Version of Record: https://www.sciencedirect.com/science/article/pii/S0307904X23003062
Manuscript_eb5392c1bd6678b1421d62d853f9c340

A heuristic approach for scheduling advanced air mobility aircraft at vertiports

Julián Alberto Espejo-Díaz[a,b], Edgar Alfonso-Lizarazo*[a], Jairo R. Montoya-Torres[b]

[a] *Université Jean Monnet Saint-Étienne, LASPI, F-42023, Saint-Etienne, France*
[b] *Universidad de La Sabana, School of Engineering, Research group Logistics Systems, Campus Universitario, Puente del Comun, Km. 7, Autopista Norte, Bogotá, Colombia*

Abstract

Recent progress in electric vertical take-off and landing (eVTOL) vehicles suggests that soon these vehicles could safely and efficiently transport people and cargo in urban areas. Therefore, advanced air mobility vehicles could become an alternative means of transport to overcome traffic congestion in cities in the upcoming years. There has been enormous interest from companies and governments in recent years in developing such technologies and enabling markets for new air transportation services. Despite the interest in the topic, little research has been done to address the aircraft scheduling problem in advanced air mobility take-off and landing areas (vertiports). The vertiports serve as the airports of eVTOL vehicles and could experience congestion problems similar to those of airports. This work proposes two optimization models for scheduling departing and landing aircraft at the vertiports' common ground taxi routes (taxiways), gates, and touchdown and lift-off (TLOF) pads. The mathematical models include advanced air mobility features such as separation rules and blocking constraints. As scheduling objectives, the first model maximizes the vertiport throughput, and the second model minimizes the deviation from the expected take-off/landing time. In addition, as a solution methodology, we developed two heuristic algorithms that use scheduling rules to assign and sequence the aircraft to the vertiport components. Computational results show that the optimization models find optimal schedules for small-sized instances of up to 10 aircraft, while the heuristic algorithms provide good results in terms of solution quality and computational time for large instances.

Keywords: advanced air mobility, electric vertical take-off and landing vehicles, vertiport, scheduling, optimization, heuristics

1. Introduction

Over the past years, there have been multiple initiatives to develop new air transportation services worldwide. An example of such initiatives is the Airbus Urban Air Mobility (UAM) program, which focuses not only on infrastructure development, regulatory compliance and technology advancement but also on the development of the CityAirbus NextGen, a four-seat electric vertical take-off and landing (eVTOL) aircraft [1]. Another example is the UAM Initiative Cities Community (UIC2) of the European Union, which fosters collaboration among different actors to shape the future of UAM in Europe [2]. In addition, the NASA Advanced Air Mobility Mission is working on air transportation systems for emerging aviation markets that use advanced air mobility (AAM) aircraft [3]. In addition, Boeing and Wisk (formerly Kitty Hawk) released the Concept of Operations (ConOps) for uncrewed passenger-carrying UAM operations [4]. Moreover, the company Volocopter is designing the VoloCity air taxi, an eVTOL aircraft with a capacity of two passengers [5]. In addition, Hyundai is working on the S-A1 aircraft, an electric air taxi with a capacity of four passengers and autonomous flights [6].

Apart from passenger carrying or cargo delivery, advanced air transportation could also include emergency medical services, humanitarian missions, or weather monitoring [7, 8]. These new air transportation services comprise the AAM paradigm. AAM involves all the systems and tools responsible for making air transportation operations safe, sustainable, affordable, efficient and accessible in emerging aviation services between urban, suburban and rural areas [9, 10].

With the growing interest in AAM technologies and their many potential applications, it is only a matter of time before the air space in urban areas becomes congested. The urban air mobility market is projected to grow to 9.1 billion by 2030, and air passenger transportation services in the metropolis will become a highly profitable market in the following years [11]. A crucial component of AAM services is their take-off/landing areas known as vertiports. Vertiports are specialized areas with adequate infrastructure components so that eVTOL vehicles can perform their take-off and landing operations. These vehicles can transport either passengers, cargo or both. In this work, both passenger and cargo aircraft use the vertiports in the same way, following the same stages.

Furthermore, vertiport traffic management is considered one of the most significant barriers to successfully expanding these new air transportation services [12, 13]. In addition, the study of Cohen et al. [9] identified air traffic management and vertiport infrastructure control as important challenges in AAM implementation. In addition, Garrow et al. [14] pointed out the need for research studies that tackle the synchronization of take-offs and landings at vertiports.

To manage the expected high volume of departing and landing aircraft, the vertiports' traffic controllers should rely on methodologies for assigning and sequencing aircraft at the different vertiport components. In this work, we studied the take-off and landing operations at vertiports. To do so, we proposed two mathematical models for scheduling departing and landing aircraft at vertiports. This approach considers AAM operational characteristics at vertiports, such as minimum separation rules and avoiding multiple aircraft using or being in the same vertiport infrastructure simultaneously for safety reasons.

We contribute to the AAM literature by proposing a mathematical approach to schedule departing and landing aircraft at vertiports. This problem consists of assigning and sequencing eVTOL vehicles on the vertiport components that are common ground routes (taxiways), the gates, and the touchdown and lift-off (TLOF) pads. Our proposal makes these scheduling decisions considering two important objectives of air transportation services: throughput maximization and minimization of the deviation from the target arrival/departure time. This work can be helpful for future vertiport traffic controllers to establish efficient schedules of slots to maximize the vertiport capacity or consider take-off/landing target times while scheduling aircraft. We list the main contributions of this work as follows:

1. We formulated the aircraft scheduling problem at vertiports as mixed integer linear programming (MILP) models considering important features of AAM operations, such as the blocking constraints, separation requirements at the TLOF pads and take-off/landing target times. The models can be implemented and solved in any commercial solver.
2. We studied two important objectives of air transportation services: maximizing the vertiport throughput (number of operations per hour)

and minimizing the deviation from the expected aircraft departure/arrival times.

3. We developed two heuristic algorithms as solution methodologies to enhance practicality in terms of the number of aircraft and vertiport size.
4. We propose a set of instances for testing and comparing the performance of MILP formulations and heuristic algorithms.

The rest of the paper is organized as follows: Section 2 reviews the related works. The vertiport take-off and landing operations that are the focus of this work are detailed in Section 3. Section 4 presents the mathematical formulations. Section 5 develops the solution methodology. Section 6 shows and discusses the results of computational experiments. Section 7 provides the conclusions and suggests future works in the field.

2. Related literature

In recent years, there has been considerable interest in AAM problems from the operational research (OR) perspective. In this section, we review and classify the related and relevant works to provide an up-to-date picture of the current state of research in the field. We review the works that considered AAM vehicles for passenger transportation since these vehicles are more likely to use vertiports. Baik et al. [15] is one of the first studies to consider AAM vehicles from an OR perspective. They proposed a mode-choice model to predict annual county-to-county person round trips considering the demand for different modes of transportation, including air taxis in the United States. The model provides the total cost and time for each transportation mode and can be used to evaluate the effect of policy-making on the transportation system. More recently, in [16], the authors used simulation to determine the number of air taxis to fulfill the demand considering AAM features such as battery charging times, vehicle maintenance, and vertiport locations. The authors tested their proposal in a case study in New York City (NYC) and recommended an operational fleet size of air taxis to find a balance between customer waiting time and vehicle usage.

Furthermore, several studies have been carried out on the selection of vertiport locations or the vertiport selection problem. In [17], the authors used an iterative clustering approach with multimodal transportation based on a warm start technique to propose potential vertiport locations. They tested their approach in a case study in NYC and recommended establishing vertiports at multiple locations, such as JFK international airport and South-Central Park. In addition, they concluded that the percentage of time savings and commuters' willingness to fly do not significantly impact the number and location of vertiports. The factors with significant impact are the on-road travel limit and the percentage of customer demand satisfaction. Another work studying the vertiport selection problem is [18]. In this work, the authors determined the vertiport locations in the Seoul metropolitan area, considering the effect of aircraft noise on the population. To do so, the authors first determined the candidate vertiport locations by analyzing and clustering statistical data from South Korea's commuting population. Next, the authors selected the vertiports and their connecting routes, comparing a business priority scenario (shortest distance) with a noise priority scenario. They concluded that the noise priority routes decreased 76.9% of the number of people affected by noise and were more efficient than the business priority routes. In addition, [19] proposed a variable neighborhood search heuristic algorithm for locating vertiports minimizing the total travel cost. To do so, the authors modeled the region under study as a grid structure and proposed a model for an uncapacitated single allocation p-hub median problem. In addition, the authors considered the possibility of excluding certain areas (grid cells) from being candidates for vertiport locations. Experimental results showed that the proposed heuristic algorithm solves almost all instances optimally. Moreover, [20] contributes to the vertiport selection problem by proposing a methodology for locating vertiports considering vehicle limitations (speed and battery range), operational strategies, and trips involving multiple vertiports (transfers). The authors modeled the problem as a single-allocation p-hub median location problem and proposed five heuristic methods for solving it. Another work that addresses the vertiport selection problem is Shin et al. [21]. In this work, the authors proposed a hub location model for determining the optimal locations of vertiports minimizing traveling costs, facility costs, and collision risk costs. As a solution methodology for large-sized instances, they developed a heuristic model based on a genetic algorithm. They tested their approach in two districts of Seoul, South Korea, and apart from testing the performance of their proposal, the numerical results also suggested that when the risk of collision is high, air-taxi services may not be viable or attractive in the region, and ground transportation should be preferred.

Kleinbekman et al. [22] were among the first authors to address the scheduling problem in vertiports. They studied the eVTOL vehicle landing problem in vertiports using a mixed-integer linear program. The proposal considers separation requirements and aircraft battery energy to determine the best aircraft arrival trajectories. Likewise, Pradeep and Wei [23] studied the vertiport scheduling problem for the eVTOL vehicle landing problem. The authors combined an insertion and local search heuristic with an MILP and a time advanced algorithm for minimizing aircraft arrival makespan. Their proposal schedules AAM aircraft arrivals in real time. In [24], the authors proposed a rolling-horizon scheduling algorithm for eVTOLs, minimizing deviations from the preferred aircraft arrival time. The authors tested their proposal on a simulation of a vertiport network, showing that a 50 s delay per eVTOL is expected during peak hours and less than 10 s in off-peak hours.

In addition, in [25], the author proposed a scheduling formulation for a heterogeneous fleet of AAM vehicles and solved it using different algorithms. The author concluded that fleet heterogeneity affects the service level and operational efficiency. Bosson and Lauderlade [26] implemented NASA's AutoResolver algorithm in an AAM context. In the implementation, they simulated a dense traffic scenario with AAM aircraft flying between 20 vertiports. The results of the simulation suggested that the algorithm has the potential to face the integration of AAM aircraft with other air-side operations. Bertram and Wei [27] proposed and tested an airspace design to handle an uncertain number of AAM aircraft. They used a Markov decision process-based algorithm for determining separation and collision avoidance while sequencing aircraft landings in a vertiport. In [28], the authors proposed a mathematical approach for tackling the cooperative scheduling problem of approach departure considering multiple vertiports. Their proposal included the design of an eVTOL operating environment and a scheduling model. Moreover, in [29], the authors proposed a scheduling model to minimize costs related to battery swapping and charging. Trajectory planning problems have also been studied using OR methodologies. Chauhan and Martins [30] proposed an optimization model to determine the optimal take-off trajectory for AAM vehicles with passengers. The authors considered flight mechanisms in the trajectory optimization model. The results of this work highlight the influence of wing loading and available power on trajectories. A classification of the previous articles and the positioning of our approach is presented in Table 1.

According to the review, there are multiple areas where OR methodolo-

Table 1: Related literature on OR applications in AAM problems

Reference	Objective	OR method	Type of approach	Case study	Main features	Decision level	Decision-making
[17]	Estimate the AAM demand and propose vertiport locations	and data analysis	Deterministic	New York City - USA	Demand, homogeneous fleet, willingness to fly, locations and travel times	Strategic	Vertiport selection
[24]	Propose a scheduling algorithm for eVTOL landing vehicles	Optimization	Deterministic	Houston USA	Battery level, homogeneous fleet, locations, separation rules, and aircraft trajectories	Operational	Trajectory planning, scheduling
[25]	Develop a scheduling formulation for AAM heterogeneous fleet	Optimization , algorithmics, and metaheuristics	Deterministic	Synthetic instances	Demand, heterogeneous fleet, travel times, and travel speed	Operational	Aircraft scheduling
[29]	Propose strategies for battery swapping and recharging of eVTOLs	Optimization	Deterministic	Hawaii USA	Battery level, demand, charging times, homogeneous fleet, and aircraft trajectories	Operational	Battery swap and recharge scheduling
[16]	Determine the number of air taxis to fulfil the demand	Simulation	Stochastic	New York City - USA	Battery level, demand, maintenance, locations, and charging times	Tactical	Fleet planning
[31]	Study the integration of AAM technology with other transportation modes	Simulation, Data analysis	Stochastic	Munich Germany	Demand, homogeneous fleet, ,aircraft capacity, and fleet, locations, travel speed	Strategic	Business model decisions
[30]	Propose an optimization model to determine AAM aircraft trajectories	Optimization	Deterministic	Synthetic instances	Battery level, aircraft trajectories, travel speed, and travel times	Strategic	Trajectory definition
[28]	Develop a control system for multi-vertiport terminal areas	Optimization, and algorithmics	Deterministic	Synthetic instances	Battery level, transit control rules, separation rules, and multiple vertiports	Tactical and operational	Transit rules and scheduling
[21]	Propose a hub location model to determine AAM vertiport locations	Optimization and heuristics	Deterministic	Seoul South Korea	Travel times, AAM costs, collision risks, vertiport locations, and travel speed	Strategic	Vertiport selection
This approach	Model and solve the scheduling problem at vertiports	Optimization and heuristics	Deterministic	Synthetic instances	Blocking constraints, target times, homogeneous, fleet, and separation rules	Operational	Aircraft scheduling

gies have been applied and can be applied to improve AAM services. Such methodologies can contribute to the development of models and solution procedures that support decision-making at strategic, tactical, and operational levels in upcoming AAM services. Straubinger et al. [32] concluded that large-scale vertiports capable of offering high throughput require handling complexity of ground operations. We tackle the complexity of ground operations by proposing an optimization model and a solution methodology for scheduling the operations of departing and landing aircraft at vertiport components.

Previous research works such as [24, 25, 28] have studied aircraft scheduling problems at vertiports. However, to the best of our knowledge, no one has addressed the topic of scheduling aircraft considering the vertiports' infrastructure components such as gates, taxiways and TLOF pads. In addition, we addressed the problem of scheduling aircraft at vertiports considering separation rules, blocking constraints and take-off/landing delays.

Scheduling aircraft at airports (traditional airline problems) and scheduling aircraft at vertiports have commonalities. For instance, in both problems, there are separation requirements between departing and landing aircraft or common objectives such as the minimization of the makespan [33]. However, scheduling aircraft at airports differs significantly from scheduling aircraft at vertiports. For example, scheduling aircraft at airports consists mainly of optimizing the use of take-off and landing surfaces (runways) [33], while scheduling aircraft at vertiports involves optimizing the use of take-off and landing surfaces (TLOF pads) and the use of common ground routes (taxiways) and gates. Therefore, for modeling the particularities in the vertiport scheduling problem, new approaches must be proposed in the literature.

Next, we introduce the problem of scheduling take-off and landing aircraft at the vertiport components.

3. Problem definition

In this section, we describe the take-off and landing operations at vertiports and discuss vertiport scheduling objectives. According to the NASA ConOps for high-density vertiport operations [34] and the work of Vascik and Hansman [35], the infrastructure components where aircraft perform their operations at vertiports are the following:

- The staging stand: A zone where aircraft remain when they are not in service. This area can be seen as the vertiports' hangars.

- The common ground routes or taxiways: They connect the vertiports' infrastructure components; they are the paths aircraft should follow when taxiing at the vertiport.
- The gates: Areas where the aircraft loads or unloads its cargo or passengers.
- The TLOF pads: Take-off and landing surfaces at vertiports.

Figure 1 illustrates the previous components in a vertiport section. Next, we describe the take-off and landing processes at the vertiports.

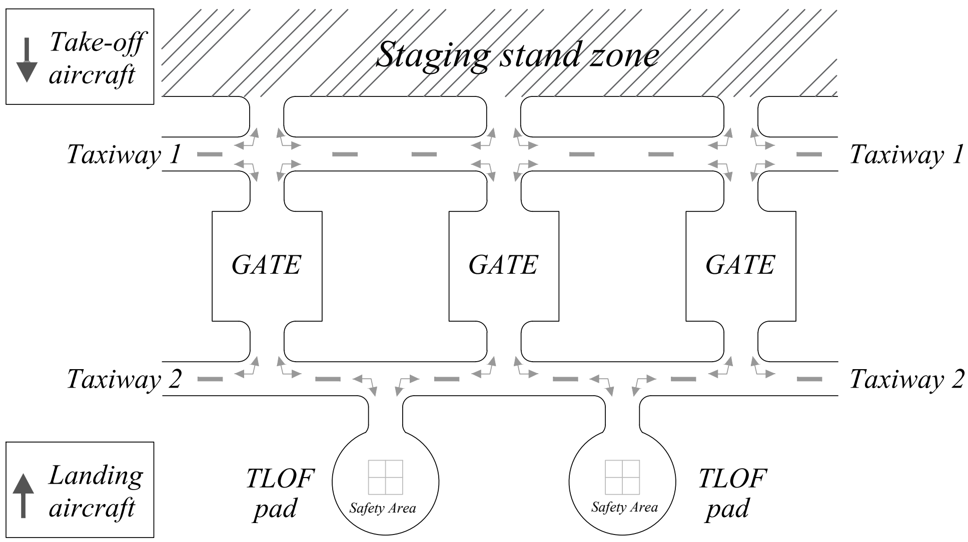

Figure 1: Vertiport section, adapted from [35]

Consider a vertiport where a set aircraft has to be scheduled. In our approach, the vertiport is divided into four stages that represent the utilization of the vertiport components as follows: the first stage corresponds to the utilization of taxiway 1; the second stage represents the utilization of the gates; the third stage corresponds to the utilization of taxiway 2; and the fourth stage corresponds to the utilization of the TLOF pads. These stages are defined regardless of the type of aircraft operation (departing or landing). For the departing aircraft subset, the process is as follows: The first step is to taxi from the staging stand zone to the gates using taxiway 1 (vertiport stage 1). In the second step, the take-off aircraft loads its cargo or passengers and prepares to leave the gate (vertiport stage 2). In the third step, the departing aircraft uses taxiway 2 for taxiing from the gate to its assigned TLOF pad (vertiport stage 3). Finally, the departing aircraft prepares to take off and then lifts off from the TLOF pad (vertiport stage 4).

For the landing aircraft, the process is as follows: In the first step, the landing aircraft proceeds from the final approach fix, positions itself above the TLOF pad, lands, and prepares to leave the TLOF pad to its assigned gate (vertiport stage 4). Next, in the second step, the landing aircraft uses taxiway 2 to taxi from the TLOF pad to its assigned gate (vertiport stage 3). In the third step, the landing aircraft unloads the passengers or cargo at the assigned gate and prepares to leave the gate (vertiport stage 2). Finally, in the fourth step, the landing aircraft taxis from the gate to the staging stand zone using taxiway 1 (vertiport stage 1).

It is worth noting that the order of the vertiport stages for the take-off and landing processes are opposite. For example, vertiport stage 4, which corresponds to the utilization of the TLOF pads, is the first step in the landing process, while it is the last step for departing aircraft. The time required for performing the previous operations depends on the type of aircraft and the cargo size or number of passengers to load or unload in the aircraft. Next, we describe the AAM features and objectives considered in this study.

3.1. Blocking constraints

For safety reasons, no more than one aircraft can simultaneously use or stay at the same vertiport component. In other words, an aircraft should wait in its vertiport component until the assigned component of the next stage is available. In addition, no other aircraft can use a component while another aircraft is waiting on it. It avoids situations where aircraft are in conflict. For instance, a landing aircraft using a taxiway in one direction encounters a take-off aircraft using the same taxiway in the other direction. We denoted this feature in our approach as blocking constraints.

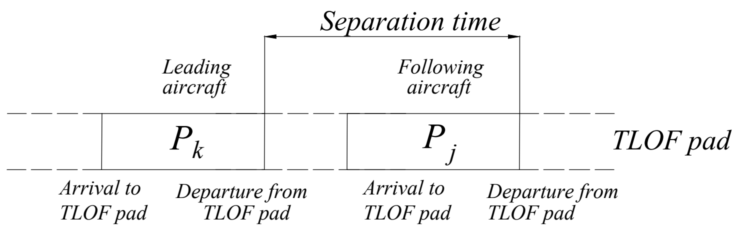

Figure 2: Calculation of separation times between consecutive aircraft at vertiports

3.2. Separation requirements at TLOF pads

Another important condition that has to be considered while scheduling aircraft at vertiports is the separation rules. When an aircraft takes off or lands from a TLOF pad, it cannot simultaneously be in the same vertical flight phase as other aircraft [36]. Thus, consecutive aircraft must meet a minimum separation time while using the same TLOF pad.

Figure 2 illustrates previous scenarios where k denotes the preceding aircraft and j the subsequent aircraft of each pair of consecutive aircraft. In addition, P_j and P_k are the processing times at the TLOF pad stage.

3.3. Objectives

Vertiports can have multiple scheduling objectives depending on the decision-maker, similar to airports. For example, AAM companies would seek schedules that minimize their operating costs, or governments could desire schedules that reduce vertiport environmental effects. In this work, we studied two important objectives of air operation services: maximizing the throughput (number of operations per time unit) and minimizing the deviation from target arrival/departure time [33, 37]. Although these objectives are frequently sought in traditional air services, they can be of interest to AAM services.

Maximizing the throughput would allow vertiports to handle the departures and landings of a high volume of aircraft. In this way, traffic controllers would propose efficient schedules that increase the vertiports' capacity. In this work, we maximize the throughput by minimizing the take-off or landing time of the last sequenced aircraft (makespan). In addition, minimizing the deviation from target arrival/departure time would allow vertiports to consider departure/landing target times. For instance, the target times can contemplate remaining battery levels for landing an aircraft. According to the works of [9, 24], battery technology is one of the main challenges in enabling AAM services. However, it has yet to advance to make these new air services safe and competitive. Thus, the initial flights made by eVTOL vehicles may have battery-related constraints or limited battery power. In addition, the target times can provide priorities to departing aircraft such as eVTOL vehicles used for medical purposes.

3.4. Remarks

Considering that the same operation aircraft follow the same flow, the take-off and landing processes can be modeled as a hybrid flowshop scheduling (HFS) problem. The hybrid flowshop is an environment in which a set of jobs are to be processed in a series of stages, and at least one stage contains more than one parallel machine [38]. Two major decisions are considered in the HFS problem. First, job assignment to machines and job sequencing in machines. In our work, the parallel machines are the vertiport infrastructure components, and the aircraft are the jobs to be assigned and sequenced in the machines. Notably, the number and integer nature of the assignment and sequencing decision variables makes the HFS a complex combinatorial problem that is, in most cases, NP-hard [38]. One version of an HFS configuration that is not NP-hard is the preemptive HFS problem presented in [39]. However, these properties do not apply to the problem at hand. For instance, in our proposal, it is not possible to stop the landing process of an aircraft when it has started its final approach and resumes it later.

Gupta [40] showed that even the simplest case of an HFS problem with two stages, one of them containing two machines, is NP-hard. Since our approach is a variant of the HFS problem with more than two stages and includes more features (e.g., separation rules), it corresponds to an NP-hard problem. The next section presents two mathematical formulations for scheduling take-off and landing eVTOL vehicles at the vertiport components. The models can be helpful to represent the problem and solve small-sized instances. The first model minimizes the completion time of the last operation, either landing or take-off (minimization of makespan), without considering the take-off/landing target times. The second model considers the aircraft target times and minimizes the deviation from target arrival/departure time. In addition, Section 5 develops two heuristic algorithms for facing large-sized instances.

4. Mathematical formulation

We modeled the problem as a variant of the HFS problem that considers the blocking constraints presented in the work of [41]. While the work of [41] considers one flow, we modeled these blocking constraints considering two types of flows, one for the departing aircraft and the other for the landing aircraft. This required us to include an additional index in the decision variables and consider more constraints to contemplate both flows. Another extension of [41] is considering the separation rules between consecutive aircraft using the same TLOF pads. In this case, we evaluated the departure time at the TLOF pad stage for each pair of consecutive aircraft using the same TLOF pad, and then we set the appropriate separation requirements. In addition, our proposal considered take-off and landing target times to grant priority to departing aircraft and observe the remaining battery levels for landing aircraft. Finally, we studied two objective functions: the minimization of the makespan and the minimization of the deviation from the target arrival/departure time. We made the following assumptions:

1. It is not possible to skip the loading or unloading stage at the gates for both departing and landing aircraft.
2. The infrastructure components at each stage are identical (parallel machines). Thus, the aircraft processing times at the infrastructure components of the same stage are the same.
3. The take-off or landing target time of each aircraft is known in advance.

4.1. Notation

The set of departing aircraft is denoted by $Dep = \{1, ..., d\}$, the set of landing aircraft by $Lan = \{d + 1, ..., d + l\}$, and the aircraft set by $A = \{1, ..., d + l\}$. We use the indices j and k to identify the aircraft. The stage set is denoted by $I = \{1, ..., 4\}$, and each element in the set is denoted by the index $i \in I$. The taxiway 1 stage corresponds to the stage $i = 1$, the gates stage to $i = 2$, the taxiway 2 stage to $i = 3$, and the TLOF pads stage to $i = 4$. We use the auxiliary index $e \in I$ to denote the stages. To refer to the vertiport infrastructure components at stage $i \in I$, we use the set $B = \{1, ..., m_i\}$, where m_i is the number of vertiport components in stage $i \in I$. Each element of the set B is denoted by l. The processing time for each aircraft $j \in A$ at stage $i \in I$ is denoted by the parameter P_{ji}. The separation time for ensuring lateral distances and avoiding aircraft in the same vertical phase flight is denoted by S, and we use a large value M for logical constraints. The decision variables are as follows:

$$X_{jkie} = \begin{cases} 1 \text{ if the aircraft } j \text{ is scheduled at stage } i \text{ after the aircraft } k \\ \quad \text{is scheduled at stage } e \\ 0 \text{ otherwise} \end{cases}$$

$$Y_{jil} = \begin{cases} 1 \text{ if the aircraft } j \text{ uses component } l \text{ at stage } i \\ 0 \text{ otherwise} \end{cases}$$

D_{ji} departure time for aircraft j at stage i.
C_{max} makespan of the last departing or landing aircraft.

4.2. Mathematical model for maximizing the vertiport throughput

Objective function 1 minimizes the take-off or landing time of the last sequenced aircraft (makespan).

$$Minimize\ C_{max} \tag{1}$$

Subject to the constraints:

1. Each aircraft uses one vertiport infrastructure component per stage.

$$\sum_{l=1}^{m_i} Y_{jil} = 1 \quad \forall j \in A; i \in I \tag{2}$$

2. An aircraft can move to an infrastructure component of the next stage if the current stage is completed. For departing aircraft, constraint set 3 computes the completion times of the taxiway 1 stage, and constraint set 4 calculates the completion time for the remaining stages. For landing aircraft, constraint set 5 calculates the completion times of the TLOF pad stage, and constraint set 6 computes the completion time for the remaining stages.

$$D_{j1} \geq \ P_{j1} - M(1 - Y_{j1l}) \ \ \forall j \in Dep; \forall l \in B \tag{3}$$

$$D_{ji} \geq \ D_{ji-1} + P_{ji} - M(1 - Y_{jil}) \ \ \forall j \in Dep; \forall i \in I | i > 1; \forall l \in B \tag{4}$$

$$D_{j4} \geq \ P_{j4} - M(1 - Y_{j1l}) \ \ \forall j \in Lan; \forall l \in B \tag{5}$$

$$D_{j4-i} \geq \; D_{j5-i} + P_{j4-i} - M(1 - Y_{j4-il}) \\ \forall j \in Lan; \forall i \in I|i < 4; \forall l \in B \tag{6}$$

3. Constraint sets 7 and 8 guarantee that the aircraft does not simultaneously use the same infrastructure component at the same stage when other aircraft are using it.

$$D_{ji} \geq \; D_{ke} + P_{ji} - M(3 - X_{jkie} - Y_{jil} - Y_{kel}) \\ \forall j, k \in A|j \neq k; \forall i, e \in I|i = e; \forall l \in B \tag{7}$$

$$D_{ke} \geq \; D_{ji} + P_{ji} - MX_{jkie} - M(2 - Y_{jil} - Y_{kel}) \\ \forall j, k \in A|j \neq k; \forall i, e \in I|i = e; \forall l \in B \tag{8}$$

4. The aircraft waits on its current stage until the infrastructure component of the next stage is available. Thus, no queues are allowed (blocking constraints). Constraint sets 9 and 10 guarantee the blocking constraints for each pair of departing aircraft, while constraint sets 11 and 12 guarantee the blocking constraints for each pair of landing aircraft.

$$D_{ji} \geq \; D_{ke+1} - M(3 - X_{jki+1e+1} - Y_{ji+1l} - Y_{ke+1l}) \\ \forall j, k \in Dep|j \neq k; \forall i, e \in I|i = e \neq 4, \forall l \in B \tag{9}$$

$$D_{ke} \geq \; D_{ji+1} - MX_{jki+1e+1} - M(2 - Y_{ji+1l} - Y_{ke+1l}) \\ \forall j, k \in Dep|j \neq k; \forall i, e \in I|i = e \neq 4, \forall l \in B \tag{10}$$

$$D_{ji} \geq \; D_{ke-1} - M(3 - X_{jki-1e-1} - Y_{ji-1l} - Y_{ke-1l}) \\ \forall j, k \in \; Lan|j \neq k; \forall i, e \in I|i = e \neq 1, \forall l \in B \tag{11}$$

$$D_{ke} \geq \; D_{ji-1} - MX_{jki-1e-1} - M(2 - Y_{ji-1l} - Y_{ke-1l}) \\ \forall j, k \in \; Lan|j \neq k; \forall i, e \in I|i = e \neq 1, \forall l \in B \tag{12}$$

For opposite operation aircraft, constraint sets 13, 14, 15, and 16 guarantee this condition. These constraints use the auxiliary set of tuples $Q = \{(j \in Dep), (k \in Lan)\} \cup \{(j \in Lan), (k \in Dep)\}$ for identifying each pair of opposite operation aircraft.

$$\begin{aligned} D_{ji} \geq \; & D_{ke+1} - M(3 - X_{jki+1e+1} - Y_{ji+1l} - Y_{ke+1l}) \\ & \forall (j,k) \in Q; \forall i, e \in I | i = e < 4; \forall l \in B \end{aligned} \tag{13}$$

$$\begin{aligned} D_{ke} \geq \; & D_{ji+1} - MX_{jki+1e+1} - M(2 - Y_{ji+1l} - Y_{ke+1l}) \\ & \forall (j,k) \in Q; \forall i, e \in I | i = e < 4; \forall l \in B \end{aligned} \tag{14}$$

$$\begin{aligned} D_{ji} \geq \; & D_{ke-1} - M(3 - X_{jki-1e-1} - Y_{ji-1l} - Y_{ke-1l}) \\ & \forall (j,k) \in Q; \forall i, e \in I | i = e > 1; \forall l \in B \end{aligned} \tag{15}$$

$$\begin{aligned} D_{ke} \geq \; & D_{ji-1} - MX_{jki-1e-1} - M(2 - Y_{ji-1l} - Y_{ke-1l}) \\ & \forall (j,k) \in Q; \forall i, e \in I | i = e > 1; \forall l \in B \end{aligned} \tag{16}$$

5. Minimum separation times between consecutive aircraft using the same TLOF pad must be considered to prevent them from being in the same vertical flight phase and ensure the lateral separations. Constraint set 17 observes the minimum separation times for consecutive aircraft.

$$\begin{aligned} D_{j4} \geq D_{k4} + P_{j4} + S - M(3 - X_{(jk)(44)} - Y_{j4l} - Y_{k4l}) \\ \forall j, k \in A | j \neq k; \forall \; l \in B \end{aligned} \tag{17}$$

6. Constraint sets 18 and 19 calculate the makespan for departing and landing aircraft, respectively.

$$C_{max} \geq D_{j4} \quad \forall j \in Dep \tag{18}$$

$$C_{max} \geq D_{j1} \quad \forall j \in Lan \tag{19}$$

7. Constrint sets 20, 21, 22, and 23 are the domain constraints for the decision variables.

$$D_{ji} \geq 0 \quad \forall j \in A; \forall i \in I \tag{20}$$

$$X_{jkie} \in \{0,1\} \quad \forall j,k \in A | j \neq k; \forall (i,e) \in I | i \neq e \tag{21}$$

$$Y_{jil} \in \{0,1\} \quad \forall j \in A; \forall i \in I; \forall l \in B \tag{22}$$

$$C_{max} \geq 0 \tag{23}$$

4.3. Mathematical model for minimizing the deviation from target arrival/departure time

In this model, we consider for each aircraft $j \in A$ the parameter T_j, which denotes the target take-off/landing time. In addition, we include the decision variable δ_j $\forall j \in A$ for computing the deviation from T_j. Objective function 24 minimizes the deviation from target arrival/departure time as presented in [33].

$$Minimize \ \sum_{j=1}^{d+l} \delta_j \tag{24}$$

Subject to the constraints (2)-(17), (20)-(22), and the following constraints for computing δ_j.

$$\delta_j \geq D_{4j} - T_j \quad \forall j \in Dep \tag{25}$$

$$\delta_j \geq T_j - D_{4j} \quad \forall j \in Dep \tag{26}$$

$$\delta_j \geq D_{1j} - T_j \quad \forall j \in Lan \tag{27}$$

$$\delta_j \geq T_j - D_{1j} \quad \forall j \in Lan \tag{28}$$

$$\delta_j \geq 0 \tag{29}$$

Constraint sets (25) and (26) compute the deviation for departing aircraft, while (27) and (28) compute for landing aircraft. Finally, constraint set (29) is the domain constraint for the decision variable δ_j.

5. Solution methodology

In this section, we present two heuristic algorithms that use scheduling procedures (e.g., first available machine) to assign and sequence the aircraft at the vertiport components. In this way, we can handle larger problems and enhance practicality. The first heuristic algorithm addresses the makespan minimization model, and the second addresses the minimization of the deviation from the expected take-off/landing times.

5.1. Heuristic algorithm for minimizing the take-off or landing time of the last sequenced aircraft (makespan)

The outline of the heuristic algorithm is as follows: The first step is to generate a random solution seed. Next, from the solution seed and using scheduling rules, the algorithm assigns and sequences the aircraft at the vertiport components considering the blocking constraints and the separation times. Then, the schedule undergoes an improvement procedure that inserts the latest aircraft operation on the vertiport component with the largest idle time. Next, the insertion procedure schedules the remaining stages considering the new position of the aircraft operation. If an insertion is not feasible, the improvement procedure stops. The previous procedure is repeated until the stopping criterion of the maximum number of iterations is met. Finally, the algorithm chooses the solution with the best (lowest) makespan. Figure 3 presents the described heuristic algorithm as a flowchart. The details of each step are presented next.

5.1.1. Solution seed generation phase

To generate a solution seed, the heuristic algorithm determines a random vertiport component of the aircraft's first stage and then sequences the aircraft randomly in that component. In other words, for each aircraft in the take-off set $j \in D$, the solution seed determines a random taxiway 1 and sequences the take-off aircraft in a random position. Then, for each aircraft in the landing set $j \in L$, the solution seed selects a random TLOF pad and sequences the landing aircraft in a random position on the selected TLOF pad.

5.1.2. Scheduling phase

Once the solution seed is generated, the next step is to schedule the aircraft at the vertiport components as follows: First, the algorithm determines

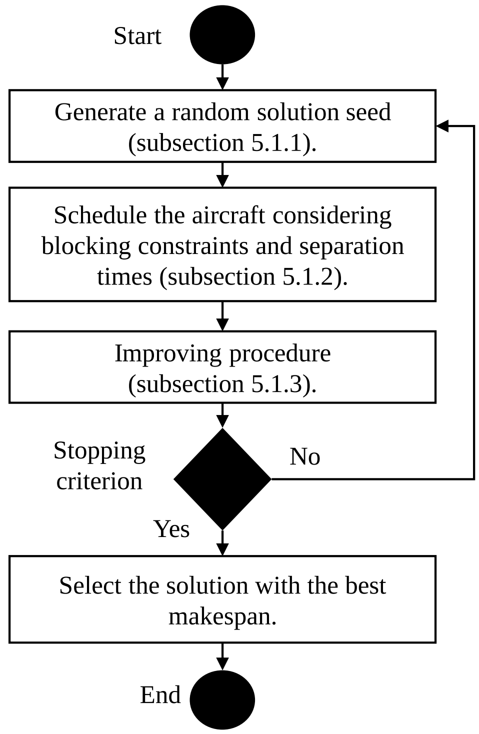

Figure 3: Proposed solution methodology for the makespan minimization problem

the first available vertiport component (FAVC) with an aircraft to be scheduled. Since no aircraft has been scheduled yet, all vertiport components are available. Therefore, in that case, and when ties are formed, the FAVC is chosen randomly. Next, the algorithm schedules the first aircraft on the solution seed of the selected FAVC. Then, the algorithm schedules the remaining aircraft stages by choosing the FAVC of the following stages, considering the completion times of the previous stages. Subsequently, the algorithm removes the aircraft from the solution seed. For scheduling the remaining aircraft in the solution seed, the algorithm repeats the previous instructions considering the blocking constraints and separation requirements (see Section 3). The pseudocode of Algorithm 1 presents the logic of the scheduling phase.

5.1.3. Improving phase

Once a potential schedule for the solution seed is generated, the next phase is to improve it. To do so, we implemented an insertion strategy that

Algorithm 1 Pseudocode of the scheduling procedure

```
for aircraft ∈ solution seed do
    Determine the first available vertiport component (FAVC)
    if FAVC ∈ Taxiway 1 then
        Schedule the first pending aircraft (take-off) in the FAVC
        for stage ← 2 to 4 do
            Determine the FAVC
            Schedule the aircraft considering the aircraft completion time of the last stage, the FAVC availability and separation rule at the TLOF pads stage
            if The aircraft to schedule has to wait for the FAVC of next stage to become available then
                Block the aircraft current vertiport component until it leaves
            end if
        end for
        Remove the aircraft from the solution seed
    else FAVC ∈ Stage 4
        Schedule the first pending aircraft (landing) in the FAVC considering the separation time
        for stage ← 2 to 4 do
            Determine the FAVC
            Schedule the aircraft considering the completion time of the last stage and the FAVC availability
            if aircraft to schedule has to wait for the FAVC to become available then
                Block the aircraft current vertiport component until it leaves
            end if
        end for
        Remove the aircraft from the solution seed
    end if
end for
```

is described in the following steps:

Step 1 The stage(s) with the least number of vertiport components is determined and the component with the longest idle time between consecutive aircraft is selected.

Step 2 The last scheduled aircraft operation of the selected component is identified and inserted at the beginning of the longest idle time of the component. The separation rule is considered if the selected component is in the TLOF pad stage. If insertion is not possible, the improvement procedure is finished.

Step 3 The remaining stages are scheduled considering the FAVC availability and the initial or completion time at the stages. If scheduling an aircraft in a stage is not feasible, insertion is impossible.

Step 4 The previous steps are repeated until an insertion is no longer possible.

5.1.4. Stopping criterion and solution selection

As the stopping criterion, we set a maximum number of iterations. The last step of the algorithm is selecting the solution with the best (lowest) makespan.

5.2. Heuristic algorithm for minimizing the deviation from the expected departure/arrival time

This heuristic algorithm follows the structure of the previous heuristic algorithm (summarized in Figure 3) with some modifications. First, the solution seed generation procedure now uses the overall slack time rule for contemplating the take-off/landing target times. Second, the algorithm generates feasible schedules using the procedure of subsection 5.1.2. Third, the heuristic improves the schedule by inserting the aircraft with the maximum take-off/landing deviation near its target time. The previous procedures are repeated until a stopping criterion is met. Finally, the algorithm selects the solution with the minimum total deviation (summation of the deviation of all aircraft). Next, we detail the heuristic algorithm steps.

5.2.1. Solution seed generation phase

The first step in this heuristic is to generate a solution seed using the overall slack time (OSL) rule considered in [42]. The OSL rule generates a permutation of n jobs $\pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$ according to their ascending overall slack times: $a_{\pi(j)} - \sum_{i=1}^{4} P_{i\pi(j)} \leq a_{\pi(j+1)} - \sum_{i=1}^{4} P_{i\pi(j+1)}$, where $\pi(j)$ specifies the job's position and $a_{\pi(j)}$ the take-off/landing target time. Ties are solved randomly. Next, for each aircraft $j \in A$, we determine a random vertiport component of its first stage and sequence j on that component according to the OSL rule. In other words, for each aircraft in the take-off set $j \in D$, we determine a random taxiway 1 and sequence the aircraft j according to the OSL rule. For each aircraft in the landing set $j \in L$, we determine a random TLOF pad and sequence aircraft j considering the OSL rule. In this way, we guarantee solution seeds with certain quality regarding the target times and diversity.

5.2.2. Improving phase

After generating the solution seed using the OSL rule and obtaining a feasible schedule using the procedure of subsection 5.1.2, the heuristic improves the schedule considering the objective of minimizing the deviation from the target take-off/landing time. To do so, we iteratively insert the aircraft with take-off/landing deviation near its target time as follows:

Step 1 A subset F of aircraft with take-off/landing deviation greater than 0 is determined.

Step 2 The aircraft with the maximum take-off/landing deviation from F is determined.

Step 3 The vertiport component with the longest idle time between consecutive aircraft is identified.

Step 4 An insertion of the operation of the aircraft determined in Step 2 is attempted at the beginning of the idle time identified in Step 3. Next, the remaining operations (stages) are scheduled considering the availability of the remaining stages and the initial or completion time at the stages. If the insertion is not possible or the insertion increases the deviation from the target take-off/landing time, the schedule of the aircraft determined in stage 2 is not modified. The aircraft determined in Step 2 is removed from F.

Step 5 Go to Step 2 until F is empty.

5.2.3. Stopping criterion and solution selection

Similar to the previous heuristic algorithm, as a stopping criterion, we define a maximum number of iterations. Once the iterations have been completed, the algorithm selects the solution with the lowest total deviation. The following section presents the experimental setting and the results.

6. Computational experiments and analysis of the results

In this section, we first present an illustrative example of a solution to show the features of the proposed approaches. Next, we present the experimental setup and the numerical results and analysis of the experimentation. Finally, this section presents some managerial implications of this work. The mathematical models of Section 4 were implemented in IBM ILOG CPLEX. The heuristic algorithms of Section 5 were implemented in Python. Experiments were run on a PC with Intel Core i7-10870H, 16 GB RAM and 2.2 GHz processor.

6.1. An illustrative example

Figure 4 presents the solution of an instance for the makespan minimization model. The processing times P_{ji} were generated following the generation scheme detailed in the next subsection. The instance was tested in a vertiport with three gates, three TLOF pads, one taxiway 1, and one taxiway 2. Aircraft 1 to 5 are departing and aircraft 6 to 10 are landing. As separation requirement we considered 90 s based on the study of [36]. We can observe in Figure 4 that the separation requirements between consecutive aircraft are met. In other words, aircraft depart or land at least 90 s after the previous aircraft has left the TLOF pad. Next, we detail the blocking constraints. Let us consider the scheduling of aircraft 8, which unloads using the first gate. When its unloading phase is completed, taxiway 1 is being used by aircraft 1. Therefore, the aircraft has a waiting time (denoted in gray) until the taxiway is available. Note that no other aircraft are scheduled in the waiting times. The makespan for this instance is 1118 s and is given by both aircraft 5 and 10.

Figure 5 presents the solution of an instance for minimizing the deviation from the target departure/arrival time. We used the same vertiport configuration and parameters as the previous example. In addition, we generated

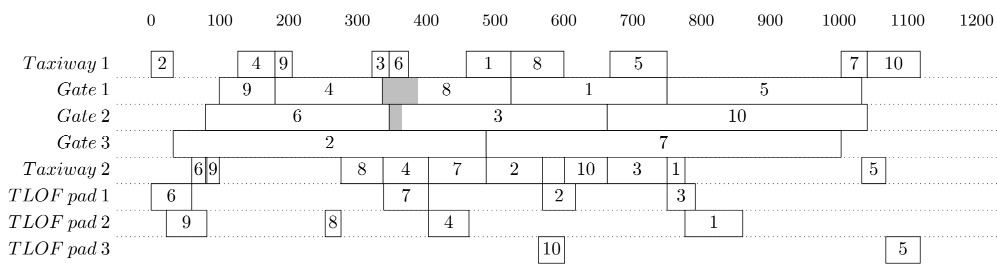

Figure 4: Gantt diagram of a sample instance solution for the objective of minimizing the take-off or landing time of the last sequenced aircraft (makespan)

the target times T_j using the scheme detailed in the following subsection. From Figure 5, we can observe that the separation rules and blocking constraints are met. The objective function (total deviation) has a value of 908 s, and the makespan of this schedule is now 1203 s. The next subsection presents the experimental design and the results of the computational study.

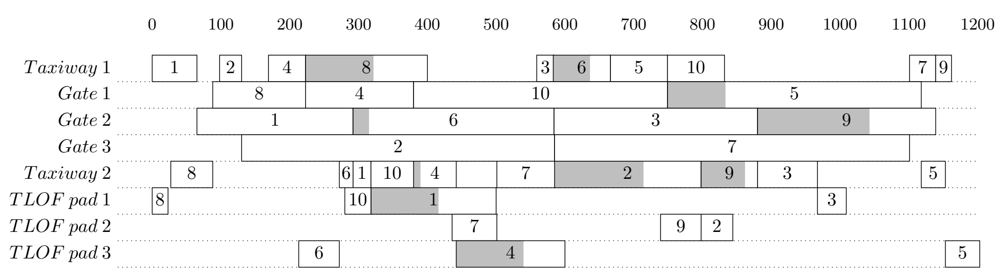

Figure 5: Gantt diagram of a sample instance solution for the objective of minimizing the deviation from the expected departure/arrival time

6.2. Experimental design

To the best of our knowledge, there are no benchmark instances available for the problem of scheduling aircraft at vertiport components. Thus, we created our own set of instances as follows: We consider $n = 10, 20, 30, 40$ and 50 aircraft with 30%, 50%, and 70% of them departing and the remaining aircraft landing. The processing times P_{ji} for each aircraft $j \in A$ at stage $i \in I$ are sampled from the uniform distribution $U[c_{ji}, d_{ji}]$ considering the times used by [35] as follows. For take-off aircraft $j \in D$, the time for entering the TLOF pad from the outside safety area and then liftoff ranges between

15-90 s. For landing aircraft, the time required for proceeding from the final approach fix, make the alignment above the TLOF pad and landing ranges between 15-90 s. The time required for loading or unloading passengers and cargo at gates ranges between 30-600 s for each aircraft $j \in A$. Last, the time required for using the taxiways in any direction ranges between 5-90 s.

The take-off/landing target times were obtained by adapting the generation procedure of [43] as follows. For each aircraft $j \in A$, the departure/landing target time T_j is drawn from $U[\beta_j, \beta_j + \beta_j\tau]$, where $\beta_j = \sum_{i \in I} P_{ji}$. In this computational study, we consider $\tau = 0.3n$. We set a separation time of 90 s based on the work of [36]. The previous instance set is tested in three different scenarios with one taxiway 1, one taxiway 2 and varying numbers of gates and TLOF pads, as shown in Table 2. As the stopping criterion for the heuristic algorithms, we set 100,000 iterations. For the MILP models, we set a maximum execution time of 1800 s to avoid excessive computation times. Next, we present the results and the analysis of the numerical experiments.

Table 2: Experimental scenarios

Scenario	Description	Number of gates and TLOF pads
1	Small-sized vertiport	2,2
2	Medium-sized vertiport	3,3
3	Large-sized vertiport	4,4

6.3. Results and analysis

Table 3 presents the computational results of the instance set tested in the vertiport of scenario 1. The first group of columns of Table 3 represents the number n of vehicles to schedule, the number of take-offs, and the number of landings. The second group of columns corresponds to the computational results of the MILP model for the first objective (makespan minimization) and its heuristic approach. The first column in this group reports a lower bound (LB) calculated using the approach of Santos et al. [44] summarized in Equation 30. We denote this lower bound as LBS in the results. Notably, this method for calculating the LB corresponds to the classic HFS problem, assuming that the processing times can be distributed perfectly in the machines at each stage. However, in this extension of the HFS problem, we consider additional characteristics such as the blocking constraints, separation rules at the TLOF pads, and two types of flow (departing and landing aircraft). Thus, even when the MILP reaches the optimal solution, there is a notable difference from the LBS. The following columns in the second group present the best solution found by the MILP model, the lower bound of the solution obtained by the solver CPLEX (LBCplex), the execution time, the optimality gap (OptGap) reported by the solver, the best solution found by the heuristic, and the heuristic execution time. The last columns in this first group report the Relative Error (RE) metric of the heuristic algorithm with respect to the MILP model (RE_{MILP}) and the RE metric of the heuristic algorithm with respect to the lower bound of [44] (RE_{LBS}). Notably, when $RE_{MILP} < 0$, we report the $RE_{LBCplex}$ metric, which compares the heuristic objective function value with the LB of the solution obtained by the solver ($RE_{LBCplex}$). This could give us an indication of how far the heuristic algorithm is from optimality.

The RE metrics are calculated using Equations 31, 32, and 33.

$$LBS = \max_{s \in I} \left\{ \min_{j \in A} \sum_{i=1}^{s-1} P_{ij} + \frac{1}{m_s} \sum_{j \in A} P_{kj} + \min_{j \in A} \sum_{i=s+1}^{k} P_{ij} \right\} \tag{30}$$

$$RE_{MILP} = \frac{HA_{sol} - MILP_{sol}}{MILP_{sol}} \tag{31}$$

$$RE_{LBS} = \frac{HA_{sol} - LBS}{LBS} \tag{32}$$

$$RE_{LBCplex} = \frac{HA_{sol} - LBCplex}{LBCplex} \tag{33}$$

The next group of columns presents and compares the results of the MILP model and the heuristic approach for the problem of minimizing the deviation from target arrival/departure time. We report the best solutions found by the MILP and the heuristic, the LB reported by the solver, the MILP OptGap, the execution times of both approaches, and the RE metrics of the heuristic algorithm with respect to the MILP solution and the solver LB. The solutions, lower bounds and CPU execution times are reported in s. Table 4 and Table 5 contain the same columns for reporting the results of the experimentation on scenarios 2 and 3, respectively. Next, we discuss the results.

Table 3: Computational results for the small-sized vertiport (scenario 1)

Instance			Minimization of the take-off/landing time of the last sequenced aircraft (makespan)										Minimization of deviation from target take-off/landing time							
				MILP				Heuristic		Comparison			MILP				Heuristic		Comparison	
n	Takeoff	Landing	LBS	Best Sol.	LB Cplex	CPU time	OptGap (%)	Best Sol.	CPU time	RE_{MILP} (%)	$RE_{LBCplex}$ (%)	RE_{LBS} (%)	Best Sol.	LB Cplex	CPU time	OptGap (%)	Best Sol.	CPU time	RE_{MILP} (%)	$RE_{LBCplex}$ (%)
	7	3	1058	1175	1175	8.15	0	1429	62.28	21.62	-	35.07	121	121	9.62	0	140	65.64	15.70	-
10	5	5	1058	1197	1197	13.14	0	1536	65.14	28.32	-	45.18	134	134	11.63	0	166	66.34	23.88	-
	3	7	1058	1167	1167	9.78	0	1499	67.31	28.45	-	41.68	242	242	17.39	0	298	63.9	23.14	-
	14	6	2627	3189	1165	1800	63.46	3172	113.74	-	172.27	20.75	594	80	1800	86.53	624	116.87	5.05	-
20	10	10	2627	3163	1119	1800	64.64	3133	114.65	-	179.98	19.26	788	133	1800	83.09	639	117.56	-	380.45
	6	14	2627	3123	1161	1800	62.82	3251	117.26	4.10	-	23.75	585	98	1800	83.19	627	121.64	7.18	-
	21	9	4790	5869	844	1800	85.62	5408	164.38	-	540.76	12.90	5028	505	1800	89.96	2980	166.81	-	490.10
30	15	15	4790	5957	920	1800	84.56	5357	168.17	-	482.28	11.84	4910	616	1800	87.45	2681	169.31	-	335.23
	9	21	4790	6354	785	1800	87.65	5176	160.32	-	559.36	8.06	6402	685	1800	89.30	4262	166.32	-	522.19
	28	12	6208	9595	787	1800	91.80	7185	244.51	-	812.96	15.74	16854	902	1800	94.65	8645	250.22	-	858.43
40	20	20	6208	9100	838	1800	90.79	7025	246.27	-	738.31	13.16	17456	1178	1800	93.25	10280	254.31	-	772.67
	12	28	6208	8126	850	1800	89.55	6762	245.91	-	695.53	8.92	19544	1151	1800	94.11	11164	250.30	-	869.94
	35	15	7183	-	-	1800	-	8386	277.95	-	-	16.75	-	-	1800	-	14476	285.07	-	-
50	25	25	7183	-	-	1800	-	8238	283.61	-	-	14.69	-	-	1800	-	16552	283.52	-	-
	15	35	7183	-	-	1800	-	8691	292.54	-	-	20.99	-	-	1800	-	18005	285.10	-	-

The first thing to notice in the results is the computational difficulty of the aircraft scheduling decisions on vertiports. In both problems, the MILP models could only find optimal solutions with instances of 10 aircraft. For large-sized instances, the MILP approaches reported solutions with a high optimality gap (reported by the solver) reaching the time limit. In addition, for some instances of 40 aircraft and all instances of 50 aircraft, the MILP models did not find feasible solutions within the execution time limit. This confirms the need to develop solution methodologies for solving real-life instances of the problems at hand. In addition, and not surprisingly, by increasing the number of aircraft, the heuristics spend more time obtaining solutions.

The results for the heuristic algorithm for the makespan minimization model indicate that it is outperformed by the MILP model in both computational time and objective function in the instances of 10 aircraft and in some instances with 20 aircraft. However, for the instances of 20 aircraft, the heuristic algorithm obtained similar results on average but in much less computational time.

In addition, the heuristic algorithm finds better solutions for larger instances ($n \geq 30$) than the MILP model. Regarding the RE_{MILP} and $RE_{LBCplex}$ metrics for the makespan minimization problem, we notice that when the MILP model achieves optimality ($n = 10$), the RE_{MILP} metric is on average 21.11%. In addition, when the MILP model does not achieve optimality (instances of $n > 10$), the solver reports high optimality gaps. Thus, the solver reports a low-quality LB (lower than the LB computed using the approach of [44]). Consequently, the $RE_{LBCplex}$ metric values are high in all scenarios, and unfortunately, it was not possible to have an idea of how far from optimality the heuristic algorithm is when the MILP does not reach optimality. In addition, the values of the RE_{LBS} metric, which compares the heuristic algorithm with the LB of [44], are high for the instances of $n = 10$. These unexpected values can be attributed to the fact that the LB of [44] does not account for the blocking constraints and separation requirements, which have a greater impact in smaller instances. For example, in the first instance of scenario 3, the difference between the LBS and the optimal solution is 24.9%. This difference contributes to the high value of the RE_{LBS}. This effect has less impact when scheduling more aircraft.

One limitation of our research is the lack of accurate lower bounds for evaluating the performance of the heuristic approach. Despite this limitation, our findings do suggest that this heuristic approach is helpful for scheduling

Table 4: Computational results for the medium-sized vertiport (scenario 2)

Instance				Minimization of the take-off/landing time of the last sequenced aircraft (makespan)									Minimization of deviation from target take-off/landing time							
				MILP				Heuristic		Comparison			MILP				Heuristic		Comparison	
n	Takeoff	Landing	LBS	Best Sol.	LB Cplex	CPU time	OptGap (%)	Best Sol.	CPU time	RE_{MILP} (%)	$RE_{LBCplex}$ (%)	RE_{LBS} (%)	Best Sol.	LB Cplex	CPU time	OptGap (%)	Best Sol.	CPU time	RE_{MILP} (%)	$RE_{LBCplex}$ (%)
	7	3	735	897	897	8.42	0	1098	71.27	22.41	-	49.39	98	98	3.77	0	122	73.72	24.49	-
10	5	5	735	905	905	6.52	0	1092	68.60	20.66	-	48.57	78	78	2.36	0	96	74.45	23.08	-
	3	7	735	910	910	8.15	0	1054	71.73	15.82	-	43.40	122	122	9.23	0	161	71.59	31.97	-
	14	6	1770	2234	782	1800	64.99	2215	119.15	-	183.25	25.14	234	84	1800	64.16	182	120.94	-	116.51
20	10	10	1770	2318	789	1800	65.99	2259	120.06	-	186.31	27.63	276	71	1800	74.23	166	127.73	-	133.44
	6	14	1770	2083	850	1800	59.24	2302	124.76	10.51	-	30.06	256	115	1800	55.15	186	121.02	-	61.92
	21	9	3209	4213	749	1800	82.22	3961	176.28	-	428.84	23.43	2157	544	1800	74.77	1329	182.37	-	144.24
30	15	15	3209	4998	776	1800	84.48	3808	175.41	-	390.72	18.67	1413	302	1800	78.66	987	180.12	-	226.66
	9	21	3209	4377	779	1800	82.2202	3863	175.94	-	395.89	20.38	3172	399	1800	87.41	734	176.36	-	83.87
	28	12	4157	5565	748	1800	86.55	5001	260.13	-	568.58	20.30	12895	1400	1800	89.14	8298	258.60	-	492.74
40	20	20	4157	5389	770	1800	85.71	4711	257.57	-	511.82	13.33	15632	1691	1800	89.18	12143	261.05	-	618.10
	12	28	4157	6381	842	1800	86.8	5005	256.86	-	494.42	20.40	14925	1596	1800	89.31	8019	255.44	-	402.46
	35	15	4808	-	-	1800	-	5895	287.49	-	-	22.61	-	-	1800	-	14476	288.52	-	-
50	25	25	4808	-	-	1800	-	5631	292.10	-	-	17.12	-	-	1800	-	16552	292.63	-	-
	15	35	4808	-	-	1800	-	5462	289.36	-	-	13.60	-	-	1800	-	18005	293.82	-	-

Table 5: Computational results for the medium-sized vertiport (scenario 3)

Instance			Minimization of the take-off/landing time of the last sequenced aircraft (makespan)										Minimization of deviation from target take-off/landing time							
				MILP				Heuristic		Comparison			MILP				Heuristic		Comparison	
n	Takeoff	Landing	LBS	Best Sol.	LB Cplex	CPU time	OptGap (%)	Best Sol.	CPU time	RE_{MILP} (%)	$RE_{LBCplex}$ (%)	RE_{LBS} (%)	Best Sol.	LB Cplex	CPU time	OptGap (%)	Best Sol.	CPU time	RE_{MILP} (%)	$RE_{LBCplex}$ (%)
	7	3	573	763	763	12.56	0	960	77.279	25.82	-	67.54	74	74	2.542	0	97	84.361	31.08	-
10	5	5	573	780	780	10.51	0	880	74.323	12.82	-	53.58	71	71	2.02	0	87	76.202	22.54	-
	3	7	573	827	827	14.717	0	944	74.312	14.15	-	64.75	99	99	3.239	0	124	79.558	25.25	-
	14	6	1348	1732	782	1800	54.84	1838	128.746	6.12	-	36.35	158	46	62.038	71.13	133	133.462	-	189.13
20	10	10	1348	1839	782	1800	57.47	1712	126.323	-	118.93	27.00	174	53	62.664	69.52	177	135.351	-	233.96
	6	14	1348	1808	849	1800	53.04	1741	137.265	-	105.06	29.15	166	31	52.655	81.46	116	130.751	-	274.19
	21	9	2420	4385	749	1800	82.92	2957	177.969	-	294.79	22.19	1813	182	1800	89.98	1130	186.91	-	520.88
30	15	15	2420	3775	784	1800	79.231	3029	188.709	-	286.35	25.17	1523	174	1800	88.56	1188	188.708	-	582.76
	9	21	2420	3918	784	1800	79.98	2854	182.433	-	264.03	17.93	1412	148	1800	89.52	866	191.044	-	485.14
	28	12	3104	-	-	1800	-	3799	270.483	-	-	22.39	-	-	1800	-	5952	267.193	-	-
40	20	20	3104	-	-	1800	-	3717	273.681	-	-	19.75	-	-	1800	-	6245	271.3	-	-
	12	28	3104	-	-	1800	-	3664	268.546	-	-	18.04	-	-	1800	-	6648	269.995	-	-
	35	15	3622	-	-	1800	-	4365	290.995	-	-	20.51	-	-	1800	-	11253	291.492	-	-
50	25	25	3622	-	-	1800	-	4331	298.483	-	-	19.57	-	-	1800	-	10235	299.11	-	-
	15	35	3622	-	-	1800	-	4326	295.445	-	-	19.44	-	-	1800	-	12425	305.377	-	-

a high volume of take-offs and landings in the vertiport taxiways, gates, and TLOF pads. Regarding the results of the heuristic approach for the problem of minimizing the deviation from target take-off/landing time, we observe the following: First, as in the makespan minimization model, the MILP model outperforms the heuristic algorithm in both objective function and execution time in the instances of 10 aircraft. As the instances grow in terms of the number of aircraft, the heuristic performance is better compared to the MILP model. Last, and not surprisingly, by increasing the instance sizes in terms of the number of aircraft and vertiport sizes, both approaches require more time to obtain solutions.

Additionally, in this computational analysis, we compare the performance of the proposed heuristic approaches. Since there is no algorithm specifically designed for the problems at hand, the presented algorithms are compared with existing scheduling algorithms. First, we compare the heuristic algorithm for the makespan maximization problem (H1) presented in subsection 5.1 with the well-known Nawaz–Enscore–Ham (NEH) algorithm [45]. Next, we compare the heuristic algorithm for minimizing the deviation from the target take-off/landing time (H2) of subsection 5.2 with the Earliest Due Date (EDD) list-scheduling algorithm (see [46]). Notably, the previous algorithms were adapted to meet the separation requirements at the TLOF pads and the blocking constraints at all stages. The comparison is made using the relative deviation index (RDI) performance measure. This index is used to compare the performance of two or more algorithms [47, 48, 49]. We calculate the RDI for the k^{th} instance using Equation 34.

$$RDI_k = \begin{cases} \frac{S_k - Min_k}{Max_k - Min_k} & if(Max_k - Min_k) \neq 0 \\ 0 \text{ otherwise} \end{cases} \tag{34}$$

Where S_k is the solution of the k^{th} instance obtained with the proposed heuristic, and Max_k and Min_k correspond to the best and worst solutions of the k^{th} instance using the algorithms under comparison, respectively. In other words, the best and worst objective function values that were obtained by all replications of one instance using the algorithms H1 and NEH for the makespan minimization problem and H2 and EDD for the minimization of the deviation from the target arrival/departure time problem.

We generated 30 problem instances for each combination of objective functions and the vertiport sizes of Table 2. We varied the number of aircraft ($n = 10, 20, 30, 40, 50$), with 50% of them departing and the remaining landing. We used the parameter generation scheme presented in subsection 6.2. Table 6 presents the average of the objective function values in s over the 30 problem instances for each vertiport size. In addition, Table 7 presents the average RDI of the 30 problem instances for each vertiport size. These results confirm the efficiency of the proposed heuristic algorithms for scheduling aircraft at vertiport components. In both problems, the heuristics are superior to the traditional scheduling algorithms. The improvement tends to increase when scheduling aircraft in vertiports with more infrastructure components. It shows that the proposed algorithms perform especially well on large-scale vertiports. Next, we present some managerial implications of this work.

Table 6: Comparison of the heuristic algorithms performance

Vertiport size	Average H1	Average NEH	Difference	Average H2	Average EDD	Difference
Small	5126.40	7367.72	43.72%	16042.23	26564.29	65.59%
Medium	3691.54	5581.68	51.20%	8953.21	15235.68	70.17%
Large	2048.78	3162.94	54.38%	5356.21	9632.23	79.83%

Table 7: RDI metric for the the algorithms under comparison

	H1 Vs NEH		H2 Vs EDD	
Vertiport size	Average RDI	Standard deviation	Average RDI	Standard deviation
Small	0.361	0.077	0.261	0.098
Medium	0.298	0.098	0.257	0.137
Large	0.287	0.106	0.224	0.126

6.4. Managerial implications

The rapid advances in eVTOL vehicles suggest that new air transportation services that use this technology will be a reality within the coming years. Currently, multiple companies from the public and private sectors are working on prototypes and developing AAM projects to enable new air transportation services. The imminent entry of such services poses multiple challenges to AAM decision-makers. One of these challenges is planning the operations of the AAM take-off and landing infrastructure known as vertiports. An efficient planning process of aircraft operations at vertiports requires the support of computational tools such as those presented in this work. Our proposal allows traffic controllers to propose efficient schedules that (i) maximize the vertiport throughput or (ii) minimize the total deviation from the expected take-off/landing times. In other words, our proposal can support vertiport planners in determining efficient slots for using common ground routes (taxiways), gates, and TLOF pads. In addition to supporting aircraft scheduling decisions, our proposal can help planners determine the vertiport capacity operating under efficient conditions.

7. Conclusions and future research

In this paper, we have studied the scheduling of take-off and landing aircraft at vertiports' TLOF pads, gates, and common ground routes (taxiways). The motivation for this study arises from the need to develop decision-making tools and support systems to be used in the operation planning of future vertiports. Recent advances and initiatives in AAM technologies indicate that new air services will be available in multiple fields (e.g., air taxis or emergency medical services). Therefore, vertiports could face dense air traffic scenarios where a high volume of aircraft has to be scheduled. If the scheduling is made efficiently while considering safety constraints, the AAM operations at vertiports can be fluid and safe. Our study tackles this scheduling problem considering essential components of AAM operations such as separation rules at TLOF pads, blocking constraints, and take-off and landing target times.

This work studied two important scheduling objectives in air transportation services: maximizing the vertiport throughput and minimizing the deviation from the expected departure/arrival time. To address the scheduling problem under these objectives, we first proposed two MILP models capable of solving optimally small sized instances (up to 10 aircraft) in short CPU times. However, when facing larger instances, the MILP models report low-quality solutions with high optimality gaps spending much CPU time. Thus, for facing larger problems and enhancing practicality, we developed two simple but efficient heuristic approaches. Our experiments prove that the heuristics approaches yield good results in terms of solution quality and computational time. In addition, by comparing the performance of the approaches, the heuristic approaches outperform the MILP model when solving instances of 20 and more aircraft.

From this study, multiple opportunities for future research arise. For instance, this approach can be used as the initial step for integrating additional

AAM decisions such as aircraft routing or vertiport capacity analysis. In addition, future research can contemplate changes in the take-off and landing processes. For instance, evaluating the effect of including a taxiway from the TLOF pads to the staging stand area for aircraft that do not load or unload cargo. Future research studies should examine different objective functions, such as minimizing the maximum tardiness or minimizing the number of late take-offs and landings. Moreover, future works can be inspired by previous research works of the traditional runway scheduling problem at airports. For instance, [50] considered different types of aircraft classes regarding their size and proposed a dynamic programming model for the aircraft landing problem at airports. This work can be extended to the vertiport scheduling problem by considering not only different sizes of eVTOL vehicles but also different sizes of gates and TLOF pads. In addition, there are multiple approaches to the runway scheduling problem at airports that can be adapted to the aircraft scheduling problem at vertiports. The work of [33] presents a complete review of such methodologies.

Another interesting direction for future research is exploring exact algorithms for solving this scheduling problem. Over the last years, there have been advances in exact methods applied to various optimization problems such as hub location [51] or scheduling [52]. The latter research addressed the Quay Crane Scheduling Problem in container terminals using Benders decomposition. This technique can be used to tackle the problem studied in this work. Here, we have proposed one of the first approaches to scheduling AAM aircraft in vertiport components. Future work could propose additional mathematical models and solution methodologies for the problem at hand that can improve the results of this study.

8. Funding Sources

This work was made under the project grant INGPhD-44-2021 awarded by the School of Engineering at Universidad de La Sabana, and by an Eiffel Excellence Scholarship from the French Ministry for Europe and Foreign Affairs awarded to the first author.

9. Declarations of interest

Declarations of interest: none.

References

[1] Airbus, Urban Air Mobility Taking urban transport into the sky, `https://www.airbus.com/en/innovation/zero-emission/urban-air-mobility` accessed 16 March 2023.

[2] European Commision, Urban Air Mobility (UAM), `https://smart-cities-marketplace.ec.europa.eu/action-clusters-and-initiatives/action-clusters/sustainable-urban-mobility/urban-air-mobility-uam` accessed 16 March 2023.

[3] NASA, Advanced Air Mobility Mission Overview, `https://www.nasa.gov/aam/overview/` accessed 16 March 2023.

[4] The Boeing Company, Urban Air Mobility Concept of Operations (ConOps), `https://www.boeing.com/innovation/con-ops/index.page` accessed 16 March 2023.

[5] Volocopter, Volocopter Opens First Public VoloCity Exhibition in Asia, `https://www.volocopter.com/newsroom/first-volocity-exhibition-asia/` accessed 16 March 2023.

[6] Hyundai, Uber and hyundai motor announce aerial ridesharing partnership, release new full-scale air taxi model at ces, `http://www.hyundai.com/worldwide/en/company/newsroom/ces-2020-hyundai-unveils-uam-s-a1-0000016375` accessed 16 March 2023.

[7] R. Goyal, A. Cohen, Advanced Air Mobility: Opportunities and Challenges Deploying eVTOLs for Air Ambulance Service, Applied Sciences 12 (3) (2022) 1183. `doi:10.3390/app12031183`.

[8] D. P. Thipphavong, R. Apaza, B. Barmore, V. Battiste, B. Burian, Q. Dao, M. Feary, S. Go, K. H. Goodrich, J. Homola, H. R. Idris, P. H. Kopardekar, J. B. Lachter, N. A. Neogi, H. K. Ng, R. M. Oseguera-Lohr, M. D. Patterson, S. A. Verma, Urban Air Mobility Airspace Integration Concepts and Considerations, in: 2018 Aviation Technology, Integration, and Operations Conference, American Institute of Aeronautics and Astronautics, 2018. `doi:10.2514/6.2018-3676`.

[9] A. P. Cohen, S. A. Shaheen, E. M. Farrar, Urban Air Mobility: History, Ecosystem, Market Potential, and Challenges, IEEE Transactions on Intelligent Transportation Systems 22 (9) (2021) 6074–6087. doi:10.1109/TITS.2021.3082767.

[10] C. Reiche, C. McGillen, J. Siegel, F. Brody, Are We Ready to Weather Urban Air Mobility (UAM)?, in: 2019 Integrated Communications, Navigation and Surveillance Conference (ICNS), Vol. 2019-April, IEEE, 2019, pp. 1–7. doi:10.1109/ICNSURV.2019.8735297.

[11] Research and Markets, Urban Air Mobility Market By Component (Infrastructure, Platform), By Operations (Piloted, Autonomous, Hybrid), By Range (Intercity, Intracity), and By Region, Forecasts to 2030, Tech. rep.

[12] J. Holden, N. Goel, Fast-Forwarding to a Future of On-Demand Urban Air Transportation (2016) 1–98
https://uberpubpolicy.medium.com/fast-forwarding-to-a-future-of-on-demand-urban-air-transportation-f6ad36950ffa, accessed 01 July 2022.

[13] B. I. Schuchardt, D. Geister, T. Lüken, F. Knabe, I. C. Metz, N. Peinecke, K. Schweiger, Air Traffic Management as a Vital Part of Urban Air Mobility—A Review of DLR's Research Work from 1995 to 2022, Aerospace 10 (1) (2023) 81. doi:10.3390/aerospace10010081.

[14] L. A. Garrow, B. J. German, C. E. Leonard, Urban air mobility: A comprehensive review and comparative analysis with autonomous and electric ground transportation for informing future research, Transportation Research Part C: Emerging Technologies 132 (2021) 103377. doi:10.1016/j.trc.2021.103377.

[15] H. Baik, A. A. Trani, N. Hinze, H. Swingle, S. Ashiabor, A. Seshadri, Forecasting Model for Air Taxi, Commercial Airline, and Automobile Demand in the United States, Transportation Research Record: Journal of the Transportation Research Board 2052 (1) (2008) 9–20. doi:10.3141/2052-02.

[16] S. Rajendran, J. Shulman, Study of emerging air taxi network operation using discrete-event systems simulation approach, Journal of

Air Transport Management 87 (July) (2020) 101857. doi:10.1016/j.jairtraman.2020.101857.

[17] S. Rajendran, J. Zack, Insights on strategic air taxi network infrastructure locations using an iterative constrained clustering approach, Transportation Research Part E: Logistics and Transportation Review 128 (January) (2019) 470–505. doi:10.1016/j.tre.2019.06.003.

[18] J. Jeong, M. So, H.-Y. Hwang, Selection of Vertiports Using K-Means Algorithm and Noise Analyses for Urban Air Mobility (UAM) in the Seoul Metropolitan Area, Applied Sciences 11 (12) (2021) 5729. doi:10.3390/app11125729.

[19] L. Chen, S. Wandelt, W. Dai, X. Sun, Scalable Vertiport Hub Location Selection for Air Taxi Operations in a Metropolitan Region, INFORMS Journal on Computing 34 (2) (2022) 834–856. doi:10.1287/ijoc.2021.1109.

[20] L. C. Willey, J. L. Salmon, A method for urban air mobility network design using hub location and subgraph isomorphism, Transportation Research Part C: Emerging Technologies 125 (2021) 102997. doi:10.1016/j.trc.2021.102997.

[21] H. Shin, T. Lee, H.-R. Lee, Skyport location problem for urban air mobility system, Computers & Operations Research 138 (2022) 105611. doi:10.1016/j.cor.2021.105611.

[22] I. C. Kleinbekman, M. A. Mitici, P. Wei, eVTOL Arrival Sequencing and Scheduling for On-Demand Urban Air Mobility, in: 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), IEEE, 2018, pp. 1–7. doi:10.1109/DASC.2018.8569645.

[23] P. Pradeep, P. Wei, Heuristic Approach for Arrival Sequencing and Scheduling for eVTOL Aircraft in On-Demand Urban Air Mobility, in: 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), IEEE, 2018, pp. 1–7. doi:10.1109/DASC.2018.8569225.

[24] I. C. Kleinbekman, M. Mitici, P. Wei, Rolling-Horizon Electric Vertical Takeoff and Landing Arrival Scheduling for On-Demand Urban Air Mobility, Journal of Aerospace Information Systems 17 (3) (2020) 150–159. doi:10.2514/1.I010776.

[25] S. H. Kim, Receding Horizon Scheduling of On-Demand Urban Air Mobility With Heterogeneous Fleet, IEEE Transactions on Aerospace and Electronic Systems 56 (4) (2020) 2751–2761. doi:10.1109/TAES.2019.2953417.

[26] C. Bosson, T. A. Lauderdale, Simulation Evaluations of an Autonomous Urban Air Mobility Network Management and Separation Service, in: 2018 Aviation Technology, Integration, and Operations Conference, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2018. doi:10.2514/6.2018-3365.

[27] J. Bertram, P. Wei, An Efficient Algorithm for Self-Organized Terminal Arrival in Urban Air Mobility, in: AIAA Scitech 2020 Forum, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2020. doi:10.2514/6.2020-0660.

[28] Q. Shao, M. Shao, Y. Lu, Terminal area control rules and eVTOL adaptive scheduling model for multi-vertiport system in urban air Mobility, Transportation Research Part C: Emerging Technologies 132 (2021) 103385. doi:10.1016/j.trc.2021.103385.

[29] C. Y. Justin, A. P. Payan, S. I. Briceno, B. J. German, D. N. Mavris, Power optimized battery swap and recharge strategies for electric aircraft operations, Transportation Research Part C: Emerging Technologies 115 (February) (2020) 102605. doi:10.1016/j.trc.2020.02.027.

[30] S. S. Chauhan, J. R. R. A. Martins, Tilt-Wing eVTOL Takeoff Trajectory Optimization, Journal of Aircraft 57 (1) (2020) 93–112. doi:10.2514/1.C035476.

[31] K. O. Ploetner, C. Al Haddad, C. Antoniou, F. Frank, M. Fu, S. Kabel, C. Llorca, R. Moeckel, A. T. Moreno, A. Pukhova, R. Rothfeld, M. Shamiyeh, A. Straubinger, H. Wagner, Q. Zhang, Long-term application potential of urban air mobility complementing public transport: an upper Bavaria example, CEAS Aeronautical Journal 11 (4) (2020) 991–1007. doi:10.1007/s13272-020-00468-5.

[32] A. Straubinger, R. Rothfeld, M. Shamiyeh, K.-D. Büchter, J. Kaiser, K. O. Plötner, An overview of current research and developments in

urban air mobility – Setting the scene for UAM introduction, Journal of Air Transport Management 87 (2020) 101852. doi:10.1016/j.jairtraman.2020.101852.

[33] S. Ikli, C. Mancel, M. Mongeau, X. Olive, E. Rachelson, The aircraft runway scheduling problem: A survey, Computers & Operations Research 132 (2021) 105336. doi:10.1016/j.cor.2021.105336.

[34] NUAIR, Crown Consulting, Mosaic ATM, NASA, High-Density Automated Vertiport Concept of Operations, https://ntrs.nasa.gov/citations/20210016168 accessed 01 July 2022.

[35] P. D. Vascik, R. J. Hansman, Development of Vertiport Capacity Envelopes and Analysis of Their Sensitivity to Topological and Operational Factors, in: AIAA Scitech 2019 Forum, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2019. doi:10.2514/6.2019-0526.

[36] K. H. Goodrich, B. Barmore, Exploratory Analysis of the Airspace Throughput and Sensitivities of an Urban Air Mobility System, in: 2018 Aviation Technology, Integration, and Operations Conference, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2018. doi:10.2514/6.2018-3364.

[37] J. A. Bennell, M. Mesgarpour, C. N. Potts, Airport runway scheduling, 4OR 9 (2) (2011) 115–138. doi:10.1007/s10288-011-0172-x.

[38] R. Ruiz, J. A. Vázquez-Rodríguez, The hybrid flow shop scheduling problem, European Journal of Operational Research 205 (1) (aug 2010). doi:10.1016/j.ejor.2009.09.024.

[39] H. Djellab, K. Djellab, Preemptive Hybrid Flowshop Scheduling problem of interval orders, European Journal of Operational Research 137 (1) (2002) 37–49. doi:10.1016/S0377-2217(01)00094-7.

[40] J. N. D. Gupta, Two-Stage, Hybrid Flowshop Scheduling Problem, The Journal of the Operational Research Society 39 (4) (1988) 359. doi:10.2307/2582115.

[41] A. Mollaei, M. Mohammadi, B. Naderi, A bi-objective MILP model for blocking hybrid flexible flow shop scheduling problem: robust possibilistic programming approach, International Journal of Management Science and Engineering Management 14 (2) (2019) 137–146. doi:10.1080/17509653.2018.1505565.

[42] Q.-K. Pan, R. Ruiz, P. Alfaro-Fernández, Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows, Computers & Operations Research 80 (2017) 50–60. doi:10.1016/j.cor.2016.11.022.

[43] S. Tanaka, S. Fujikuma, A dynamic-programming-based exact algorithm for general single-machine scheduling with machine idle time, Journal of Scheduling 15 (3) (2012) 347–361. doi:10.1007/s10951-011-0242-0.

[44] D. Santos, J. Hunsucker, D. Deal, Global lower bounds for flow shops with multiple processors, European Journal of Operational Research 80 (1) (1995) 112–120. doi:10.1016/0377-2217(93)E0326-S.

[45] M. Nawaz, E. E. Enscore, I. Ham, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, Omega 11 (1) (1983) 91–95. doi:10.1016/0305-0483(83)90088-9.

[46] M. L. Pinedo, Scheduling, Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-26580-3.

[47] S. Jun, J. Park, A hybrid genetic algorithm for the hybrid flow shop scheduling problem with nighttime work and simultaneous work constraints: A case study from the transformer industry, Expert Systems with Applications 42 (15-16) (2015) 6196–6204. doi:10.1016/j.eswa.2015.03.012.

[48] M. Akhshabi, R. Tavakkoli-Moghaddam, F. Rahnamay-Roodposhti, A hybrid particle swarm optimization algorithm for a no-wait flow shop scheduling problem with the total flow time, The International Journal of Advanced Manufacturing Technology 70 (5-8) (2014) 1181–1188. doi:10.1007/s00170-013-5351-9.

[49] A. Missaoui, Y. Boujelbene, An effective iterated greedy algorithm for blocking hybrid flow shop problem with due date window, RAIRO - Operations Research 55 (3) (2021) 1603–1616. doi:10.1051/ro/2021076.

[50] A. Lieder, D. Briskorn, R. Stolletz, A dynamic programming approach for the aircraft landing problem with aircraft classes, European Journal of Operational Research 243 (1) (2015) 61–69. doi:10.1016/j.ejor.2014.11.027.

[51] S. Wandelt, W. Dai, J. Zhang, X. Sun, Toward a Reference Experimental Benchmark for Solving Hub Location Problems, Transportation Science 56 (2) (2022) 543–564. doi:10.1287/trsc.2021.1094.

[52] D. Sun, L. Tang, R. Baldacci, A Benders decomposition-based framework for solving quay crane scheduling problems, European Journal of Operational Research 273 (2) (2019) 504–515. doi:10.1016/j.ejor.2018.08.009.

Highlights

A heuristic approach for scheduling advanced air mobility aircraft at vertiports

Julián Alberto Espejo-Díaz, Edgar Alfonso-Lizarazo*, Jairo R. Montoya-Torres

- We studied the advanced air mobility aircraft scheduling problem at vertiports.
- We considered separation rules at touchdown and lift-off pads and blocking constraints.
- Two mixed integer linear programming formulations are presented for optimally solving small instances.
- We propose two heuristic algorithms for solving real-life sized instances.
- The computational results provide insights into vertiport operations.