraft through an optimized profile descent using low engine thrust settings and a low drag configuration. Unlike the conventional Step-down Descent Operation (SDO), CDO aircraft descend continuously from the Top of Descent (TOD), aiming to avoid level segments. This approach leads to reduced fuel consumption and pollutant emissions [5]. Nevertheless, managing the separation between consecutive aircraft during the CDO process can pose challenges for ATCs as they need to closely monitor aircraft trajectories and increase separation buffers to avoid flight conflicts. As a result, the capacity of both airspace and runways in the TMA can be negatively affected, particularly during peak hours [10]. In our previous work [11], we developed a small proof of concept to demonstrate the successful implementation of a tactical lateral path stretching strategy in conjunction with the CDO for arrival aircraft in high-density traffic. This approach, termed ps-CDO (CDO with a lateral path stretching strategy), provides alternative lateral paths to each arrival aircraft, offering greater flexibility in avoiding conflicts within the TMA. By providing multiple options for the lateral path of each aircraft during the CDO phase, considering different arrival times, an arrival aircraft scheduling problem can be formulated and systematically resolved. This enables effective sequencing of the aircraft and prevents potential conflicts.

In this paper, we present a decision support framework that builds upon our previous work on collaborative aircraft trajectory optimization and arrival scheduling. The primary objective of this framework is to assist ATCs in effectively managing descent operations for arrival traffic within the TMA, while concurrently optimizing the utilization of the TMA's capacity. Our proposed framework consists of three synthesis modules: descent trajectory generation, aircraft arrival scheduling, and optimal trajectory selection. These modules specifically focus on three distinct descent operations: the SDO, the CDO, and the ps-CDO. To elaborate, the first module formulates and solves an Aircraft Trajectory Optimization

Problem (ATOP), which generates descent trajectories for each arriving aircraft considering all three descent operations. These descent trajectories define the time windows for each air segment across all available descent routes. Based on the generated descent trajectories, the second module assigns each aircraft a conflict-free trajectory along with the corresponding descent route, resulting in the determination of precise arrival times at each waypoint. This process involves solving a trajectory-based Aircraft Arrival Scheduling Problem (AASP) formulated as a Mixed Integer Programming (MIP). To find solutions, we employ a Variable Neighborhood Search (VNS) algorithm. Lastly, the third module determines the optimal descent operation selection among the three options by considering both total delay and total fuel consumption costs. This selection process aims to strike a balance between minimizing overall delays and reducing fuel consumption.

1.1. Literature Review

The aviation industry has developed several decision support tools to assist ATCs in managing arriving flows in the TMA, such as the Center-TRACON Automation Systems [12] and the Arrival Manager systems [13]. These systems primarily offer landing aircraft with arrival time decisions and runway assignments while aiding the ATCs in guiding aircraft to fly along specific waypoints along the established air route. These aforementioned decision support tools, however, have difficulty providing more complex trajectory management solutions, which may result in an increase in the ATCs' workload. Developments in communication, navigation, and surveillance technologies have contributed to increasing trajectory accuracy, leading to expectations for a more advanced decision support tool based on trajectories. The tool attempts to sequence and merge arrival flow by performing time and energy management operations on descending trajectories.

The efficient management of aircraft descent trajectories is crucial for reducing fuel consumption, gas emissions, and noise during aircraft operations. Although effective trajectory management during aircraft descent is successful in airports with orderly and steady traffic, it is challenging in heavy congestion traffic situations. Recent studies have focused mainly on trajectory optimization technologies to address the challenges of managing aircraft descent trajectories in congested traffic situations. Itoh et al. [14] and Callantine et al. [15] investigate the application of flight-deck interval management for aircraft descent trajectories in a TMA. Turgut et al. [16] introduce a fixed flight path angle descent (FFPAD) technology to help management aircraft descent trajectories in a congested traffic environment. The feasibility of managing the arrival aircraft trajectories in a busy TMA with FFPAD is fully investigated in [17]. Sun et al. [18] use the four-dimensional waypoints technology to achieve a conflict-free trajectory in high-density airspace, and evaluate the feasibility of operations with a leader-follower example. Nonetheless, the aforementioned methods are only practical for particular operational equipment and aircraft and limited in terms of broader practical applications.

On the other hand, an effective trajectory management solution based on the TBO concept involves sequencing and merging arrival traffic by assigning required times of arrival (RTAs) at one or several fixes along a predetermined route. Pawelek et al. [19] propose a two-step approach to implement the full CDOs with RTA in a TMA. First, they formulated and solved a trajectory optimization problem to generate a set of candidate CDO trajectories with times of arrival at a specific metering fix for arrival aircraft. Next, they built an aircraft scheduling model to optimally assign RTAs to the arriving traffic, ensuring safe time separation. As a result, aircrew can fly optimal descent trajectories using the FMS while meeting the required times of arrival. However, the feasibility of this approach was only evaluated under a low air traffic environment. Sáez et al. [10] develop a four-dimensional trajectory negotiation and synchronization process to complete the CDO procedure with a path stretching strategy known as the tromboning paradigm in dense TMAs with the same approach. They take into account assigning optimal routes of the trombone and RTAs to the arriving traffic simultaneously. Nevertheless, the aircraft scheduling model used by [10,19] was established with only a certain metering fix, which

did not include all descent waypoints in the TMA. In addition, Samà et al. [20] propose an integrated approach for aircraft trajectory optimization and scheduling for both arrival and departure aircraft, and their method involves having arrival aircraft descend using the SDO and determining optimal trajectories and sequencing for all aircraft simultaneously to demonstrate its benefits. Yet they fail to account for the scenario where an aircraft could have multiple descent routes, nor do they apply the CDO procedure.

The classical aircraft arrival scheduling problem aims to allocate a runway to each aircraft for landing and schedule a corresponding landing time for each. The classical aircraft arrival scheduling problem is commonly referred to as the Runway Scheduling Problem (RSP). The RSP is typically formulated and solved as a job-shop scheduling model [21]. Bennell et al. [22] conducted an extensive literature review of the RSP, while a more recent comprehensive survey of relevant studies was made by [23]. There are a number of algorithms that have been developed to solve the RSP since it has been proven to be a typical NP-hard problem [24]. The particular nature of the RSP prompts researchers to compute high-quality solutions with low computational time. Most of the research thus favors heuristic and meta-heuristic algorithms. The commonly used meta-heuristic algorithms are the genetic algorithm [25–27], the simulated annealing [28,29], and the ant colony optimization [30,31]. Additionally, the Variable Neighborhood Descent (VND), which belongs to the family of VNS, has received much attention as well. Salehipour et al. (2009) [32] develop VND to address the RSP, wherein four neighborhood structures were designed to improve the incumbent solutions. Salehipour et al. (2013) [33] consider the same problem as [32] and use VND in a hybrid algorithm by applying the simulated annealing framework. Furthermore, VND was also used in [34,35] to address the RSP. Recently, Salehipour and Ahmadian [36] propose a novel hybrid optimization approach to tackle the RSP. This approach can be decomposed into two steps. Firstly, an initial sequence is generated using the target landing time first rule. Then, the sequence is improved by a local search algorithm. With the sequence fixed, the runway scheduling problem is formed as a linear program, and the landing time is obtained by solving the linear program with the solver CPLEX. Ahmadian and Salehipour [37] extend this research by introducing a novel framework called "Relax-and-Solve" (R&S) as the sequence local search. They called it "Relax-and-Solve" because it improves the aircraft sequence at the runway by iteratively destructing (relaxing) a sub-sequence of the current sequence and reconstructing (solving) a feasible sequence using optimization techniques. In this paper, we incorporate the R&S framework into our VNS algorithm to address the AASP.

In a more realistic context, Bianco et al. [38] take into account the restrictions of various TMA resources for aircraft arrival scheduling, such as holding circles and air segments in addition to runways. Specifically, this model is formulated as a no-wait job shop scheduling problem, where the runways and air segments are interpreted as the machines, and the aircraft to be scheduled are the jobs. The separation time between each pair of aircraft corresponds to the sequence-dependent set-up time, and the earliest and latest landing time are interpreted as the release date and due date, respectively. Solving the resulting model is expected to yield the start time of each aircraft at every resource along its descent route, such that the aircraft have conflict-free descent trajectories satisfying all restrictions. Based on this effort, D'Ariano et al. [39] formulate an aircraft scheduling problem with fixed routes, which is represented by an alternative graph. Refs. [40–43] extend the work by including more assumptions and objectives to improve the level of accuracy in modeling the aircraft scheduling problem with fixed routes. Our previous work [11] covers similar ground. Samà et al. (2014) [44] propose a new formulation that considers alternative routes for each aircraft. In order to solve the aircraft scheduling problem with alternative routes, Samà et al. (2017) [45] present a VNS algorithm. They employ five distinct neighborhood structures by capturing the characteristic of the objective function to improve the search for executing the routes for each aircraft. The AGLIBRARY solver, a state-of-the-art optimization solver designed to deal with complex routing and scheduling problems, is then used to solve the problem with the searched routes. Our approach differs

from [45] in both modeling and solution approaches. Specifically, our AASP formulation aims to minimize a linear combination of the total delay and the total difference between the scheduled landing time and the minimum fuel-burning landing time. This contrasts with their study, which primarily focuses on minimizing the maximum delay, with the VNS implementation adhering to this objective as well. In addition, our approach encompasses three descent operations, modeled with different equations of motion for the trajectory optimization, whereas [45] considers only the step-down descent procedure. The algorithm proposed by [45] is not applicable in our framework due to the different models used, and we thus develop a new VNS algorithm that employs an extended R&S framework to establish neighborhood structures during the search for improved routes for each aircraft.

1.2. Our Contributions

This paper introduces a novel decision support framework for managing aircraft descent operations in the TMA and makes several contributions to the existing literature:

(1) We develop a comprehensive decision support framework that can assist ATCs in determining the optimal descent operations and arrival schedules for approaching aircraft in TMAs. To the best of our knowledge, this is the first approach that integrates the aircraft trajectory optimization and arrival scheduling and incorporates various options for selecting descent operations, including SDO, CDO, and ps-CDO. By extensively comparing three different descent operations and their impacts on flight delays and fuel consumption in various traffic flows, our proposed tool can provide its users with an important reference for selecting the most suitable operations for arrival aircraft.

(2) We propose a novel VNS algorithm that employs three neighborhood structures to enhance the efficiency and quality of solutions for solving the AASP. This algorithm leverages an extended R&S framework to establish the neighborhood structures, and incorporates a Tabu Search (TS) algorithm to efficiently search the solution space.

(3) We implement extensive experiments using real traffic data from Guangzhou Baiyun International Airport (GBIA), considering both arrival and departure traffic flows. The computational results demonstrate the effectiveness and efficiency of our proposed decision support framework and our proposed solution approach.

1.3. Organization of This Paper

The remainder of this paper is organized as follows: Section 2 provides an overview of the decision support framework's architecture and describes the mathematical models used in the framework. Section 3 discusses the solution methodologies employed for addressing the problem. The numerical experiments conducted and the corresponding results are presented in Section 4. Finally, Section 5 offers a conclusion summarizing the findings of this study and suggests potential directions for future research.

2. Decision Support Framework

In this section, we present our proposed decision support framework, starting with the introduction of the concept of operations proposed in this work (Section 2.1). We then provide a detailed description of the mathematical models proposed for the three modules within the decision support framework (Sections 2.2–2.4).

2.1. Concept of Operations

In Figure 1, we provide a simplified scenario that serves as an illustration of the problem addressed in this paper. The scenario considers a time-limited decision horizon where multiple aircraft seek to enter an extended-TMA (E-TMA) from different directions through various entry points. The ATCs are responsible for scheduling the descent process for each aircraft.

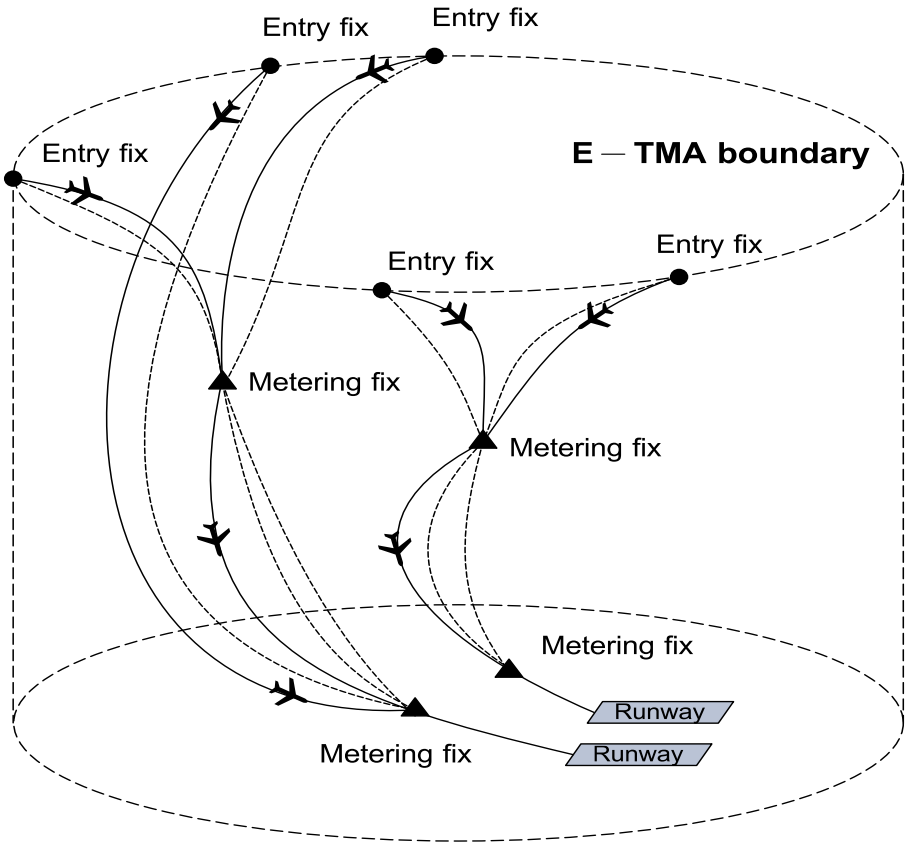


Figure 1. A simplified scenario for our studied problem.

For each approach, there are predefined descent routes available, and the scheduling process aims to assign a suitable descent route to each aircraft along with a scheduled arrival time at each waypoint. This ensures that every aircraft follows a conflict-free descent trajectory through the different air segments and eventually lands on a runway. The traffic flows may converge at various metering waypoints, where the aircraft trajectories merge until they ultimately reach the runways.

Well before the TOD point, while still in the cruise phase, the ATC requires the aircraft to compute its travel time window for each air segment. The travel time window is determined by calculating the difference between the earliest and latest possible times for flying through the air segments, corresponding to the earliest and latest descent trajectories. The FMS is responsible for calculating these trajectories, taking into account various factors such as aircraft performance, flight envelope, and weather conditions. The width of the travel time window can be influenced by the choice of descent operation employed by the aircraft. In this paper, we consider three descent operations: the SDO, the CDO, and the ps-CDO.

Figure 2a illustrates the vertical profiles of the earliest and latest trajectories for the three descent operations. It also presents the travel time window for each descent operation on an air segment. It can be observed that the SDO offers a higher degree of operational flexibility, resulting in a wider time window. However, this flexibility comes at the cost of increased fuel consumption. On the other hand, the ps-CDO extends the width of the CDO time window through a lateral path stretching strategy, which may also lead to increased fuel consumption. The CDO is the most restrictive descent operation, with the narrowest time window. However, it is also the most fuel-efficient descent operation.

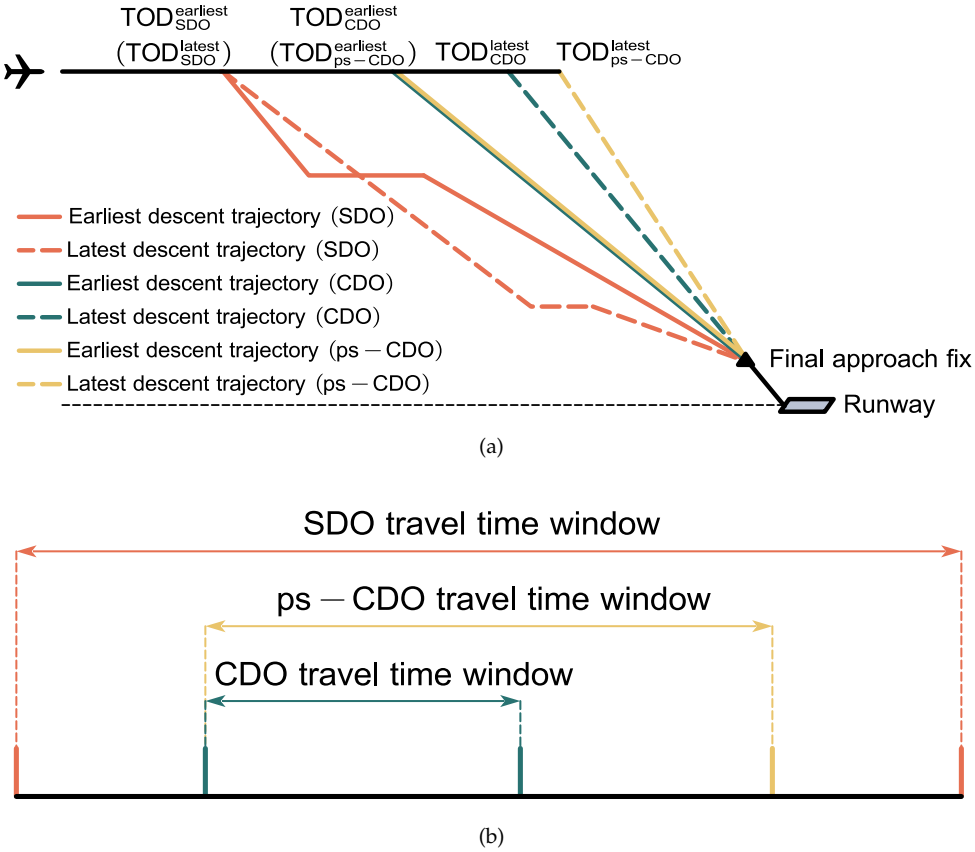


Figure 2. A schematic diagram of the earliest and latest trajectories for three descent operations: (**a**) Vertical profile of descent trajectories. (**b**) Travel time window in an air segment.

After entering the E-TMA during the cruising phase, the aircraft engages in a trajectory and scheduled time negotiation process with the ATCs. In order to facilitate this synchronization process, we propose a decision support framework, which is illustrated in Figure 3. The framework consists of three modules, outlined as follows:

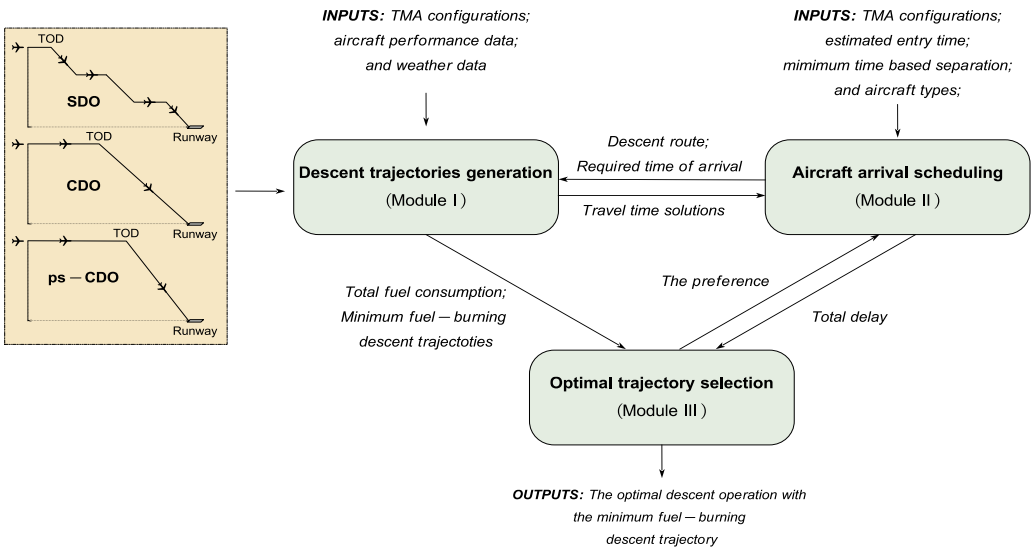


Figure 3. An overview of the proposed decision support framework.

- Module I: Descent trajectories generation.
  This module is responsible for generating descent trajectories for arriving aircraft, considering predefined descent operations such as the SDO, the CDO, and the ps-CDO. These trajectories encompass the earliest and latest descent trajectories, as well as the minimum fuel-burning descent trajectory. The travel time window is determined based on the earliest and latest descent trajectories, while the minimum fuel-burning landing time is associated with the minimum fuel-burning descent trajectory. To achieve this, an ATOP is formulated and solved, taking into account factors such as aircraft performance, flight envelope, and weather conditions (refer to Section 2.2). It is important to note that multiple descent routes are considered for each aircraft, and the descent trajectories differ for each route. In practice, these trajectories would be generated using advanced functionality in the FMS for each aircraft.
- Module II: Aircraft arrival scheduling.
  In the second module, based on the computed descent trajectories from the previous module, a descent route with a conflict-free trajectory is assigned to each aircraft operating under one of the three descent operations. This assignment process ensures a safe separation between aircraft during their descent procedure and determines the exact arrival time at each waypoint. The objective of this model is to minimize a linear combination of the total delay and the total difference between the minimum fuel-burning landing time and the scheduled landing time. The preferences for the total delay and the total difference are typically provided by the decision makers, in this case, the ATCs in our studied system. The AASP is formulated as a MIP (see Section 2.3) and a VNS algorithm is developed (see Section 3.2) to solve the problem.
- Module III: Optimal trajectory selection.
  This module is initiated by the decision makers (i.e., the ATCs) who choose the optimal descent operation for every arrival aircraft within the decision time horizon. We define a cost function as a linear combination of the total delay cost and total fuel consumption cost to determine the priority among the three descent operations (see Section 2.4). In this model, the total delay and the required landing time for all arriving aircraft can be calculated by solving the AASP in Module II. Additionally, the ATOP in Module I can be utilized to compute the minimum fuel-burning descent trajectory that satisfies the required landing time, from which the resulting fuel consumption can be determined. Consequently, the approaching aircraft will follow the optimal descent operation with the minimum fuel-burning descent trajectories that meet the required landing time. Notably, in the AASP, if decision makers increase the weight of the total delay indicator in the objective function to achieve a smaller total delay, it will inevitably lead to an increase in the total difference between the minimum fuel-burning landing time and the scheduled landing time. This increased difference signifies a greater deviation from the minimum fuel-burning descent trajectory, thereby leading to higher fuel consumption for each aircraft. Conversely, decreasing the weight of the total delay indicator will have the opposite effect. This approach allows for the selection of the most suitable descent operation based on the cost function and the specific requirements of each aircraft.

Without loss of generosity, we make the following assumptions to expedite the discussion of our proposed framework: (1) All waypoints are predefined in the descent procedure for each arriving aircraft. (2) There are no operational failures during descent, such as missed approaches, emergency landings, or engine failures, and operating time is not taken into account. (3) The aircraft maintains its original cruise speed until it reaches the TOD. (4) Only one aircraft is allowed to land or take-off on each runway at a time.

2.2. Descent Trajectories Generation

This section presents the methodology employed to generate aircraft descent trajectories as they approach their destination. The descent trajectories consist of two segments: the cruise segment from an entry waypoint to the TOD, and the descent segment from the

TOD to the runway. The descent segment is further divided into multiple phases, which correspond to different changes in aircraft dynamics. To optimize the aircraft's vertical profile, including altitude and speed, while maintaining a fixed lateral route and distance, we formulate the problem as a multi-phase optimal control problem. This formulation takes into consideration altitude and speed constraints, as well as aircraft performance bounds. In this study, we utilize a point mass model [46] to describe the aircraft's equations of motion. The trajectory of each arriving aircraft along a pre-designed descent route is planned such that the aircraft follows the point mass equations in each air segment $p \in \mathscr{P} = \{1, 2, \cdots, |\mathscr{P}|\}$ and satisfies boundary conditions between any two consecutive air segments. The ATOP mathematical formulation is presented below.

$$
\begin{aligned}
(\textbf{ATOP}): \quad \min \quad & J && \text{(1a)}\\
\text{s.t.} \quad & \dot{V}_T^{(p)} = \frac{T^{(p)} - D^{(p)}(V_T, h, \beta)}{m} - g\gamma^{(p)} \quad \forall p \in \mathscr{P} && \text{(1b)}\\
& \dot{s}^{(p)} = V_T^{(p)} \cos\gamma^{(p)} + W_s^{(p)}(h) \quad \forall p \in \mathscr{P} && \text{(1c)}\\
& \dot{h}^{(p)} = V_T^{(p)} \sin\gamma^{(p)} \quad \forall p \in \mathscr{P} && \text{(1d)}\\
& \dot{h}_{\min}^{(p)} \le \frac{dh}{dt} \le \dot{h}_{\max}^{(p)} \quad \forall p \in \mathscr{P} && \text{(1e)}\\
& T_{\min}^{(p)} \le T^{(p)} \le T_{\max}^{(p)} \quad \forall p \in \mathscr{P} && \text{(1f)}\\
& 0 \le \beta^{(p)} \le \beta_{\max}^{(p)} \quad \forall p \in \mathscr{P} && \text{(1g)}\\
& \gamma_{\min}^{(p)} \le \gamma^{(p)} \le 0 \quad \forall p \in \mathscr{P} && \text{(1h)}\\
& V_{CAS,\min}^{(p)} \le V_{CAS}^{(p)} \le V_{CAS,\max}^{(p)} \quad \forall p \in \mathscr{P} && \text{(1i)}\\
& M_{\min}^{(p)} \le M^{(p)} \le M_{\max}^{(p)} \quad \forall p \in \mathscr{P} && \text{(1j)}\\
& [V_T, h]^\intercal(t_0^{(1)}) = [V_{T0}, h_0]^\intercal && \text{(1k)}\\
& [V_T, s, h]^\intercal(t_f^{(|\mathscr{P}|)}) = [V_{Tf}, s_f, h_f]^\intercal && \text{(1l)}\\
& [V_T, s, h]^\intercal(t_f^{(p-1)}) = [V_T, s, h]^\intercal(t_0^{(p)}) \quad \forall p \in \mathscr{P} \setminus \{1\} && \text{(1m)}
\end{aligned}
$$

For each air segment p, the variables have the following definitions: $V_T^{(p)}$ is the true airspeed; $s^{(p)}$ is the along-track distance from the runway; $h^{(p)}$ is the altitude; $\gamma^{(p)}$ is the aerodynamic flight path angle; $T^{(p)}$ and $D^{(p)}$ represent the thrust and the aerodynamic drag, respectively; $\beta^{(p)}$ is the speed-brakes deflection; $W_s^{(p)}(h)$ is the longitudinal wind; $V_{CAS}^{(p)}$ and $M^{(p)}$ are the calibrated airspeed and Mach number, respectively; $t_0^{(p)}$ and $t_f^{(p)}$ denote the initial and final time, respectively. The parameters g and m are the aircraft gravity acceleration and mass, respectively.

Constraints (1b)–(1d) represent the point-mass dynamics of an aircraft in the vertical plane, where $[V_T, s, h]^\intercal$ is the state vector and $[T, \beta, \gamma]^\intercal$ is the control vector. Constraints (1e)–(1j) specify the value range requirements of $\frac{dh}{dt}$, T, β, γ, V_{CAS} and M to ensure flight envelope protection and passenger comfort, where $\dot{h}_{\min}$ and $\dot{h}_{\max}$ are the minimum and maximum descent rate; $T_{\min}$ and $T_{\max}$ are the idle and maximum thrust; $\beta_{\max}$ is the maximum speed-brakes deflection; $\gamma_{\min}$ is the minimum descent gradient; $V_{CAS,\min}$ and $V_{CAS,\max}$ are the minimum and maximum calibrated airspeed; $M_{\min}$ and $M_{\max}$ are the minimum and maximum Mach number. Constraints (1k) and (1l) represent the boundary condition at the TOD and the runway. Constraint (1m) is the link constraint.

Module I calculates the earliest and latest descent trajectories, as well as the minimum fuel-burning descent trajectory, for each arrival aircraft. Each trajectory serving its specific objectives:

Earliest descent trajectory, which is to minimize the flight time,

$$(\mathbf{ATOP_E}): \quad \min \quad J_{time} = \underbrace{\frac{s(t_0^{(1)}) - s_{\max}}{V_{cr}}}_{\text{cruise segment flight time}} + \underbrace{t_f^{(|\mathscr{P}|)}}_{\text{descent segment flight time}} \tag{2a}$$

$$\text{s.t.} \quad \text{(1b)–(1m)} \tag{2b}$$

where $s(t_0^{(1)})$ is an along-track distance at TOD; $s_{\max}$ is the along-track distance of entry waypoint; V_{cr} is the cruise speed.

Latest descent trajectory, which is to maximize the flight time:

$$(\mathbf{ATOP_L}): \quad \max \quad J_{time} \tag{3a}$$

$$\text{s.t.} \quad \text{(1b)–(1m)} \tag{3b}$$

Minimum fuel-burning descent trajectory, which is to minimize the fuel consumption:

$$(\mathbf{ATOP_F}): \quad \min \quad J_{fuel} = \underbrace{\frac{f_{cr}(s(t_0^{(1)}) - s_{\max})}{V_{cr}}}_{\text{cruise segment fuel consumption}} + \underbrace{\sum_{p\in\mathscr{P}} \int_{t_0^{(p)}}^{t_f^{(p)}} f_{idle}(V_T, h)dt}_{\text{descent segment fuel consumption}} \tag{4a}$$

$$\text{s.t.} \quad \text{(1b)–(1m)} \tag{4b}$$

where f_{cr} and f_{idle} are the cruise fuel flow rate and idle fuel flow rate in the descent, respectively.

The ATOP may encounter different operational constraints for each aircraft in each air segment depending on the chosen descent operation. The parameters related to aircraft performance, which are utilized in both the point mass model and the fuel burn function, are derived from the Base of Aircraft Data (BADA) [47]. It is important to note that the additional thrust and speed-brakes are only permitted for the SDO. For the CDO and the ps-CDO, the point-mass model is reduced to a so-called $\gamma - command$ model, which assumes continuous vertical equilibrium through the descent process. Specifically, in order to obtain an environmentally friendly CDO trajectory, the idle thrust is imposed and the use of speed brakes is not allowed throughout the descent. As a result, the flight path angle (γ) is the only control variable in this dynamical system to manage the energy of the aircraft and achieve different travel time through the descent air segments.

We solve the $(\mathbf{ATOP_E})$ formulation to obtain the earliest travel time for each aircraft in each air segment p, which is $t_f^{(p)} - t_0^{(p)}$. Similarly, solving the $(\mathbf{ATOP_L})$ formulation provides us with the aircraft's latest travel time in each air segment p. The minimum fuel-burning landing time (which is $t_f^{(|\mathscr{P}|)}$) for each aircraft is determined by solving the $(\mathbf{ATOP_F})$ formulation. These times are then utilized in Module II.

2.3. Aircraft Arrival Scheduling

In this section, we formulate and solve the AASP to assign a conflict-free trajectory with a descent route to each aircraft. This scheduling process aims to determine the precise arrival time of each aircraft at every waypoint, ensuring compliance with safety separation requirements. The AASP takes into account various inputs, including the travel time window for each air segment and the minimum fuel-burning landing time (both obtained from the ATOP). Additionally, estimated earliest and latest arrival times for aircraft entering the TMA and the estimated target landing time are considered. It is worth mentioning that the estimated target landing time remains consistent regardless of the chosen descent operation.

The AASP is formulated using a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ with a set of waypoints $\mathscr{V}$ and a set of arcs $\mathscr{A}$. For each aircraft $i \in \mathscr{F}$, a descent air route can be chosen in a set of alternative descent air routes $\mathscr{R}_i$, where $\mathscr{F}$ is the set of all arrival aircraft. Specifically,

a descent air route r_i for aircraft i contains a set of waypoints $(o_i, \cdots, k, l, \cdots, d_i) \in \mathscr{V}_i^{r_i}$. The origin o_i and the destination d_i represent the entry waypoint and the runway, respectively. Each arc $(k,l) \in \mathscr{A}_i^{r_i}$ indicates the air segment between two adjacent waypoints in the directed graph. Therefore, the set of waypoints that can be used by aircraft i is $\mathscr{V}_i = \bigcup_{r_i \in \mathscr{R}_i} \mathscr{V}_i^{r_i}$. Similarly, the set of arcs that can be used by aircraft i is $\mathscr{A}_i = \bigcup_{r_i \in \mathscr{R}_i} \mathscr{A}_i^{r_i}$. In this connection, $\mathscr{V}_i \subset \mathscr{V}$, $\mathscr{A}_i \subset \mathscr{A}$ in the digraph $\mathscr{G}$.

We introduce two sets of binary decision variables to respectively model the descent air route selection and the sequential relationship between aircraft at shared waypoints. The variable $y_i^{r_i}$ represents the selection of descent air route r_i for aircraft i. The variable z_{iju} indicates the sequential relationship between aircraft i and j as they traverse the same waypoint $u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j}$. This is equal to 1 if aircraft i arrives at waypoint u before aircraft j and 0 otherwise. We use the continuous variable $t_{ik}^{r_i}$ to represent the arrival time at waypoint k for aircraft i following descent air route r_i.

The objective of the AASP is to minimize a linear combination of the total delay and the total difference between the scheduled landing time and the minimum fuel-burning landing time. The sets, parameters, and decision variables used in the formulation of the AASP are summarized in Table 1. The AASP is formulated as follows:

$$
\begin{aligned}
(\mathbf{AASP}): \quad \min \quad & \lambda \sum_{i \in \mathscr{F}} \alpha_i + (1-\lambda) \sum_{i \in \mathscr{F}} \mu_i && \text{(5a)}\\
\text{s.t.} \quad & \sum_{r_i \in \mathscr{R}_i} y_i^{r_i} = 1 \quad \forall i \in \mathscr{F} && \text{(5b)}\\
& t_{ik}^{r_i} \le B y_i^{r_i} \quad \forall i \in \mathscr{F}, \forall k \in \mathscr{V}_i^{r_i}, \forall r_i \in \mathscr{R}_i && \text{(5c)}\\
& \Gamma_{io_i}^{\text{early}} y_i^{r_i} \le t_{io_i}^{r_i} \le \Gamma_{io_i}^{\text{late}} y_i^{r_i} \quad \forall i \in \mathscr{F}, \forall r_i \in \mathscr{R}_i && \text{(5d)}\\
& \underline{\Theta}_{ik\_il}^{r_i} y_i^{r_i} \le t_{il}^{r_i} - t_{ik}^{r_i} \le \overline{\Theta}_{ik\_il}^{r_i} y_i^{r_i} \\
& \quad \forall i \in \mathscr{F}, \forall (k,l) \in \mathscr{A}_i^{r_i}, \forall r_i \in \mathscr{R}_i && \text{(5e)}\\
& t_{ju}^{r_j} \ge t_{iu}^{r_i} + S_{iju} - B(3 - z_{iju} - y_i^{r_i} - y_j^{r_j}) \\
& \quad \forall i,j \in \mathscr{F}, i<j, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j}, \forall r_i \in \mathscr{R}_i, \forall r_j \in \mathscr{R}_j && \text{(5f)}\\
& t_{iu}^{r_i} \ge t_{ju}^{r_j} + S_{jiu} - B(2 + z_{iju} - y_i^{r_i} - y_j^{r_j}) \\
& \quad \forall i,j \in \mathscr{F}, i<j, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j}, \forall r_i \in \mathscr{R}_i, \forall r_j \in \mathscr{R}_j && \text{(5g)}\\
& z_{iju} - z_{ijv} \ge \sum_{\substack{r_i \in \mathscr{R}_i \\ (u,v) \in \mathscr{A}_i^{r_i}}} y_i^{r_i} + \sum_{\substack{r_j \in \mathscr{R}_j \\ (u,v) \in \mathscr{A}_j^{r_j}}} y_j^{r_j} - 2 \\
& \quad \forall i,j \in \mathscr{F}, i<j, \forall (u,v) \in \mathscr{A}_i^{r_i} \cap \mathscr{A}_j^{r_j} && \text{(5h)}\\
& z_{ijv} - z_{iju} \ge \sum_{\substack{r_i \in \mathscr{R}_i \\ (u,v) \in \mathscr{A}_i^{r_i}}} y_i^{r_i} + \sum_{\substack{r_j \in \mathscr{R}_j \\ (u,v) \in \mathscr{A}_j^{r_j}}} y_j^{r_j} - 2 \\
& \quad \forall i,j \in \mathscr{F}, i<j, \forall (u,v) \in \mathscr{A}_i^{r_i} \cap \mathscr{A}_j^{r_j} && \text{(5i)}\\
& \alpha_i \ge t_{id_i}^{r_i} - \Gamma_{id_i} \quad \forall i \in \mathscr{F}, \forall r_i \in \mathscr{R}_i && \text{(5j)}\\
& \alpha_i \ge 0 \quad \forall i \in \mathscr{F} && \text{(5k)}\\
& \mu_i \ge t_{id_i}^{r_i} - \Gamma_{id_i}^{\text{fuel}} \quad \forall i \in \mathscr{F}, \forall r_i \in \mathscr{R}_i && \text{(5l)}\\
& \mu_i \ge \Gamma_{id_i}^{\text{fuel}} - t_{id_i}^{r_i} \quad \forall i \in \mathscr{F}, \forall r_i \in \mathscr{R}_i && \text{(5m)}\\
& t_{ik}^{r_i} \ge 0 \quad \forall i \in \mathscr{F}, \forall k \in \mathscr{V}_i^{r_i}, \forall r_i \in \mathscr{R}_i && \text{(5n)}\\
& z_{iju} \in \{0,1\} \quad \forall i,j \in \mathscr{F}, i<j, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j} && \text{(5o)}\\
& y_i^{r_i} \in \{0,1\} \quad \forall i \in \mathscr{F}, \forall r_i \in \mathscr{R}_i && \text{(5p)}
\end{aligned}
$$

where $\lambda \in [0,1]$ is a weight parameter that allows for the adjustment of the relative importance of total delay and total difference of landing times. The variable α_i denotes the delay of aircraft i with respect to its reference landing time Γ_{id_i}. The variable μ_i denotes the absolute value of the difference $|t^{r_i}_{id_i} - \Gamma^{\text{fuel}}_{id_i}|$.

Table 1. Sets, parameters, and variables in the AASP.

Sets with indices	Explanation
$\mathscr{F}$	A set of aircraft (index i, j).
$\mathscr{R}_i$	A set of alternative descent air routes for aircraft i (index r_i).
$\mathscr{V}$	A vertex set of waypoints in the TMA (index $o_i, \cdots, k, l, \cdots, d_i$), where $\mathscr{V}_i \in \mathscr{V}$.
$\mathscr{A}$	An air segment set of descent air route in the TMA, where $\mathscr{A}_i \in \mathscr{A}$.
$\mathscr{G}$	A directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$.
Parameters	**Explanation**
i, j	Aircraft ID.
r_i	Descent air route for aircraft i.
k, l, u, v	Transit waypoint.
o_i	The entry waypoint for aircraft i.
d_i	The runway for aircraft i.
$\Gamma^{\text{early}}_{io_i}, \Gamma^{\text{late}}_{io_i}$	Estimated earliest (latest) arrival time for aircraft i enter the TMA.
$\Gamma^{\text{early}}_{id_i}, \Gamma^{\text{target}}_{id_i}, \Gamma^{\text{late}}_{id_i}$	Estimated earliest (target, latest) landing time at the runway d_i for aircraft i.
Γ_{id_i}	The reference landing time at the runway d_i for aircraft i, where $\Gamma_{id_i} = \max\{\Gamma^{\text{early}}_{id_i}, \Gamma^{\text{target}}_{id_i}\}$.
$\Gamma^{\text{fuel}}_{id_i}$	The landing time at the runway d_i with the minimum fuel consumption for aircraft i.
$\underline{\Theta}^{r_i}_{ik\_il}, \overline{\Theta}^{r_i}_{ik\_il}$	The minimum (maximum) travel time in air segment (k, l) for aircraft i.
S_{iju}	The minimum time-based separation for a preceding aircraft i and another trailing aircraft j in a same waypoint u.
B	Large artificial variable.
Decision variables	**Explanation**
$y^{r_i}_i$	1, if aircraft i uses descent air route $r_i \in R_i$; 0, otherwise.
z_{iju}	1, if aircraft i flies through the same waypoint u before aircraft j; 0, otherwise.
$t^{r_i}_{ik}$	The arrival time of waypoint k by descent air route r_i for aircraft i, where $t^{r_i}_{ik} \geq 0$.
α_i	The delay of aircraft i to its reference landing time, where $\delta_i = \max\{0, t^{r_i}_{id_i} - \Gamma_{id_i}\}$.
μ_i	The absolute value of the difference $|t^{r_i}_{id_i} - \Gamma^{\text{fuel}}_{id_i}|$.

Constraint (5b) ensures that exactly one descent air route can be chosen for each aircraft. Constraint (5c) enforces that $t^{r_i}_{ik}$ is a non-zero value when the descent air route r_i is selected, where B is a sufficiently large number such as the sum of the maximum travel time of all aircraft flying through all arcs. Constraint (5d) restricts that each aircraft i to enter the TMA between the estimated earliest arrival time $\Gamma^{\text{early}}_{io_i}$ and the latest arrival time $\Gamma^{\text{late}}_{io_i}$. Constraint (5e) allows the travel time in each air segment (k, l) to take any value within the range $\left[\underline{\Theta}^{r_i}_{ik\_il}, \overline{\Theta}^{r_i}_{ik\_il}\right]$. These lower and upper bounds are obtained by solving the aforementioned $(\mathbf{ATOP_E})$ and $(\mathbf{ATOP_L})$ formulations, respectively. Constraints (5f) and (5g) represent separation constraints, ensuring that if aircraft j follows aircraft i at the same waypoint $u \in \mathscr{V}^{r_i}_i \cap \mathscr{V}^{r_j}_j$, the arrival time at waypoint u of aircraft j must be greater than or equal to the arrival time of aircraft i plus the minimum time-based separation S_{iju}. Constraints (5h) and (5i) ensure that there is no overtake between aircraft i and j traveling through the common air segment $(u, v) \in \mathscr{A}_i \cap \mathscr{A}_j$, where $\sum_{r_i \in \mathscr{R}_i, (u,v) \in \mathscr{A}^{r_i}_i} y^{r_i}_i = 1$ ($\sum_{r_j \in \mathscr{R}_j, (u,v) \in \mathscr{A}^{r_j}_j} y^{r_j}_j = 1$) if aircraft i (j) uses arc (u, v), 0, otherwise. Constraints (5j) and (5k) define that the delay of aircraft i equals to $\max\{0, t^{r_i}_{id_i} - \Gamma_{id_i}\}$, where $\Gamma_{id_i} = \max\{\Gamma^{\text{early}}_{id_i}, \Gamma^{\text{target}}_{id_i}\}$. $\Gamma^{\text{early}}_{id_i}$ and $\Gamma^{\text{target}}_{id_i}$ are the earliest landing time and target landing time at the runway d_i of aircraft i, respectively, where $\Gamma^{\text{early}}_{id_i} = \Gamma^{\text{early}}_{io_i} + \sum_{(k,l) \in \mathscr{A}^{r_i}_i} \underline{\Theta}^{r_i}_{ik\_il}$.

Constraints (5l) and (5m) define the absolute value of difference between $t_{id_i}^{r_i}$ and $\Gamma_{id_i}^{\text{fuel}}$. The minimum fuel-burning landing time $\Gamma_{id_i}^{\text{fuel}}$ is obtained by solving the $(\mathbf{ATOP_F})$ formulation. Constraints (5n)–(5p) specify the domains of the decision variables.

Note that the $(\mathbf{AASP})$ can also be employed to allocate a feasible ascent route and the start time at each waypoint for every departing aircraft within a specified decision horizon in a TMA. In this context, each take-off aircraft follows an ascent trajectory, flying from the runway towards its assigned exit point. In the problem formulation, the origin o_i refers to the runway, while the destination d_i represents the exit waypoint.

2.4. Optimal Trajectory Selection

This module helps the ATCs select the optimal descent operation for each approaching aircraft. A cost function is established to prioritize the descent operations, which is formulated as the sum of the cost of total delay and the cost of total fuel consumption, as shown below:

$$\Omega = c_{delay}\Omega_{delay} + c_{fuel}\Omega_{fuel} \tag{6}$$

where c_{delay} and c_{fuel} are the unit delay cost (€/s) and unit fuel cost (€/kg), respectively. Ω_{delay} denotes the total delay experienced by all aircraft during a given decision time horizon, which is computed by solving the $(\mathbf{AASP})$. Ω_{fuel} denotes the total fuel consumption of all aircraft within the decision time horizon. To calculate the fuel consumption of each aircraft, an ATOP is formulated with a fixed required time of arrival at the runway. The objective of the ATOP is to minimize fuel consumption, and the formulation is as follows:

$$(\mathbf{ATOP_{F'}}): \quad \min \quad J_{fuel} \tag{7a}$$

$$\text{s.t.} \quad (1b)\text{–}(1m) \tag{7b}$$

$$t_f^{(|\mathscr{P}|)} = t_{RTA} \tag{7c}$$

where t_{RTA} is the required time of arrival at the runway, which is obtained by solving the $(\mathbf{AASP})$.

3. Solution Methods

The central component of our decision support framework consists of the proposed ATOP and AASP models. The ATOP is formulated as a Non-Linear Programming (NLP) optimization problem with interior point constraints, which can be solved numerically using two primary methods: direct and indirect [20,48]. In this study, a pseudospectral method [49] is employed as a direct method for solving the ATOP. On the other hand, the AASP is formulated as a MIP problem. While it can be solved using commercial solvers such as Gurobi, the high number and complexity of decision variables and constraints for a busy TMA can result in extremely long computational times. To address this issue, we have developed a VNS algorithm specifically tailored for solving the AASP. The VNS algorithm provides efficient solutions within a reasonable time frame, allowing for real-time decision-making. In practice, air traffic flow management in a TMA operates in a dynamic and evolving environment. To accommodate this dynamism, we adopt a rolling horizon approach to solve the dynamic cases of our decision support framework. This approach allows for continuous adaptation and optimization of trajectories and schedules based on updated information and changing conditions, ensuring effective and timely decision-making.

3.1. Pseudospectral Method

The pseudospectral method discretizes the time interval (time grid) and approximates the state and control variables in an optimal control problem. By converting the original infinite-dimensional problem into a finite-dimensional nonlinear optimization problem, it can be solved using standard NLP solvers like SNOPT [50]. In the pseudospectral method, the state and control inputs are approximated as piecewise polynomials, and collocation points are chosen using a quadrature rule over each time step. This approach offers high accuracy and efficiency, and for a large number of collocation points, it can yield an exact optimal solution [49].

For solving the ATOP in our study, we employ the GPOPS 5.0 package [51], which is a Matlab-based (R2021a) software specifically designed for solving multi-phase optimal control problems using the pseudospectral method. The SNOPT solver is integrated in the GPOPS as the underlying NLP solver, which is known for its robustness and efficiency in handling NLP problems. By utilizing GPOPS with the SNOPT solver, we can effectively solve the ATOP and obtain optimal trajectory solutions for aircraft descent operations in our decision support framework.

3.2. VNS Algorithm

The AASP consists of three main operations: routing, sequencing, and scheduling. Routing involves selecting the descent route for each aircraft, sequencing determines the arrival order of aircraft at each waypoint, and scheduling involves determining the arrival time for each aircraft at each waypoint. When a feasible sequence is given, which includes the selected descent route for each aircraft and the execution order of the aircraft at each waypoint, the optimal schedule for that sequence can be obtained in a polynomial time. This paper proposes a VNS algorithm that starts with an initial sequence and iteratively explores different neighborhood structures to search for improved sequences. Note that given a feasible sequence, obtaining an optimal schedule for each aircraft can be achieved using available algorithms or commercial solvers such as Gurobi.

The VNS algorithm follows a three-phase process: shaking, local search, and move. It starts with an initial sequence Π and its corresponding objective value $z(\Pi)$. The algorithm alternates between these phases to iteratively improve the sequence. In the shaking phase, the algorithm generates a neighbor sequence Π' by applying one of the available neighborhood structures $\mathcal{N}_q$, for $q = 1, 2, \cdots, q_{max}$, or a fixed structure $\mathcal{N}_1$. This neighbor sequence Π' is evaluated by calculating its objective value $z(\Pi')$. Next, in the local search phase, the algorithm manipulates the neighbor sequence Π' using the available neighborhood structures to further improve it. If an improved sequence Π'' with a lower objective value ($z(\Pi'') < z(\Pi)$) is obtained, it replaces the current best sequence, and the local search continues. If no improvement is achieved, the algorithm returns to the shaking phase. This process continues until the stopping criterion is met. The algorithm terminates based on two criteria: the maximum number of iterations ($iter_{max}$) and the maximum number of iterations without improvement ($noimproving_{max}$). Once the algorithm terminates, the improved sequence Π with its objective value $z(\Pi)$ is reported as the final solution. Algorithm 1 provides an overview of the VNS algorithm, while Algorithm 2 summarizes the steps involved in the local search phases. The generation of initial solutions and the definition of neighborhood structures are discussed in the following sections.

Algorithm 1 The variable neighborhood search (VNS) algorithm for AASP

Input: Integers $iter_{max}$, $noimproving_{max}$ and q_{max}. A set of neighbourhood structures $\mathcal{N}_q$ for $q = 1, 2, \cdots, q_{max}$. An initial sequence Π and its corresponding objective value $z(\Pi)$.

$iter \leftarrow 1$
$\xi \leftarrow 0$
while $iter \leq iter_{max}$ and $\xi < noimproving_{max}$ **do**
  /* **Shaking:** */
  $\Pi', z(\Pi') \leftarrow \mathcal{N}_1(\Pi)$
  /* **Local search:** */
  $\Pi'', z(\Pi'') \leftarrow LS(\Pi', q)$ /* See **Algorithm** 2 */
  /* **Move or not:** */
  if $z(\Pi'') < z(\Pi)$ **then**
    $\Pi \leftarrow \Pi''$
    $z(\Pi) \leftarrow z(\Pi'')$
    $\xi \leftarrow 0$
  else
    $\xi \leftarrow \xi + 1$
    $iter \leftarrow iter + 1$
  end if
end while

Output: An improved sequence Π and its objective value $z(\Pi)$.

Algorithm 2 The local search procedure $LS(\Pi', q)$ in VNS

$TL \leftarrow \emptyset$
$q \leftarrow 1$
while $q \leq q_{amx}$ **do**
  $cc_q \leftarrow$ Randomly select one from all positions of Π'.
  if cc_q not in TL **then**
    $temp, z(temp) \leftarrow \mathcal{N}_q(\Pi', cc_q, cr_q)$
    if $z(temp) < z(\Pi')$ **then**
      $\Pi' \leftarrow temp$
      $z(\Pi') \leftarrow z(temp)$
      $q \leftarrow 1$
      $TabuMemoryFilter(TL)$
    else
      $q \leftarrow q + 1$
      $TL \leftarrow TabuMemoryUpdate(TL, TL_{max}, cc_q, \Pi')$
    end if
  end if
end while

3.2.1. Initial Solution Generation Method

The AASP exhibits a pattern where the order of aircraft at each waypoint corresponds to the order of aircraft landing on the runway based on their scheduled landing times. This alignment is a result of designing descent routes to converge aircraft arrivals towards the runway (see in Figure 1) and the prohibition of overtaking in any air segment (see constraints (5i) and (5j)). Therefore, we can represent the sequence $\Pi = \left(\pi_1^{r_1}, \pi_2^{r_2}, \cdots, \pi_{|\mathscr{F}|}^{r_{|\mathscr{F}|}}\right)$ for the AASP, where the order list $\{1, 2, \cdots, |\mathscr{F}|\}$ denotes the execution order of the aircraft at each waypoint, and the set $\{r_1, r_2, \cdots, r_{|\mathscr{F}|}\}$ represents the descent route allocations.

To generate an initial solution for the AASP, we propose an algorithm outlined in Algorithm 3. The algorithm starts by sorting all aircraft at each waypoint in non-decreasing order based on their estimated earliest landing time $\Gamma_{id_i}^{\text{early}}$. This sorting ensures that the aircraft are initially arranged in a feasible order at each waypoint. We then formulate the

problem $(\mathbf{AASP_I})$, which fixes the aircraft order at each waypoint, and use the Gurobi 9.5.0 solver to solve it. The default parameters of Gurobi are used except for the time limit, which may be adjusted as needed. The initial sequence Π is obtained by sorting all aircraft in non-decreasing order of their scheduled time at the runway. This sequence, along with its objective value $z(\Pi)$, is used to initialize the local search.

Algorithm 3 The algorithm for generating the initial solution

1: Sort the aircraft at each waypoint in non-decreasing order of $\Gamma(i) = \Gamma_{id_i}^{\text{early}}$.
2: Solve the problem $(\mathbf{AASP_I})$ utilizing a commercial solver like Gurobi.
3: Obtain the initial sequence Π and its objective value $z(\Pi)$.

// The problem $(\mathbf{AASP_I})$ *is formulated as follows:*

$$
\begin{aligned}
(\mathbf{AASP_I}): \quad \min \quad & (5a) && (8a)\\
\text{s.t.} \quad & t_{ju}^{r_j} \geq t_{iu}^{r_i} + S_{iju} - B(2 - y_i^{r_i} - y_j^{r_j}) \\
& \forall i, j \in \mathscr{F}, \Gamma(i) < \Gamma(j), \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j} && (8b)\\
& (5b)\text{–}(5e),\ (5j)\text{–}(5n),\ (5p) && (8c)
\end{aligned}
$$

It is worth noting that we conducted preliminary experiments to determine the optimal rule for sorting aircraft at each waypoint. We tested three sorting rules based on estimated target landing time ($\Gamma_{id_i}^{\text{target}}$), earliest estimated landing time ($\Gamma_{id_i}^{\text{early}}$), and latest estimated landing time ($\Gamma_{id_i}^{\text{late}}$). Our findings suggest that sorting the aircraft by the earliest estimated landing time yields the best solution.

3.2.2. Improvement Algorithm

Once a feasible initial sequence Π has been obtained for the AASP, the VNS algorithm attempts to improve it by iteratively exploring its neighborhood structures for a designated number of iterations. Our VNS algorithm employs three unique neighborhood structures denoted as $\mathscr{N}_1$, $\mathscr{N}_2$, and $\mathscr{N}_3$.

The neighborhood structure $\mathscr{N}_1$ is derived from the R&S framework that allows for the exploration of improved solutions by relaxing routing and sequencing constraints for a selected sub-sequence of aircraft. This neighborhood structure differs significantly from traditional approaches. The process of the neighborhood structure $\mathscr{N}_1$ is illustrated in Figure 4. Given an initial sequence Π, the relaxing neighborhood selects a sub-sequence Π^R from Π based on two parameters: a randomly selected center cc_1 and a radius cr_1. Every sub-sequence Π^R is relaxed through relaxing the routing and sequencing constraints of the aircraft in the sub-sequence, therefore letting the aircraft change their descent route and order in the sequence. The remaining aircraft in the sequence, denoted as $\Pi^{NR} = \Pi \setminus \Pi^R$, are not subject to changes in routing or sequencing. We then reduce the $(\mathbf{AASP})$ with a given Π^R and Π^{NR} to problem $(\mathbf{AASP_R})$ (see Appendix A) and solve it using a commercial solver (e.g., the Gurobi). In the $(\mathbf{AASP_R})$ formulation, the descent routes and the execution orders at each waypoint for all aircraft in the sequence Π^{NR} are kept unchanged (see constraints (A1c)–(A1e)). However, for the aircraft in the sequence Π^R, the routing and sequencing constraints are relaxed, allowing for potential adjustments (see constraints (A1f)–(A1i)). As a result, solving the $(\mathbf{AASP_R})$ would involve re-routing, re-sequencing, and re-scheduling for the aircraft in Π^R, while only re-scheduling would occur for the aircraft in Π^{NR}. The sequence is updated if an improved solution is found, and the search continues.

Note that, while optimization solvers often require significant time to find the optimal route, sequence, and schedule for large instances, they display high efficiency in small-sized instances. Therefore, by relaxing the routing and sequencing constraints for a select few

aircraft, one can leverage the solver's capabilities and achieve an optimal or nearly optimal solution for the relaxed aircraft.

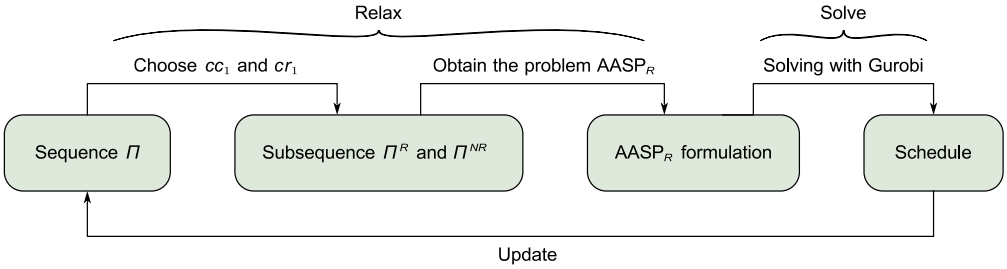


Figure 4. The process for the neighborhood structure $\mathcal{N}_1$.

Figure 5 shows an example of the $\mathcal{N}_1$ for 10 aircraft (labeled *a* to *j*) with 3 descent routes (1 to 3). The disjunctive arcs, represented by dashed orange lines, indicate that these aircraft are subject to possible re-routing, re-sequencing, and re-scheduling operations. On the other hand, the remaining aircraft in the sequence follow conjunctive arcs represented by solid black lines. These aircraft are only subject to re-scheduling operations, meaning their routing and sequencing remain unchanged.

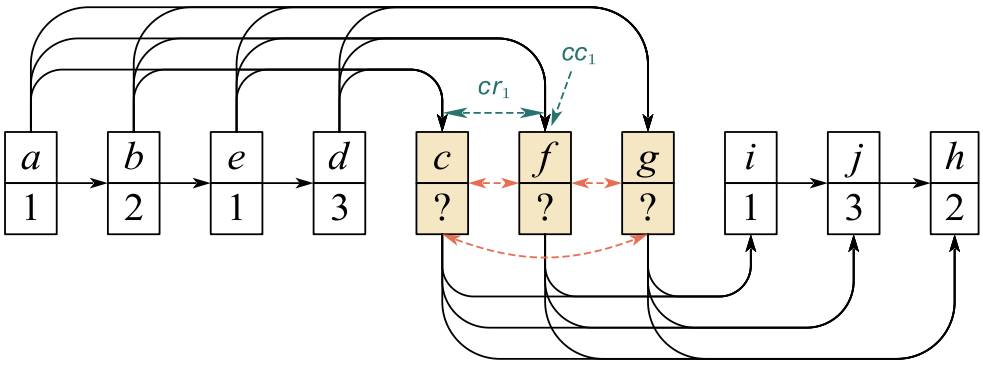


Figure 5. An example of the relaxing neighborhood.

The neighborhood structure $\mathcal{N}_2$ involves sequentially swapping the location of a randomly selected center cc_2 with the positions within a radius cr_2 for a given sequence Π. By applying the swap operation to the current sequence Π, a new sequence Π' is generated. To evaluate its quality, the AASP problem is reformulated using Π' and converted into a problem denoted as $(\mathbf{AASP_S})$ (see Appendix B). This reformulation captures the changes in the sequence and enables efficient solving using a commercial solver such as Gurobi, with the time limit parameter set appropriately. During the exploration process, only swaps that result in an improved solution are accepted and retained. An example of the swap operation in neighborhood structure $\mathcal{N}_2$ is illustrated in Figure 6.

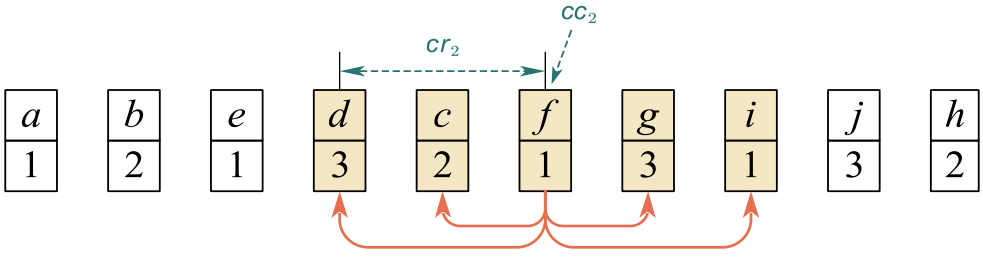


Figure 6. An example of a swap operation in neighborhood structure $\mathcal{N}_2$.

The neighborhood structure $\mathcal{N}_3$ involves changing the current descent route for each aircraft within a sub-sequence Π^C to a randomly selected alternative descent route, if one exists. The sub-sequence Π^C is randomly selected from a given sequence Π using a center cc_3 and a radius cr_3. A new sequence Π' is generated by applying the change operation to the current sequence Π within this neighborhood. Then, we obtain a solution from solving the $(\mathbf{AASP_S})$ with the fixed sequence Π'. Only the change operations that improve the current solution are accepted. Figure 7 shows an example of a change operation within neighborhood structure $\mathcal{N}_3$.

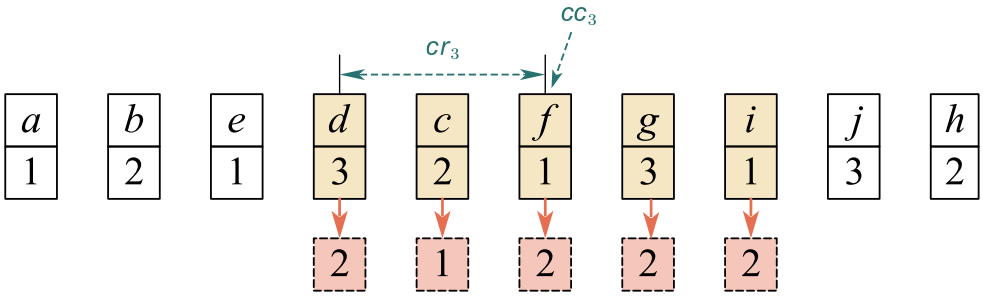


Figure 7. An example of a change operation within neighborhood structure $\mathcal{N}_3$.

Within the local search phase, a TS [52] approach is used to avoid being trapped in local optima and revisiting the same solution, as described in Algorithm 2. The TS efficiently employs a memory structure to move away from previous solutions. To facilitate this process, we introduce two operations: the tabu memory filter and the tabu memory update. The former operation eliminates all of the tabu moves from the current neighborhood (no aspiration criteria is used), while the latter operation updates the list of tabu moves during the search. The tabu list (TL) utilized in the local search phase has a maximum of TL_{max} moves, with each implemented move storing two pieces of key information: the current sequence and the value of the central parameter. It is important to note that in some cases, the application of neighborhood structures $\mathcal{N}_2$ and $\mathcal{N}_3$ may result in infeasible solutions for problem $(\mathbf{AASP_S})$. When this occurs, these operations are skipped, and the VNS algorithm proceeds to the next process.

We also note that the VNS algorithm only uses the neighborhood structure $\mathcal{N}_1$ in its shaking phase. The parameters cc_1 and cr_1 within $\mathcal{N}_1$ demonstrate variable values between the shaking and local search phases. In the shaking phase, we employ $\mathcal{N}_1$ to infuse diversification into our search, thereby facilitating an exploration of novel sectors within the solution space. In the local search, however, we deploy $\mathcal{N}_1$ to intensify the search process, thereby capitalizing on the incumbent solution to realize superior outcomes.

3.3. Rolling Horizon Approach

The rolling horizon approach is a dynamic strategy implemented in our decision support framework to handle real-time changes in the air traffic flow management. As illustrated in Figure 8, it involves breaking down the overall decision-making process into smaller decision time windows, each with a duration of $W_{decision}$. At the start of each time window, the decision support framework is executed to determine the optimal descent operation for each aircraft within the given time window based on the available information at that time. After completing the decision-making process for the current decision time window, the rolling horizon approach moves to the next time window with a time difference of R_{roll}. The rolling horizon approach iterates through the decision-making procedure for each subsequent decision time window until the end of the overall traffic decision. This iterative process ensures that the decision support framework remains responsive to the evolving traffic conditions and provides up-to-date recommendations for the optimal descent operations of the arriving aircraft.

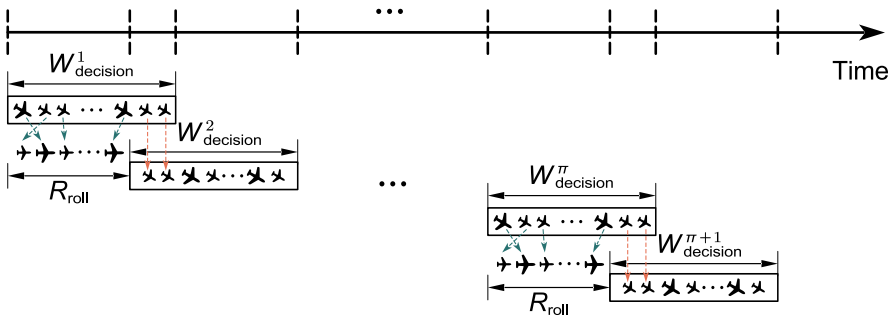


Figure 8. The rolling horizon approach.

In the rolling horizon approach, each decision time window consists of two groups of aircraft: ongoing aircraft from the previous window and new aircraft arriving in the current window. The ongoing aircraft's descent processes continue from the previous window, while their descent routes and arrival times serve as constraints for the new aircraft. The decision support framework considers these constraints to optimize the descent operations for both groups, ensuring safe separation between them. This approach allows for efficient and safe management of the entire set of aircraft within each decision time window.

4. Experimental Results

This section presents the numerical experiments conducted to evaluate the performance of the proposed decision support framework and the VNS algorithm. We implemented the pseudospectral technique to solve the ATOP problem using the Matlab R2021a programming language, and implemented the VNS algorithm to solve the AASP problem using the Python 3.9 programming language. The problems $(\mathbf{AASP})$, $(\mathbf{AASP_I})$, $(\mathbf{AASP_R})$, and $(\mathbf{AASP_S})$ are solved using the commercial solver Gurobi 9.5.0 [53] with default parameters, except for the time limit. All experiments are performed on a PC with an Intel(R) Core(TM) i7-10710U CPU@1.10GHz ARM 16G under the Windows 10 operating system.

4.1. Test Instances and Parameters Setting

4.1.1. Traffic Instances in Gbia

GBIA is the largest transport hub in China and serves as the primary hub for China Southern Airlines. According to the published Aeronautical Information Publication (AIP), Figure 9a displays the designed north orientation arrival routs utilized to approach the hub by incoming aircraft. Figure 9b highlights the path stretching routes for the ps-CDO procedure. We consider an E-TMA with a radius of 220 NM around the GBIA. The GBIA features three parallel runways expressly designed for arriving and departing procedures: runway 01/19, runway 02L/20R, and runway 02R/20L, all of which are shown in Figure 9c. Among them, Runway 01/19 is operational for both arrivals and departures, runway 02R/20L is exclusively reserved for arrivals, and runway 02L/20R is solely used for departures. This study assumes that the independent instrument approaches between runway 01/19 and runway 02R/20L, and the segregated parallel approaches and/or departures between runway 02R/20L and runway 02L/20R.

In our study, we conducted simulations using actual aircraft routing and scheduling data from GBIA TMA for a full day in November 2019. The test data set consists of 1290 aircraft, with 653 arrivals and 637 departures, including three types of aircraft: 267 Airbus A332 (heavy), 452 Airbus A320 (medium), and 571 Boeing B738 (medium). For all arrival aircraft, we assume a north orientation approach. To analyze different air traffic scenarios, we divided the day into three distinct time periods: low-density, normal-density, and high-density. Figure 10 illustrates the hourly movements during the simulated day, with the three highlighted periods representing the identified air traffic situations. We created six separate test instances based on these time periods. The instances labeled A1,

A2, and A3 considered only arrival traffic, while the instances labeled AD1, AD2, and AD3 included both arrival and departure traffic.

Our study focuses on analyzing the effects of arrival traffic exclusively within the E-TMA phase while disregarding enroute fluctuations. We set the arrival time at the entry waypoint identical to the estimated target arrival time for every arriving aircraft. For departure traffic, a designated runway is assigned for takeoff within a given time window. We set the departure window at 900 s, and assume the minimum fuel-burning take-off time is equal to the estimated target take-off time. The take-off air segments from runway exit for the departure aircraft is not considered in this paper. Therefore, the runways are the only potential points of conflict between the arrival and departure traffic.

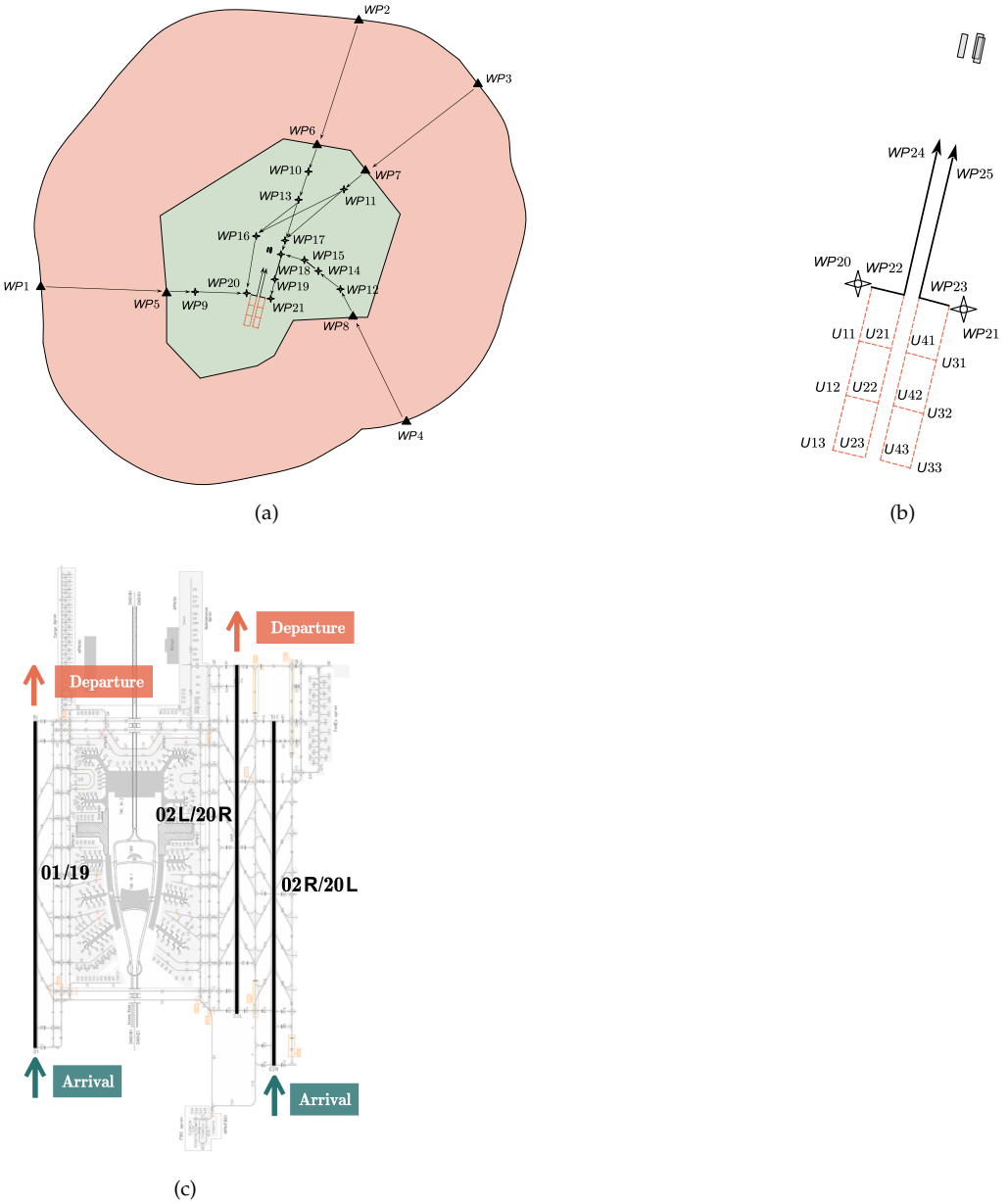


Figure 9. GBIA E-TMA configurations: (**a**) E-TMA of GBIA. (**b**) Path stretching design. (**c**) Parallel runway operation pattern.

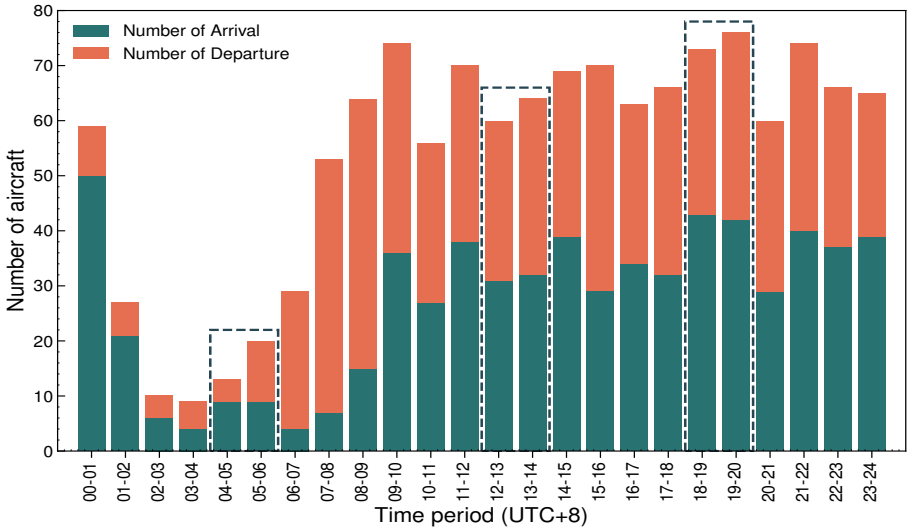


Figure 10. Hourly movements at GBIA.

4.1.2. Parameters Setting

In our numerical experiments, we computed the descent trajectories for every arriving aircraft, starting from the last part of the cruise prior to the TOD and ending at the runway. The aircraft maintain a cruise altitude of 35,000 feet until the TOD, where they enter the descending phase. The runway altitude is set to 50 feet. We refer to [11,48] for more details on altitude and speed constraints at each air segment in the ATOP for our three analyzed descent operations, including the SDO, the CDO, and the ps-CDO. The altitude-dependent wind profile W_s (h) used in our ATOP is taken from the meteorological terminal aviation routine weather report of the GBIA. The remaining performance parameters of the aircraft, consisting of the aerodynamic data, the fuel flow data, and the geometric data, are available in BADA.

In the AASP, the minimum time-based separation between two consecutive aircraft in the airspace is calculated by incorporating their angles and speeds. The calculation algorithm is based on the work of [2]. The minimum time-based separation matrices for operations on the same runway or close parallel runways are provided in Table 2 [54] and Table 3 [55], respectively. Aircraft involved in arrivals and departures are symbolized by "A" and "D", whereas aircraft weight categories are represented by "H", "M", and "L" for heavy, medium, and light aircraft, respectively. For example, "AH" denotes the arrival of a heavy aircraft.

Table 4 lists a summary of the parameter values used in the VNS algorithm. We classify all test instances in Table 4 into three classes based on the number of aircraft: up to 30 aircraft, 30 to 70 aircraft, and more than 70 aircraft. We solve all test instances within each group using these parameter values.

Table 5 displays the delay cost per unit c_{delay}, which is calculated based on [56] full tactical costs but excludes the fuel cost incurred by the delay. The fuel cost per unit c_{fuel} is set to 0.8 €/kg, in line with the base scenario in [56].

Table 2. The minimum time-based separation matrix for operations on the same runway (unit: s).

Preceding	AH	AM	AL	DH	DM	DL
AH	96	157	196	75	75	75
AM	60	69	131	75	75	75
AL	60	69	82	75	75	75
DH	60	60	60	90	120	120
DM	60	60	60	60	60	60
DL	60	60	60	60	60	60

Table 3. The minimum time-based separation matrix for operations on the close parallel runways (unit: s).

Preceding	Trailing					
	AH	**AM**	**AL**	**DH**	**DM**	**DL**
AH	-	-	-	68	68	80
AM	-	-	-	62	62	80
AL	-	-	-	48	55	80
DH	54	58	80	-	-	-
DM	54	58	80	-	-	-
DL	54	58	80	-	-	-

Note: Symbol "-" indicates that no valid operation combination is possible.

Table 4. Values of the parameters for the VNS algorithm.

Parameters	Number of Aircraft		
	$\|\mathscr{F}\| \leq$ **30**	**30** $< \|\mathscr{F}\| \leq$ **70**	$\|\mathscr{F}\| >$ **70**
$iter_{max}$	3	5	$\max\{5, [\frac{\|\mathscr{F}\|}{40}]\}$
$noimproving_{max}$	2	2	$\max\{3, [\frac{\|\mathscr{F}\|}{25}]\}$
cr_1 (shaking)	2	4	$\min\{[\frac{\|\mathscr{F}\|}{20}], 6\}$
cr_1 (local search)	4	6	$\min\{[\frac{\|\mathscr{F}\|}{15}], 10\}$
cr_2	2	2	$\max\{3, [\frac{\|\mathscr{F}\|}{35}]\}$
cr_3	2	2	$\max\{3, [\frac{\|\mathscr{F}\|}{35}]\}$
$timelimit_{initial}$ (s)	1	1	10
$timelimit_{LS}$ (s)	1	1	3
TL_{max}	10	15	$\max\{15, [\frac{\|\mathscr{F}\|}{10}]\}$

Note: $timelimt_{initial}$ and $timelimt_{LS}$ refer to the time limits set for the Gurobi solver during the initial solution generation phase and the local search phase, respectively.

Table 5. Values of the unit delay cost (€/s).

Delay (s)	Arrivals			Departures		
	A332	**A320**	**B738**	**A332**	**A320**	**B738**
(0, 300]	1.25	0.67	0.67	0.91	0.4	0.4
(300, 900]	1.64	0.89	0.91	1.29	0.63	0.64

4.2. Framework Decision Solutions

In this section, we investigate the performances of our proposed framework with respect to three descent operations (i.e., SDO, CDO, and ps-CDO). We arbitrarily select the results with respect to $\lambda = 0.5$. Table 6 displays the computational results for instances A1–AD3. In Table 6, the rows are organized in blocks of six rows for the three descent operations, including the values of the total cost, the total delay cost, the total fuel consumption cost, the total delay, the total difference, and the total fuel consumption.

Based on the analysis of total cost, the results show that different descent optimization methods yield optimal solutions depending on the specific scenarios and traffic densities. For instances A1 and AD1, which represent low traffic density scenarios, the CDO approach is found to be the optimal choice. For instances A2 and A3, which represent normal and high traffic density scenarios considering only arrival traffic, the ps-CDO approach proves to be the optimal choice. In scenarios that involve both arrivals and departures, such as instances AD2 and AD3, the SDO approach is identified as the best choice.

Note that the implementation of CDO in instances A2, A3, AD2, and AD3, which represent scenarios with dense traffic, is found to be infeasible. In such cases, ensuring safety while implementing CDO becomes challenging due to the need for extra separation buffers. As a result, arrival aircraft are unable to follow the CDO trajectories in these

scenarios. With the path stretching approach, on the other hand, each aircraft approaching based on the ps-CDO is provided with various options for its lateral path with respect to different arrival times. An AASP can thus be solved to properly sequence the aircraft in order to avoid aircraft conflicts. However, the inclusion of path stretching within the ps-CDO leads to increased travel distances for aircraft during descent, resulting in higher fuel consumption and reduced travel time efficiency.

Table 6. The computational results on instances A1–AD3.

	Indicators	A1	A2	A3	AD1	AD2	AD3
SDO	Total cost (€)	12,842.31	58,119.74	75,851.6	12,979.91 (12,882.92)	**63,028.02 (61,399.59)**	**80,465.05 (77,989.02)**
	Total delay cost (€)	547.89	11,018.2	12,392.15	678.67 (600.07)	15,839.66 (14,522.25)	17,510.85 (15,508.58)
	Total fuel consumption cost (€)	12,294.42	47,101.54	63,459.45	12,301.24 (12,282.85)	47,188.36 (46,877.34)	62,954.2 (62,480.44)
	Total delay (s)	655	11,984	13,480	807 (687)	18,200 (15,915)	19,991 (16945)
	Total difference (s)	1222	13,655	20,651	1356 (1236)	14,875 (12,590)	21,044 (17,998)
	Total fuel consumption (kg)	15,368.03	58,876.92	79,324.31	15,376.55 (15,353.57)	58,985.45 (58,596.68)	78,692.75 (78,100.55)
CDO	Total cost (€)	**11,483.37**	-	-	**11,589.12 (11,485.14)**	-	-
	Total delay cost (€)	1554.65	-	-	1618.84 (1534.64)	-	-
	Total fuel consumption cost (€)	9928.72	-	-	9970.28 (9950.5)	-	-
	Total delay (s)	1833	-	-	1996 (1862)	-	-
	Total difference (s)	969	-	-	1174 (1040)	-	-
	Total fuel consumption (kg)	12,410.89	-	-	12,462.85 (12,438.13)	-	-
ps-CDO	Total cost (€)	11,675.87	**56,729**	**74,834.28**	11,902.97 (11,799)	63,362.42 (61,272.22)	81256.84 (78,649.21)
	Total delay cost (€)	1444.2	15,652.89	20047.09	1657.49 (1573.29)	22,109.76 (20,401.98)	26,463.63 (24,355.8)
	Total fuel consumption cost (€)	10,231.67	41,076.11	54,787.19	10,245.48 (10,225.71)	41,252.66 (40,870.24)	54,793.21 (54,293.41)
	Total delay (s)	1798	16,566	21,115	2013 (1879)	23,138 (20,242)	28545 (25485)
	Total difference (s)	1109	10,078	14,264	1248 (1114)	12,284 (9388)	15,866 (12,806)
	Total fuel consumption (kg)	12,789.59	51,345.14	68,483.99	12,806.86 (12,782.14)	51,565.82 (51,087.79)	68,491.51 (67,866.76)

Note: Symbol "-" indicates that the aircraft scheduling problem obtains no feasible solution. A number within each parenthesis indicates the value for the arrival aircraft.

While the SDO approach performs well in terms of reducing total delay compared to the CDO and ps-CDO approaches, it leads to increased fuel consumption for arrival aircraft. On the other hand, the CDO-based approach demonstrates a fuel consumption reduction of approximately 15% compared to the SDO approach. As traffic flows increase and TMA capacity remains constant, total delay and its associated costs significantly increase. However, the increase in total fuel consumption and associated costs is negligible. The SDO approach, which reduces total delay, is thus more advantageous in terms of total cost and recommended for instances AD2 and AD3, which involve both arrival and departure traffic and higher traffic flows.

These findings highlight the trade-offs and considerations associated with different descent optimization approaches in terms of safety, fuel consumption, travel time efficiency, and delay reduction, particularly in scenarios with dense traffic. The choice of the optimal approach depends on the specific operational requirements and priorities in each scenario.

4.3. Sensitivity Analysis on Weight Parameter

In this section, we compare the performance of the SDO against the ps-CDO with varying values of the weight parameter λ within our framework. We increase the weight parameter λ from 0 to 1 with a step size of 0.1. Figure 11 illustrates the effect of altering the weight parameter λ on the total delay and fuel consumption of the SDO (solid lines) and the ps-CDO (dashed lines) for instances A1-AD3. The corresponding cost results of the SDO and the ps-CDO are presented in Figure 12, including the total cost, the total delay cost, and the total fuel consumption cost.

As the weight parameter λ increases, the importance of total delay in the AASP model increases, resulting in solutions with smaller total delay for both the SDO and the ps-CDO approaches. However, total fuel consumption and its associated costs gradually increase. The ps-CDO approach outperforms the SDO in terms of total fuel consumption and its associated cost, while the SDO performs better in terms of total delay and its associated costs, regardless of the value of λ.

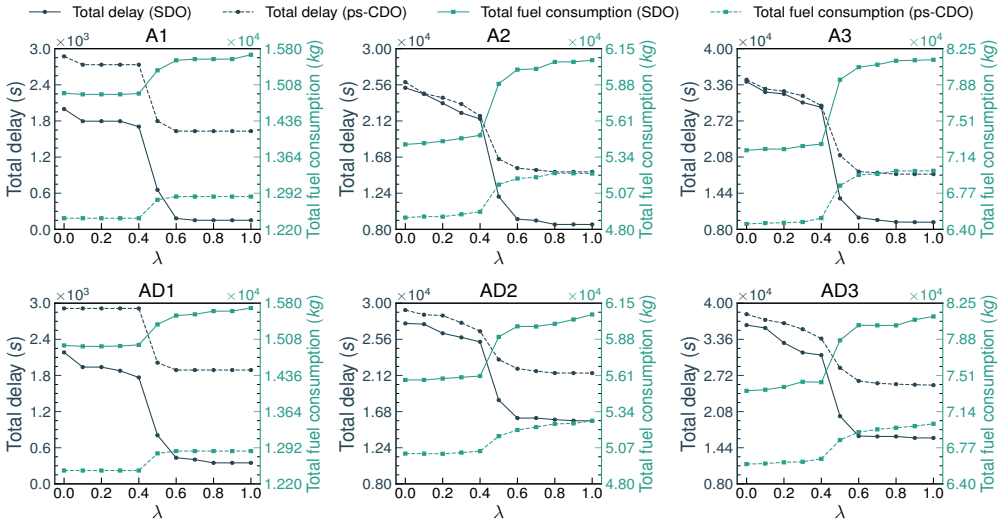


Figure 11. Effect on the total delay and fuel consumption for the SDO (solid lines) and the ps-CDO (dashed lines) to weight parameter λ.

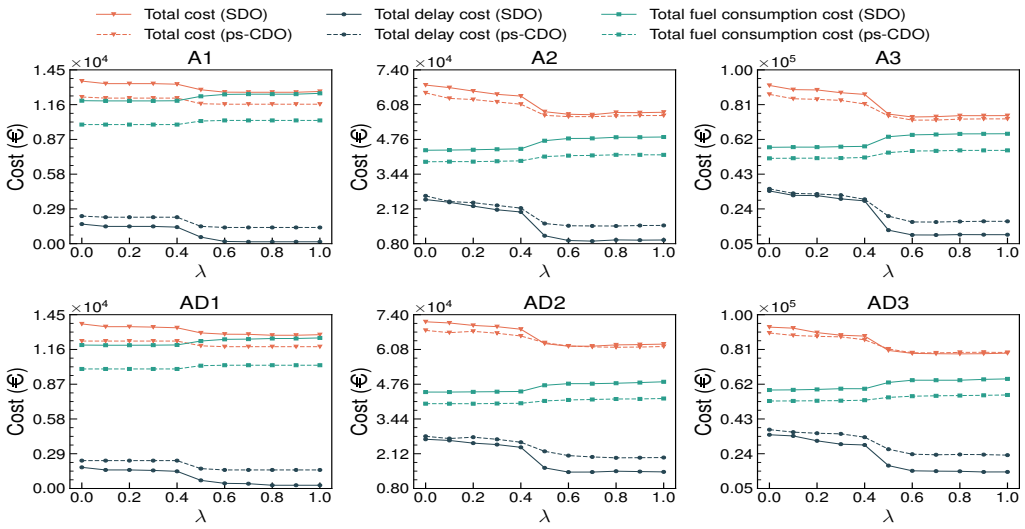


Figure 12. Costs results for the SDO (solid lines) and the ps-CDO (dashed lines) to weight parameter λ.

In terms of total cost, during low or normal traffic scenarios (instances A1, A2, A3, and AD1), the ps-CDO consistently yields better total cost results compared to the SDO. However, during high traffic scenarios (instances AD2 and AD3), the SDO generates smaller total cost compared to the ps-CDO when $\lambda = 0.5$ or 0.6 in AD2 and $\lambda \geq 0.5$ in AD3, respectively. This detailed trade-off information can assist ATCs in determining the optimal compromise between different performance indices using our framework.

4.4. Effectiveness of the VNS Algorithm

In this section, we evaluate the effectiveness of our proposed VNS algorithm for the AASP model. Here, we arbitrarily set $\lambda = 0.5$ and provide the findings in Table 7, in which we compare the VNS algorithm against the Gurobi solver with a time limit of 1800 s. The first three rows of the table list the outcomes of the Gurobi for the SDO, including the best objective function value, CPU time, and optimality gap at the time limit. Rows 4–6 present the results of our VNS algorithm for the same instance with the SDO run ten times, including the best objective function value, the average objective function value, and the average CPU time. Rows 7–12 provide the statistics for the ps-CDO. The number within each parenthesis indicates the frequency of obtaining the best solution among the 10 runs of the VNS algorithm.

Table 7. The computational results for the AASP on instances A1–AD3.

Index			A1	A2	A3	AD1	AD2	AD3
SDO	Gurobi	Obj. (s)	938.5	12,819.5	17,065.5	1081.5	16,706.5	20,590.5
		CPU time (s)	0.16	1800.1	1800.18	0.14	1800.21	1800.23
		Gap (%)	0	0.11	0.08	0	12.61	5.44
	Our VNS	Obj. (best) (s)	938.5 (10)	12,819.5 (10)	17,065.5 (10)	1081.5 (10)	16,537.5 (2)	20,517.5 (5)
		Obj. (avg.) (s)	938.5	12,819.5	17,065.5	1081.5	16,667.1	20,554
		CPU time (avg.) (s)	0.34	24.83	29.42	0.56	106.4	121.55
ps-CDO	Gurobi	Obj. (s)	1453.5	13,322	17,689.5	1630.5	17,713	22,205.5
		CPU time (s)	0.11	1800.07	1800.12	0.12	1800.11	1800.23
		Gap (%)	0	1.56	1.11	0	15.74	10.16
	Our VNS	Obj. (best) (s)	1453.5 (10)	13,322 (10)	17,689.5 (10)	1630.5 (10)	17,711 (8)	22,205.5 (10)
		Obj. (avg.) (s)	1453.5	13,322	17,689.5	1630.5	17,711.4	22,205.5
		CPU time (avg.) (s)	0.32	26.6	33.54	0.41	93.01	124.22

Note: A number within each parenthesis indicates the frequency of obtaining the best solution among the 10 runs of the VNS algorithm.

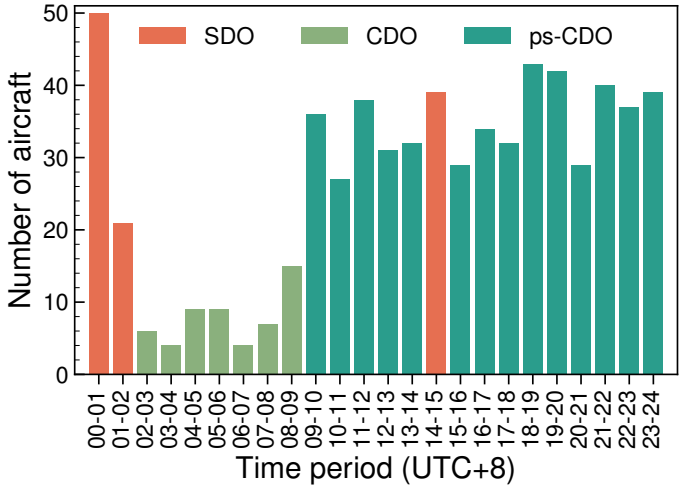
Our VNS algorithm performed significantly better than the Gurobi solver in terms of solution quality. In all instances, our algorithm produced solutions that were either equal to or better than the best solutions obtained by Gurobi. Notably, the average solution quality achieved by our VNS algorithm was comparable to the best solutions generated by Gurobi.

The VNS algorithm can produce near-optimal solutions for all instances within a few seconds, whereas the Gurobi algorithm fails to provide better solutions within the specified time limit for all instances except A1 and AD1. Furthermore, the reliability of our VNS algorithm is evident as it consistently delivers the best solutions after multiple runs, while maintaining reasonably short CPU times. As the number of aircraft increases, resulting in more variables and constraints, the iteration times naturally increase. However, the CPU time of our VNS algorithm does not increase proportionately, showcasing its efficiency in solving the AASP problem.

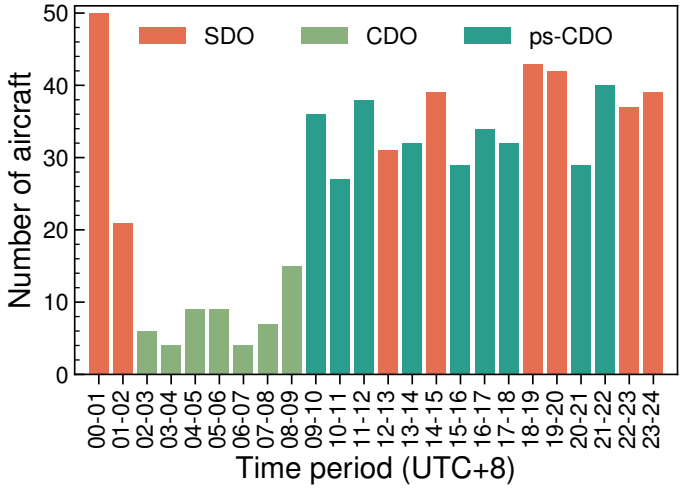
4.5. Decision Solutions for Daily Operations

In this section, we employ the rolling horizon approach to explore the decision solutions of our decision support framework over a full day. The complete day instances can be found in Section 4.1.1 (see Figure 10). We consider two cases: (i) arrival traffic only, and (ii) both arrival and departure traffic. The rolling horizon method is employed with parameter settings of $W_{decision} = 2$ h and $R_{roll} = 1$ h. The results are analyzed based on a weight parameter of $\lambda = 0.5$. Figure 13a shows the optimal descent operation solutions for arriving aircraft hourly throughout the day in case (i), while the outcomes for (ii) are presented in Figure 13b.

The effectiveness of the CDO procedure for arriving aircraft is most prominent during periods of low traffic density, specifically between 02:00 to 09:00 in both case (i) and case (ii). However, its practicality diminishes during the remaining periods of the day, as it fails to satisfy the minimum time-based separation requirement in the TMA. Implementing the CDO during these times would render the AASP problem infeasible. Regarding the ps-CDO, in case (i), it is the optimal choice for arriving aircraft between 09:00–14:00 and 15:00–24:00. For case (ii), it remains optimal during periods of normal traffic density, namely 09:00–12:00, 13:00–14:00, 15:00–18:00, and 21:00–22:00. However, during high traffic density periods, following the ps-CDO would lead to significantly increased delay costs, making it less favorable in terms of total cost. Thus, it is advisable to utilize the SDO procedure during high traffic density periods, as it minimizes the total cost. It should be noted that the ps-CDO cannot be utilized from 00:00–01:00, as it cannot provide conflict-free trajectories for arriving aircraft during this time.



(a)



(b)

Figure 13. Optimal descent operation solutions for arriving aircraft hourly throughout the day: (**a**) Case (i): arrival traffic only. (**b**) Case (ii): both arrival and departure traffic.

5. Conclusions

In this paper, we present an innovative collaborative decision support framework for future trajectory-based operations that integrates aircraft trajectory optimization, arrival scheduling, optimal trajectory selection, and their interactive processes within a single framework. First, the framework is capable of computing the aircraft descent trajectories by performing three distinct descent operations (SDO, CDO, and ps-CDO). Second, based on the descent trajectories, the framework allocates a conflict-free trajectory to each aircraft while minimizing the total delay and total fuel consumption. Finally, the framework selects the best descent operation out of the three considering minimizing the total cost of total delay and total fuel consumption. We evaluate the performance of our framework for traffic handling in the GBIA TMA, considering both arrival and departure traffic. The computational results demonstrate that our framework can support the ATCs in efficiently managing the descent operations in the TMA. Additionally, our VNS algorithm featuring three novel neighborhood structures proves to be efficient for solving the AASP. Therefore, we propose that this framework will serve as a technical enabler for future air traffic flow management.

One avenue for future research is to extend our decision support framework to include the point merge system [57] and vectoring maneuvers [58]. Our proposed framework heav-

ily relies on accurate trajectory planning, yet uncertainties such as weather [59], navigation accuracy, and pilot operations may undermine its performance and pose potential safety risks. Addressing these uncertainties in future research would be valuable for enabling us to develop a more robust decision support framework for realistic scenarios.

Author Contributions: Conceptualization, D.G., Z.H. and A.D.; methodology, D.G.; software, D.G.; validation, D.G. and Z.H.; formal analysis, D.G.; investigation, D.G.; resources, M.L.; data curation, D.G.; writing—original draft preparation, D.G.; writing—review and editing, M.L., Z.H. and A.D.; visualization, D.G.; supervision, M.L. and Z.H.; funding acquisition, Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the China Postdoctoral Science Foundation [Grants 2022T150319 and 2022M711611], the Postgraduate Research and Practice Innovation Program of Jiangsu Province [Grant KYCX22_0376] and the Humanities and Social Science Fund of the Ministry of Education [Grant 23YJC790027].

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TMAs	Terminal Maneuvering Areas
ATM	Air Traffic Management
ATCs	Air Traffic Controllers
TBO	Trajectory Based Operations
FMS	Flight Management System
SDO	Step-down Descent Operation
CDO	Continuous Descent Operation
TOD	Top of Descent
ATOP	Aircraft Trajectory Optimization Problem
AASP	Aircraft Arrival Scheduling Problem
VNS	Variable Neighborhood Search
VND	Variable Neighborhood Descent
FFPAD	Fixed Flight Path Angle Descent
RTAs	Required Times of Arrival
RSP	Runway Scheduling Problem
R&S	Relax-and-Solve
TS	Tabu Search
GBIA	Guangzhou Baiyun International Airport
E-TMA	extended-TMA

Appendix A. ($\mathbf{AASP_R}$) Model Formulation

Let $\mathscr{F}_{pre}^{\Pi^{NR}}$, $\mathscr{F}^{\Pi^R}$ and $\mathscr{F}_{suc}^{\Pi^{NR}}$ denote the aircraft set of the preceding part in Π^{NR}, the aircraft set in Π^R and the aircraft set of the succeeding part in Π^{NR}, respectively. If Π^{NR} contains the beginning (or end) of the Π, the $\mathscr{F}_{pre}^{\Pi^{NR}} \in \emptyset$ (or $\mathscr{F}_{suc}^{\Pi^{NR}} \in \emptyset$). The problem ($\mathbf{AASP_R}$) is formulated as follows:

$$
\begin{aligned}
(\mathbf{AASP_R}): \quad \min \quad & (5a) && \text{(A1a)}\\
\text{s.t.} \quad & (5b)\text{–}(5e), (5j)\text{–}(5p) && \text{(A1b)}\\
& t_{ju}^{r_j} \geq t_{iu}^{r_i} + S_{iju} \\
& \quad \forall i, j \in \mathscr{F}_{pre}^{\Pi^{NR}} \cup \mathscr{F}_{suc}^{\Pi^{NR}}, i < j, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j} && \text{(A1c)}\\
& t_{ju}^{r_j} \geq t_{iu}^{r_i} + S_{iju} - B(2 - y_i^{r_i} - y_j^{r_j}) \\
& \quad \forall i \in \mathscr{F}_{pre}^{\Pi^{NR}}, j \in \mathscr{F}^{\Pi^R}, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j}, \forall r_j \in \mathscr{R}_j && \text{(A1d)}
\end{aligned}
$$

$$t_{ju}^{r_j} \geq t_{iu}^{r_i} + S_{iju} - B(2 - y_i^{r_i} - y_j^{r_j})$$
$$\forall i \in \mathscr{F}^{\Pi^R}, j \in \mathscr{F}_{suc}^{\Pi^{NR}}, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j}, \forall r_i \in \mathscr{R}_i \quad \text{(A1e)}$$
$$t_{ju}^{r_j} \geq t_{iu}^{r_i} + S_{iju} - B(3 - z_{iju} - y_i^{r_i} - y_j^{r_j})$$
$$\forall i, j \in \mathscr{F}^{\Pi^R}, i < j, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j}, \forall r_i \in \mathscr{R}_i, \forall r_j \in \mathscr{R}_j \quad \text{(A1f)}$$
$$t_{iu}^{r_i} \geq t_{ju}^{r_j} + S_{jiu} - B(2 + z_{iju} - y_i^{r_i} - y_j^{r_j})$$
$$\forall i, j \in \mathscr{F}^{\Pi^R}, i < j, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j}, \forall r_i \in \mathscr{R}_i, \forall r_j \in \mathscr{R}_j \quad \text{(A1g)}$$
$$z_{iju} - z_{ijv} \geq \sum_{\substack{r_i \in \mathscr{R}_i \\ (u,v) \in \mathscr{A}_i^{r_i}}} y_i^{r_i} + \sum_{\substack{r_j \in \mathscr{R}_j \\ (u,v) \in \mathscr{A}_j^{r_j}}} y_j^{r_j} - 2$$
$$\forall i, j \in \mathscr{F}^{\Pi^R}, i < j, \forall (u,v) \in \mathscr{A}_i^{r_i} \cap \mathscr{A}_j^{r_j} \quad \text{(A1h)}$$
$$z_{ijv} - z_{iju} \geq \sum_{\substack{r_i \in \mathscr{R}_i \\ (u,v) \in \mathscr{A}_i^{r_i}}} y_i^{r_i} + \sum_{\substack{r_j \in \mathscr{R}_j \\ (u,v) \in \mathscr{A}_j^{r_j}}} y_j^{r_j} - 2$$
$$\forall i, j \in \mathscr{F}^{\Pi^R}, i < j, \forall (u,v) \in \mathscr{A}_i^{r_i} \cap \mathscr{A}_j^{r_j} \quad \text{(A1i)}$$
$$y_i^{r_i} = 1 \quad \forall i \in \mathscr{F}_{pre}^{\Pi^{NR}} \cup \mathscr{F}_{suc}^{\Pi^{NR}} \quad \text{(A1j)}$$

Appendix B. ($\mathbf{AASP_S}$) Model Formulation

The problem ($\mathbf{AASP_S}$) is formulated as follows:

$$(\mathbf{AASP_S}): \quad \min \quad (5a) \quad \text{(A2a)}$$
$$\text{s.t.} \quad \Gamma_{io_i}^{\text{early}} \leq t_{io_i}^{r_i} \leq \Gamma_{io_i}^{\text{late}} \quad \forall i \in \mathscr{F} \quad \text{(A2b)}$$
$$\underline{\Theta}_{ik\_il}^{r_i} \leq t_{il}^{r_i} - t_{ik}^{r_i} \leq \overline{\Theta}_{ik\_il}^{r_i} \quad \forall i \in \mathscr{F}, \forall (k,l) \in \mathscr{A}_i^{r_i} \quad \text{(A2c)}$$
$$t_{ju}^{r_j} \geq t_{iu}^{r_i} + S_{iju} \quad \forall i, j \in \mathscr{F}, i < j, \forall u \in \mathscr{V}_i^{r_i} \cap \mathscr{V}_j^{r_j} \quad \text{(A2d)}$$
$$\alpha_i \geq t_{id_i}^{r_i} - \Gamma_{id_i} \quad \forall i \in \mathscr{F} \quad \text{(A2e)}$$
$$\alpha_i \geq 0 \quad \forall i \in \mathscr{F} \quad \text{(A2f)}$$
$$\mu_i \geq t_{id_i}^{r_i} - \Gamma_{id_i}^{\text{fuel}} \quad \forall i \in \mathscr{F} \quad \text{(A2g)}$$
$$\mu_i \geq \Gamma_{id_i}^{\text{fuel}} - t_{id_i}^{r_i} \quad \forall i \in \mathscr{F} \quad \text{(A2h)}$$
$$t_{ik}^{r_i} \geq 0 \quad \forall i \in \mathscr{F}, \forall k \in \mathscr{V}_i^{r_i} \quad \text{(A2i)}$$

References

1. Lemetti, A.; Hardell, H.; Polishchuk, T. Arrival flight efficiency in pre- and post-Covid-19 pandemics. *J. Air Transp. Manag.* **2023**, *107*, 102327. [CrossRef] [PubMed]
2. Huo, Y.; Delahaye, D.; Sbihi, M. A dynamic control method for extended arrival management using enroute speed adjustment and route change strategy. *Transp. Res. Part Emerg. Technol.* **2023**, *149*, 104064. [CrossRef]
3. Eurocontrol. Daily Traffic Variation-States. Report, EUROCONTROL. 2020. Available online: https://www.eurocontrol.int/Economics/DailyTrafficVariation-States.html (accessed on 1 May 2024).
4. Chen, H.; Solak, S. Lower Cost Arrivals for Airlines: Optimal Policies for Managing Runway Operations under Optimized Profile Descent. *Prod. Oper. Manag.* **2015**, *24*, 402–420. [CrossRef]
5. Solak, S.; Chen, H. Optimal Metering Point Configurations for Optimized Profile Descent Based Arrival Operations at Airports. *Transp. Sci.* **2017**, *52*, 150–170. [CrossRef]
6. Riahi, V.; Newton, M.H.; Polash, M.; Su, K.; Sattar, A. Constraint guided search for aircraft sequencing. *Expert Syst. Appl.* **2019**, *118*, 440–458. [CrossRef]
7. Djokic, J.; Lorenz, B.; Fricke, H. Air traffic control complexity as workload driver. *Transp. Res. Part Emerg. Technol.* **2010**, *18*, 930–936. [CrossRef]

8. SESAR. European ATM Master Plan. Report, 2020 Edition SESAR JU. 2020. Available online: https://ec.europa.eu/research/participants/data/ref/h2020/other/call-doc-annexes/jtis/call-doc-annex_h2020-sesar-2020-1_atm-master-plan_en.pdf (accessed on 1 May 2024).
9. FAA. NextGen Implementation Plan. Report, 2018–2019 Edition Federal Aviation Administration. 2018. Available online: https://www.faa.gov/nextgen/2018-2019-nextgen-implementation-plan (accessed on 1 May 2024).
10. Sáez, R.; Prats, X.; Polishchuk, T.; Polishchuk, V. Traffic synchronization in terminal airspace to enable continuous descent operations in trombone sequencing and merging procedures: An implementation study for Frankfurt airport. *Transp. Res. Part Emerg. Technol.* **2020**, *121*, 102875. [CrossRef]
11. Gui, D.; Le, M.; Huang, Z.; Zhang, J.; D'Ariano, A. Optimal aircraft arrival scheduling with continuous descent operations in busy terminal maneuvering areas. *J. Air Transp. Manag.* **2023**, *107*, 102344. [CrossRef]
12. Erzberger, H.; Davis J.T.; Green, S. Design of center-tracon automation system. In Proceedings of the AGARD Guidance and Control Symposium on Machine Intelligence in Air Traffic Management, Berlin, Germany, 11–15 May 1993. Available online: https://ntrs.nasa.gov/citations/19940025065 (accessed on 1 May 2024).
13. Eurocontrol. *Arrival Manager, Implementation Guidelines and Lessons Learned*; Report, EUROCONTROL; Eurocontrol: Brussels, Belgium, 2010.
14. Itoh, E.; Uejima, K. Applying Flight-deck Interval Management based Continuous Descent Operation for Arrival Air Traffic to Tokyo International Airport. In Proceedings of the Tenth USA/Europe Air Traffic Management Research and Development Seminar, Chicago, IL, USA, 10–13 June 2013. Available online: https://pdfs.semanticscholar.org/3727/380c04588e023013c2558fe1cc093c9fe652.pdf (accessed on 1 May 2024).
15. Callantine, J.; Kupfer, M.; Martin, L.; Prevot, T. Simulations of Continuous Descent Operations with Arrival-Management Automation and mixed Flight-Dec Interval Management Equipage. In Proceedings of the Tenth USA/Europe Air Traffic Management Research and Development Seminar, Chicago, IL, USA, 10–13 June 2013.
16. Turgut, E.T.; Usanmaz, O.; Cavcar, M.; Dogeroglu, T.; Armutlu, K. Effects of Descent Flight-Path Angle on Fuel Consumption of Commercial Aircraft. *J. Aircr.* **2019**, *56*, 313–323. [CrossRef]
17. Toratani, D.; Wickramasinghe, N.K.; Westphal, J.; Feuerle, T. Feasibility study on applying continuous descent operations in congested airspace with speed control functionality: Fixed flight-path angle descent. *Aerosp. Sci. Technol.* **2020**, *107*, 106236. [CrossRef]
18. Sun, M.; Rand, K.; Fleming, C. 4 Dimensional waypoint generation for conflict-free trajectory based operation. *Aerosp. Sci. Technol.* **2019**, *88*, 350–361. [CrossRef]
19. Pawełek, A.; Lichota, P.; Dalmau, R.; Prats, X. Fuel-Efficient Trajectories Traffic Synchronization. *J. Aircr.* **2019**, *56*, 481–492. [CrossRef]
20. Samà, M.; D'Ariano, A.; Palagachev, K.; Gerdts, M. Integration methods for aircraft scheduling and trajectory optimization at a busy terminal manoeuvring area. *OR Spectr.* **2019**, *41*, 641–681. [CrossRef]
21. Beasley, J.E.; Krishnamoorthy, M.; Sharaiha, Y.M.; Abramson, D. Scheduling Aircraft Landings—The Static Case. *Transp. Sci.* **2000**, *34*, 180–197. [CrossRef]
22. Bennell, J.A.; Mesgarpour, M.; Potts, C.N. Airport runway scheduling. *Ann. Oper. Res.* **2013**, *204*, 249–270. [CrossRef]
23. Ikli, S.; Mancel, C.; Mongeau, M.; Olive, X.; Rachelson, E. The aircraft runway scheduling problem: A survey. *Comput. Oper. Res.* **2021**, *132*, 105336. [CrossRef]
24. Prakash, R.; Piplani, R.; Desai, J. An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput. *Transp. Res. Part Emerg. Technol.* **2018**, *95*, 570–581. [CrossRef]
25. Beasley, J.E.; Sonander, J.; Havelock, P. Scheduling aircraft landings at London Heathrow using a population heuristic. *J. Oper. Res. Soc.* **2001**, *52*, 483–493. [CrossRef]
26. Hu, X.B.; Chen, W.H. Receding Horizon Control for Aircraft Arrival Sequencing and Scheduling. *IEEE Trans. Intell. Transp. Syst.* **2005**, *6*, 189–197. [CrossRef]
27. Hu, X.B.; Di Paolo, E. A Ripple-Spreading Genetic Algorithm for the Aircraft Sequencing Problem. *Comput. Oper. Res.* **2011**, *19*, 77–106. [CrossRef] [PubMed]
28. Hancerliogullari, G.; Rabadi, G.; Al-Salem, A.H.; Kharbeche, M. Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem. *J. Air Transp. Manag.* **2013**, *32*, 39–48. [CrossRef]
29. Hammouri, A.I.; Braik, M.S.; Al-Betar, M.A.; Awadallah, M.A. ISA: A hybridization between iterated local search and simulated annealing for multiple-runway aircraft landing problem. *Neural Comput. Appl.* **2019**, *32*, 11745–11765. [CrossRef]
30. Zhi-Hui, Z.; Jun, Z.; Yun, L.; Ou, L.; Kwok, S.K.; Ip, W.H.; Kaynak, O. An Efficient Ant Colony System Based on Receding Horizon Control for the Aircraft Arrival Sequencing and Scheduling Problem. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 399–412.[CrossRef]
31. Xu, B. An efficient Ant Colony algorithm based on wake-vortex modeling method for aircraft scheduling problem. *J. Comput. Appl. Math.* **2017**, *317*, 157–170. [CrossRef]
32. Salehipour, A.; Moslemi Naeni, L.; Kazemipour, H. Scheduling aircraft landings by applying a variable neighborhood descent algorithm: Runway-dependent landing time case. *Int. J. Appl. Oper. Res.* **2009**, *1*, 39–49.
33. Salehipour, A.; Modarres, M.; Moslemi Naeni, L. An efficient hybrid meta-heuristic for aircraft landing problem. *Comput. Oper. Res.* **2013**, *40*, 207–213. [CrossRef]

34. Sabar, N.R.; Kendall, G. An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem. *Omega* **2015**, *56*, 88–98. [CrossRef]
35. Benlic, U.; Brownlee, A.E.I.; Burke, E.K. Heuristic search for the coupled runway sequencing and taxiway routing problem. *Transp. Res. Part Emerg. Technol.* **2016**, *71*, 333–355. [CrossRef]
36. Salehipour, A.; Ahmadian, M.M. A heuristic algorithm for the aircraft landing problem. In Proceedings of the The 22nd International Congress on Modelling and Simulation (MODSIM), Hobart, Tasmania, Australia, 3–8 December 2017.
37. Ahmadian, M.M.; Salehipour, A. Heuristics for flights arrival scheduling at airports. *Int. Trans. Oper. Res.* **2020**, *29*, 2316–2345. [CrossRef]
38. Bianco, L.; Dell'Olmo, P.; Giordani, S. Scheduling models for air traffic control in terminal areas. *J. Sched.* **2006**, *9*, 223–253. [CrossRef]
39. D'Ariano, A.; D'Urgolo, P.; Pacciarelli, D.; Pranzo, M. Optimal sequencing of aircrafts take-off and landing at a busy airport. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 19–22 September 2010. [CrossRef]
40. Samà, M.; D'Ariano, A.; Pacciarelli, D. Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports. *Transp. Res. Part Logist. Transp. Rev.* **2013**, *60*, 140–155. [CrossRef]
41. D'Ariano, A.; Pacciarelli, D.; Pistelli, M.; Pranzo, M. Real-time scheduling of aircraft arrivals and departures in a terminal maneuvering area. *Networks* **2015**, *65*, 212–227. [CrossRef]
42. Samà, M.; D'Ariano, A.; D'Ariano, P.; Pacciarelli, D. Scheduling models for optimal aircraft traffic control at busy airports: Tardiness, priorities, equity and violations considerations. *Omega* **2017**, *67*, 81–98. [CrossRef]
43. Samà, M.; D'Ariano, A.; Corman, F.; Pacciarelli, D. Coordination of scheduling decisions in the management of airport airspace and taxiway operations. *Transp. Res. Part Pol. Prac.* **2018**, *114*, 398–411. [CrossRef]
44. Samà, M.; D'Ariano, A.; D'Ariano, P.; Pacciarelli, D. Optimal aircraft scheduling and routing at a terminal control area during disturbances. *Transp. Res. Part Emerg. Technol.* **2014**, *47*, 61–85. [CrossRef]
45. Samà, M.; D'Ariano, A.; Corman, F.; Pacciarelli, D. Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas. *Transp. Res. Part Emerg. Technol.* **2017**, *80*, 485–511. [CrossRef]
46. Hull, D. *Fundamentals of Airplane Flight Mechanics*; Springer: Berlin/Heidelberg, Germany, 2007.
47. Eurocontrol. *User Manual for the Base of Aircraft Data (BADA) Revision 3.9*; Report, EEC Technical/Scientific Report No. 11/03/08-08; Eurocontrol: Brussels, Belgium, 2011.
48. Sáez, R.; Prats, X. Time-based-fuel-efficient aircraft descents: Thrust-idle descents along re-negotiated routes vs. powered descents along published routes. *Transp. Res. Part D Transp. Environ.* **2023**, *114*, 103563. [CrossRef]
49. Garg, D.; Patterson, M.; Hager, W.W.; Rao, A.V.; Benson, D.A.; Huntington, G.T. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica* **2010**, *46*, 1843–1851. [CrossRef]
50. Gill, P.E.; Murray, W.; Saunders, M.A. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Rev.* **2005**, *47*, 99–131. [CrossRef]
51. Rao, A.V.; Benson, D.A.; Darby, C.; Patterson, M.A.; Francolin, C.; Sanders, I.; Huntington, G.T. Algorithm 902: GPOPS, A MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method. *Acm Trans. Math. Softw.* **2010**, *37*, 1–39. [CrossRef]
52. Glover, F. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **1986**, *13*, 533–549. [CrossRef]
53. Gurobi Optimization Limited Liability Company. *Gurobi Optimizer Reference Manual*; Gurobi: Beaverton, ON, USA, 2022.
54. Balakrishnan, H.; Chandran, B.G. Algorithms for Scheduling Runway Operations Under Constrained Position Shifting. *Oper. Res.* **2010**, *58*, 1650–1665. [CrossRef]
55. Solak, S.; Sölveling, G.; Clarke, J.P.B.; Johnson, E.L. Stochastic Runway Scheduling. *Transp. Sci.* **2018**, *52*, 917–940. [CrossRef]
56. Cook, A.J.; Tanner, G. *European Airline Delay Cost Reference Values*; Report; Transport Studies Group, University of Westminster: London, UK, 2015.
57. Liang, M.; Delahaye, D.; Marechal, P. Conflict-free arrival and departure trajectory planning for parallel runway with advanced point-merge system. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 207–227. [CrossRef]
58. Jun, L.Z.; Alam, S.; Dhief, I.; Schultz, M. Towards a greener Extended-Arrival Manager in air traffic control: A heuristic approach for dynamic speed control using machine-learned delay prediction model. *J. Air Transp. Manag.* **2022**, *103*, 102250. [CrossRef]
59. Lui, G.N.; Hon, K.K.; Liem, R.P. Weather impact quantification on airport arrival on-time performance through a Bayesian statistics modeling approach. *Transp. Res. Part Emerg. Technol.* **2022**, *143*, 103811. [CrossRef]