



Optimization of departure runway scheduling incorporating arrival crossings

Ji Ma, Mohammed Sbihi, Daniel Delahaye

► To cite this version:

Ji Ma, Mohammed Sbihi, Daniel Delahaye. Optimization of departure runway scheduling incorporating arrival crossings. *International Transactions in Operational Research*, 2021, 28 (2), pp.615-637. 10.1111/itor.12657 . hal-02064458

HAL Id: hal-02064458

<https://enac.hal.science/hal-02064458>

Submitted on 26 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimization of departure runway scheduling incorporating arrival crossings

Ji Ma^{a,b}, Mohammed Sbihi^a and Daniel Delahaye^a

^aLab ENAC, Ecole Nationale de l'Aviation Civile, 7 Av. Edouard Belin, Toulouse, 31400, France

^bSIAE, Civil Aviation University of China, 2898 Jinbei Road, Tianjin, 300300, China

E-mail: ji.ma@recherche.enac.fr [Ma]; mohammed.sbihi@enac.fr [Sbihi]; daniel@recherche.enac.fr [Delahaye]

Abstract

The runway is a key airport resource and an efficient runway operation is critical to enhance the airport efficiency and to reduce delays. This paper addresses the problem of scheduling aircraft departures incorporating arrival crossings. Constraints for wake turbulence separations, flight time window restrictions, and holding queue capacity at runway threshold are explicitly considered. We present two Integer Linear Programming (ILP) models and a simulated annealing (SA) algorithm. Comparison tests are conducted for the south side of Paris Charles De Gaulle Airport and show that the three proposed methods can significantly improve the solutions based on the simple first-come-first-served rule. Moreover, SA has sufficiently small computation time for real-time deployment.

Keywords: Aircraft Sequencing Problem; ILP; Simulated Annealing

1. Introduction

Continuous air traffic growth has led to congestions in many major airports. Limited capacity is the major cause of congestion. Surface congestion may result in significant delays and operational inefficiencies. Of all airport ground resources, the runway is the main bottleneck, as it is shared by all types of operations (arrivals, departures and crossings) indicated by field observations (Anagnostakis, 2004). Efficient runway operations planning and control is critical to enhance the airport efficiency and to reduce delays while satisfying all system constraints. One important aspect of runway operations is the departure sequencing and scheduling, termed by departure management (DMAN).

The DMAN is a tactical planning tool supporting the departure schedule. It assists in optimizing take-off time while ensuring safety by taking into account several operational constraints: the minimum wake turbulence separation between successive aircraft, the time window of runway usage for individual flights, the interactions with landing runways with regard to airport layout, etc. The minimum separation requirement is a key factor for deciding the runway capacity. The International Civil Aviation Organization (ICAO) divides aircraft into four different wake turbulence weight categories (Light, Medium, Heavy, and Super Heavy) based on the maximum certificated take-off mass. Due to wake vortex, an

aircraft is perturbed by the precedent one and a minimum separation must be respected to ensure safety. The separation time depends on the categories of the leading and trailing aircraft. For example, a Heavy aircraft followed by a Light aircraft generates longer separation than a Light aircraft followed by a Heavy aircraft. The time window constraint represents the earliest and latest time of arrival at the runway threshold. The time window can be applied to both the flights with CFMU (Central flow management Unit) slots and the flights without CFMU slots to ensure fairness. CFMU slot is a take-off slot that can be assigned to a departing flight in order to avoid congestion in some particular airspace. It is defined as the period between 5 minutes before and 10 minutes after the CTOT (Calculated Take-Off Times). It can be transferred to the gate by subtracting the taxi time. For the flights without CFMU slots, the earliest and latest time of the time window represent the EOBT (Estimated Off-Block Time) and a maximum gate holding time with regard to the EOBT, respectively.

The Estimated Off-Block Time (EOBT) is the estimated time when an aircraft will be ready after all doors are closed, boarding bridge removed, and is ready to push back upon reception of clearance from controller. Next, a Target Startup Approval Time (TSAT) is the time provided by controller when an aircraft can expect to receive push back approval taking into account EOBT, and the traffic situation etc. TSAT assignment is important to reduce aircraft waiting time with engine on. If many aircraft arrive at the runway threshold waiting for take-off, a long queue is formed and leads to extra fuel burn. It is more efficient to delay departures before the engines start up so that the traffic arrives at the runway smoothly.

Air traffic controller officers are responsible for sequencing traffic and ensuring minimum separations for departures. In large airports, departure management is handled by three types of controllers: the apron controller (or ramp controller), the ground controller, and the local controller. The apron controller dictates aircraft push back time from the gate and directs aircraft into gates after landing. They also coordinate with airline operators in implementing ground delays if necessary. The ground controller handles all intermediary taxiing routes to avoid conflicting movements of aircraft, and gives the pilot instructions on reaching the runway holding area via the appropriate taxiways. The local controller clears aircraft for take-off or landing ensuring the prescribed runway separation.

Runway can be used in segregated mode (only landings or only take-offs) or in mixed mode (both take-offs and landings). The mixed mode can achieve higher runway throughput since separation between arrival and departure is shorter. Nevertheless, segregated mode is more often used due to airport layout. Departures and arrivals are highly coupled processes with complex interactions with regard to runway configurations, weather, mix of aircraft, etc. Departing and arrival flights may cross an active runway. Furthermore, controllers delay runway-crossing clearances until a group of aircraft has accumulated at various holding points (Anagnostakis, 2004). However, there is an upper limit on how many aircraft can hold at the same crossing point. For safety reasons, arrivals always have priority over departures and crossings.

Aircraft Sequencing Problem (ASP) has been studied intensively in the past decades. Research has been conducted both in the US (Anagnostakis et al., 2000; Balakrishnan and Chandran, 2007; Gupta et al., 2010) etc. and in Europe (Böhme, 2005; Atkin et al., 2007; Deau et al., 2009) etc. Bennell et al. (2011) presented an extensive overview on landing and take-off scheduling problems. Lieder and Stolletz (2016) recently summarized different models and solution approaches on the ASP that consider heterogeneous or interdependent runways, featuring articles up to 2015.

ASP can be divided into Aircraft Landing Problem (ALP), Aircraft Take-off Problem (ATP), and a combination of both problems. Numerous physical and operational constraints as well as the layout of

runway system must be considered. Various exact and heuristic approaches have been proposed. The solution quality and the computational time of the approach are two important factors. In real situations, controllers can use only algorithms which can quickly find a good solution. Optimal solutions arising from lengthy computation times are of little practical use (Bennell et al., 2011).

In the rest of this section, we mainly review the ATP and its interactions with the arrival aircraft. Considering the exact approach, Balakrishnan and Chandran (2006, 2007, 2010) introduce a dynamic programming algorithm with Constrained Position Shift (CPS) that set a limit on the number of positions an aircraft can occupy in the sequence for the ATP based on their model for the ALP. They also provide complexity analyses of their algorithms and extend their approach for multiple runways and active runway crossings. Montoya et al. (2014) formulate ATP as a multi-objective optimization problem with respect to total aircraft delay and runway throughput and use dynamic programming to solve. Gupta et al. (2009) present a Mixed Integer Linear Program (MILP) to handle the departure queuing area. A following work by the same authors (Gupta et al., 2010) presented a MILP formulation for incorporating active runway crossings in departure scheduling. However, the computational performance was not good due to the lack of good lower bounds. Moreover, the MILP formulation only provides optimal solution for each short time frame (e.g. 15 minutes), which is addressed and improved in Bae et al. (2013).

Different heuristic approaches have been proposed. Atkin et al. (2008a) applied three different heuristic methods for the ATP taking into account some specific constraints concerning London Heathrow airport. Further work can be found in (Atkin et al., 2007, 2008b, 2009). Anagnostakis and Clarke (2003) proposed a two-stage heuristic algorithm to solve the departure and runway crossing problem. Mori (2017) applied a tabu search technique to develop a pushback time assignment algorithm in order to reduce take-off delay.

Regarding comparative studies between exact and heuristic approaches, Furini et al. (2015) present two MILP models and a tabu search heuristic for finding solutions to the ASP in a short computation time. With a time limit of 15 seconds, the tabu search heuristic outperforms the two MILP models. Malik and Jung (2016) proposed one exact and two heuristic algorithms to solve the ATP, and they conclude that heuristics are more suitable than the exact approach to produce good quality solutions in a relatively short computation time. D'Ariano et al. (2010) compare branch and bound (BB) and heuristic algorithms, setting a time limit of two minutes for the algorithm. Their results show that BB outperforms their heuristics for instances up to 60 minutes of traffic prediction.

In this paper we focus on the departure sequencing problem incorporating arrival crossings. Different from the literature works, besides specific constraints such as wake turbulence separations, flight time window restrictions etc, we consider explicitly the holding queue capacity at runway threshold. Moreover, holding point for arrivals is introduced as a decision variable to allow simultaneous crossings. Gate and runway threshold are two control points to hold departures. Two ILP models and a metaheuristic method are proposed to compare the solution quality and the computational performance. The paper is organized as follows. In Section 2, the problem statement is presented. In Section 3, two ILP formulations are proposed in detail, followed by the metaheuristic method in Section 4. Finally, computational results and conclusions are summarized in Section 5 and Section 6 respectively.

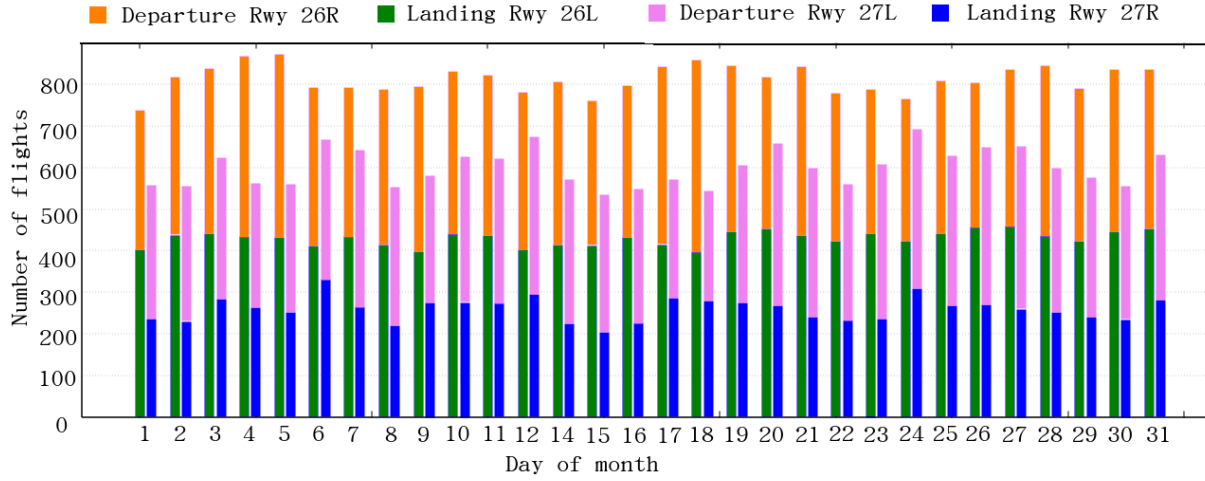


Fig. 1: Paris CDG airport runway throughput chart in July 2017. Southern pair of runways (26L/26R) is in higher demand compared to northern side (27L/27R). July 13th is excluded due to abnormal data record.

2. The Problem Statement

Paris CDG airport is one of the busiest passenger airports in Europe, composed of four parallel runways (two for landings and two for departures) and three terminals. Fig. 1 illustrates one month runway throughput of CDG in July 2017; it handled more than 1,400 flights/day. The southern pair of runways is in higher demand compared to the northern side due to the fact that it is closer to the main terminal (Terminal 2) of CDG. In this paper, we focus on the departure scheduling problem incorporating arrival crossings on the southern pair of runways: 26L for arrival and 26R for departure. As shown in Fig. 2, there are three main landing runway exits: V2, V3, and V4 and three holding points before crossing: S1, S2, and S3. In practice, the landing runway exit is chosen by the pilot with regard to the aircraft type and how fast it can brake after touching down. Controllers prefer to give priority to arrival aircraft to cross the departure runway without stopping when the departure runway is not fully occupied. When the departure traffic demand is high, controllers indicate arrival aircraft to wait at different holding points and to cross simultaneously. For departures, controllers can hold aircraft at the gate with engines off, or have them waiting at the runway threshold before taking off. In order to ensure safety, the minimum wake turbulence separation between successive aircraft must be respected. Moreover, several operational constraints such as time window of runway usage for individual flights, maximum runway queue capacity, and interactions between arrival crossings and departures need to be considered.

To be more precise, let $\mathcal{F} = \mathcal{A} \cup \mathcal{D}$ be the set of aircraft (arrivals and departures) that will use the runway during the time horizon $\mathcal{T}$, which is discretized into time intervals $= \{1, \dots, T\}$. Flight taking off/crossing at the time interval $t \in \mathcal{T}$ refers that it occurs at the beginning of time interval t .

We need to define the following notations for our problem formulation:

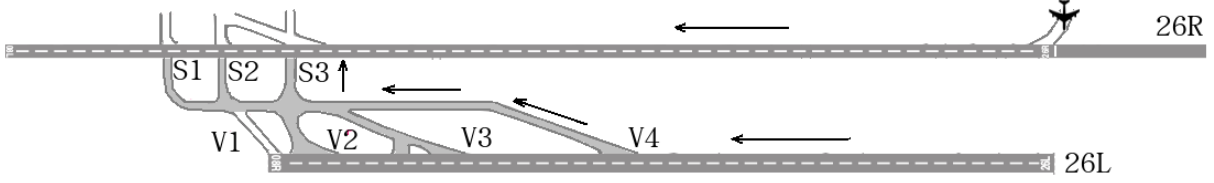


Fig. 2: CDG south side runway layout. Runway 26L is for landing and 26R is for taking off. The arrival aircraft have to cross the departure runway to reach the taxi area. Three arrival holding points before crossing 26R: S1, S2, and S3. Three main runway exits: V2, V3, and V4.

$\mathcal{F}$	set of flights, $\mathcal{F} = \mathcal{A} \cup \mathcal{D}$,
$\mathcal{A}$	set of arrivals,
$\mathcal{D}$	set of departures,
$\mathcal{E}$	set of landing runway exits, $\mathcal{E} = \{V2, V3, V4\}$,
$\mathcal{H}$	set of holding points before crossing departure runway, $\mathcal{H} = \{S1, S2, S3\}$,
$d_{e,h}$	taxi time from runway exit $e \in \mathcal{E}$ to holding point $h \in \mathcal{H}$, shown in Table 1,
C_d	departure capacity at the runway threshold,
C_a	arrival capacity at holding points,
ρ	time slot, which is a multiple of time intervals,
P_i	scheduled pushback time for flight $i \in \mathcal{D}$,
X_i	taxi time from spot to runway threshold for flight $i \in \mathcal{D}$,
T_i^0	initial runway usage time for flight $i \in \mathcal{D}$, $T_i^0 = P_i + X_i$,
R_i^p	maximum pushback delay (in number of slots) for departure flight $i \in \mathcal{D}$,
R_i^w	maximum holding time (in number of slots) for departure flight $i \in \mathcal{D}$,
R_i	maximum delay (in number of slots) for departure flight $i \in \mathcal{D}$, $R_i = R_i^p + R_i^w$,
$\mathcal{R}_i^p$	set of possible pushback delays for flight $i \in \mathcal{D}$, $\mathcal{R}_i^p = \{1, \dots, R_i^p\}$,
$\mathcal{R}_i$	set of possible total delays for flight $i \in \mathcal{D}$, $\mathcal{R}_i = \{1, \dots, R_i\}$,
t_i^T	time arriving at the runway threshold for flight $i \in \mathcal{F}$,
T_{ir}	possible runway usage time for flight $i \in \mathcal{D}$, $T_{ir} = T_i^0 + r\rho$, $r \in \mathcal{R}_i$,
L_i	scheduled landing time for flight $i \in \mathcal{A}$,
O_i	runway occupancy time for flight $i \in \mathcal{A}$,
e_i	runway exit point for flight $i \in \mathcal{A}$,
T_{ih}^0	initial runway usage time for flight $i \in \mathcal{A}$ if i uses the holding point h , $T_{ih}^0 = L_i + O_i + d_{e_i,h}$,
R_i^a	maximum holding time (in number of slots) for flight $i \in \mathcal{A}$,
$\mathcal{R}_i^a$	set of possible holding delays for flight $i \in \mathcal{A}$, $\mathcal{R}_i^a = \{1, \dots, R_i^a\}$,
T_{ihr}	possible runway usage time for flight $i \in \mathcal{A}$ if i uses the holding point h , $T_{ihr} = T_{ih}^0 + r\rho$, $r \in \mathcal{R}_i$,
s_{ij}	minimum separation in runway usage time of two successive flights i and j with regard to their wake turbulence categories, as shown in Table 2,
M	large positive constant.

Our aircraft runway scheduling consists in:

- Choosing a pushback delay d_i , i.e., the difference between the EOBT and the modified pushback time, and a holding duration w_i at the runway threshold for each departure $i \in \mathcal{D}$. With these decisions, we know the arrival time at the runway threshold $t_i^T = P_i + d_i + X_i$, and the runway usage time $t_i = t_i^T + w_i$;
- Choosing a holding point h_i and a holding duration w_i for each arrival $i \in \mathcal{A}$. With these decisions, we

Table 1: Average time from runway exit of 26L to holding point, $T_{e,h}$ (in seconds). The average values are calculated based on one-month real traffic data.

	S1	S2	S3
V2	104	70	59
V3	118	102	75
V4	158	142	115

Table 2: Single-runway separation requirements for departures and arrival crossings (in seconds).

Category		Trailing Aircraft g			
		<i>Heavy</i>	<i>Medium</i>	<i>Light</i>	<i>Crossing</i>
Leading Aircraft f	<i>Heavy</i>	96	120	120	60
	<i>Medium</i>	60	60	60	60
	<i>Light</i>	60	60	60	60
	<i>Crossing</i>	40	40	40	10

know the arrival time at holding point $t_i^T = L_i + O_i + d_{e_i, h_i}$, and the runway usage time $t_i = t_i^T + w_i$.

In accordance with the operational practices, pushback delay and holding time are given as multiples of time slot ρ . A runway schedule is feasible if and only if for each pair of distinct flights $(i, j) \in \mathcal{F} \times \mathcal{F}$ with $t_i \leq t_j$, we have

- $t_j - t_i \geq s_{ij}$ and $t_j^T \geq t_i^T$ if $(i, j) \in \mathcal{D} \times \mathcal{D}$, i.e., the minimum wake turbulence separation is respected, and the first aircraft arriving at the holding point departs first;
- $t_j - t_i \geq s_{ij}$ and $t_j^T \geq t_i^T$ if $(i, j) \in \mathcal{A} \times \mathcal{A}$ and $h_i = h_j$, i.e., two aircraft using the same holding point follow the FCFS order, and the minimum separation is respected;
- $t_j - t_i \geq s_{ij}$ if $(i, j) \in \mathcal{A} \times \mathcal{D} \cup \mathcal{D} \times \mathcal{A}$, i.e., the minimum separation is guaranteed between arrival and departure, vice versa.

Our problem consists of finding a feasible schedule while minimizing the total delay $\sum_{i \in \mathcal{D}} (d_i + w_i) + \sum_{i \in \mathcal{A}} w_i$.

The subproblem of scheduling aircraft take-offs can be seen as a machine scheduling problem with release times, due dates, and sequence-dependent processing times (but zero setup times) (Chaudhry and Khan, 2016). A related problem, the simple total tardiness problem (without sequence dependent setup times), has been shown to be NP-hard (Du and Leung, 1990). Consequently, our runway scheduling problem is NP-hard too.

3. The ILP Formulations

Two ILP formulations are given in this section. First, we present Model A that uses the number of time slots as decision. This model involves many so called “big-M” constraints that makes the LP relaxation rather weak. In order to reduce this number and to strengthen the formulation, we propose Model B termed as delay-indexed formulation.

3.1. Model A: time slot based formulation

The Decision Variables. We define the following decision variables:

$d_i \equiv$ number of pushback delay slots for departure flight $i \in \mathcal{D}$;

$w_i \equiv$ number of holding time slots for flight $i \in \mathcal{F}$;

$t_i \equiv$ runway usage time for flight $i \in \mathcal{F}$;

$t_i^T \equiv$ time arriving at the runway threshold for flight $i \in \mathcal{F}$;

$\delta_{ij} = \begin{cases} 1, & \text{if } i \text{ uses runway before } j, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{F}, j \in \mathcal{F}, i \neq j;$

$\alpha_{is} = \begin{cases} 1, & \text{if } i \text{ arrives at the runway threshold before time } s, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{D}, s \in \mathcal{T};$

$\beta_{is} = \begin{cases} 1, & \text{if } i \text{ uses runway after time } s, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{D}, \forall s \in \mathcal{T};$

$z_{ih} = \begin{cases} 1, & \text{if } i \text{ crosses runway via holding point } h, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{A}, h \in \mathcal{H};$

$\beta_{ih} = \begin{cases} 1, & \text{if } i \text{ crosses via holding point } h \text{ after time } s, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{A}, \forall h \in \mathcal{H}, \forall s \in \mathcal{T}.$

The Objective Function. Our objective is to minimize the total delay:

$$\text{Min } \sum_{i \in \mathcal{D}} (d_i + w_i) + \sum_{i \in \mathcal{A}} w_i.$$

The Constraints.

$$\sum_{h \in \mathcal{H}} z_{ih} = 1, \quad \forall i \in \mathcal{A}, \tag{1}$$

$$t_i^T = T_i^0 + \rho d_i, \quad \forall i \in \mathcal{D}, \tag{2}$$

$$t_i^T = \sum_{h \in \mathcal{H}} T_{ih}^0 z_{ih}, \quad \forall i \in \mathcal{A}, \tag{3}$$

$$t_i = t_i^T + \rho w_i, \quad \forall i \in \mathcal{F}, \tag{4}$$

$$\delta_{ij} + \delta_{ji} = 1, \quad \forall i, j \in \mathcal{F}, i < j, \tag{5}$$

$$t_j^T - t_i^T \geq -M(1 - \delta_{ij}), \quad \forall i, j \in \mathcal{D}, i \neq j, \tag{6}$$

$$t_j - t_i \geq s_{ij} - M(1 - \delta_{ij}), \quad \forall i, j \in \mathcal{D}, i \neq j, \quad (7)$$

$$t_j^T - t_i^T \geq -M(3 - \delta_{ij} - z_{ih} - z_{jh}), \quad \forall h \in \mathcal{H}, \forall i, j \in \mathcal{A}, i \neq j, \quad (8)$$

$$t_j - t_i \geq s_{ij} - M(3 - \delta_{ij} - z_{ih} - z_{jh}), \quad \forall h \in \mathcal{H}, \forall i, j \in \mathcal{A}, i \neq j, \quad (9)$$

$$t_j - t_i \geq s_{ij} - M(1 - \delta_{ij}), \quad \forall (i, j) \in \mathcal{D} \times \mathcal{A} \cup \mathcal{A} \times \mathcal{D}, \quad (10)$$

$$\alpha_{is} \leq \alpha_{i(s+1)}, \quad \forall i \in \mathcal{D}, \forall s \in \mathcal{T} \setminus \{T\}, \quad (11)$$

$$\beta_{is} \geq \beta_{i(s+1)}, \quad \forall i \in \mathcal{D}, \forall s \in \mathcal{T} \setminus \{T\}, \quad (12)$$

$$\beta_{ihs} \geq \beta_{ih(s+1)}, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \forall s \in \mathcal{T} \setminus \{T\}, \quad (13)$$

$$t_i - \sum_{s \in \mathcal{T}} \beta_{is} = 0, \quad \forall i \in \mathcal{D}, \quad (14)$$

$$t_i^T - \sum_{s \in \mathcal{T}} (1 - \alpha_{is}) = 0, \quad \forall i \in \mathcal{D}, \quad (15)$$

$$\sum_{s \in \mathcal{T}} \beta_{ihs} \leq M z_{ih}, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \quad (16)$$

$$t_i - \sum_{h \in \mathcal{H}} \sum_{s \in \mathcal{T}} \beta_{ihs} = 0, \quad \forall i \in \mathcal{A}, \quad (17)$$

$$\sum_{i \in \mathcal{D}} (\beta_{is} + \alpha_{is} - 1) \leq C_d, \quad \forall s \in \mathcal{T}, \quad (18)$$

$$\sum_{i \in \mathcal{A} | T_{ih}^0 \leq s} \beta_{ihs} \leq C_a, \quad \forall s \in \mathcal{T}, \forall h \in \mathcal{H}, \quad (19)$$

$$0 \leq d_i \leq R_i^p, \text{ integer}, \quad \forall i \in \mathcal{D}, \quad (20)$$

$$0 \leq w_i \leq R_i^w, \text{ integer}, \quad \forall i \in \mathcal{D}, \quad (21)$$

$$0 \leq w_i \leq R_i^a, \text{ integer}, \quad \forall i \in \mathcal{A}, \quad (22)$$

$$t_i^T, \text{ integer}, \quad \forall i \in \mathcal{F}, \quad (23)$$

$$t_i, \text{ integer}, \quad \forall i \in \mathcal{F}, \quad (24)$$

$$\delta_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{F}, i \neq j, \quad (25)$$

$$\alpha_{is} \in \{0, 1\}, \quad \forall i \in \mathcal{D}, \forall s \in \mathcal{T}, \quad (26)$$

$$\beta_{is} \in \{0, 1\}, \quad \forall i \in \mathcal{D}, \forall s \in \mathcal{T}, \quad (27)$$

$$z_{ih} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \quad (28)$$

$$\beta_{ihs} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \forall s \in \mathcal{T}. \quad (29)$$

Constraints (1) ensure only one holding point for each arrival. Constraints (2) link the time arriving at runway threshold and the pushback delay for departures. Constraints (3) link the time arriving at holding point and the holding point decision variables for arrivals. Constraints (4) state the runway usage time as a sum of time to reach the holding point and the holding time. Constraints (5) guarantee that given any two aircraft, one leads the other. Constraints (6) and (7) ensure the first-come-first-served (FCFS) order at the runway threshold and for take-off as well as the minimum separation for departures. There are two cases to consider here: if $\delta_{ij} = 1$, i.e., i uses runway before j , then Constraints (6) and Constraints (7) become $t_j^T \geq t_i^T$ and $t_j - t_i \geq s_{ij}$, ensuring that the FCFS order and the separation are respected. If

$\delta_{ij} = 0$, j uses runway before i , then Constraints (6) and Constraints (7) become $t_j^T - t_i^T \geq -M$ and $t_j - t_i \geq s_{ij} - M$, i.e., $t_j^T - t_i^T$ and $t_j - t_i$ are bigger than some large negative values, thereby ensuring that this constraint is effectively inactive. Similarly, Constraints (8) and Constraints (9) guarantee that at the same holding point, the crossing order of two landing aircraft keeps the same with the order they reach the holding point, and ensure the minimum separation while crossing. Constraints (10) guarantee the minimum separation between arrival and departure (vice versa). Constraints (11), (12) and (13) ensure the connectivity in time. For example, if flight i arrives at the runway threshold before time s , then for all the time intervals $s' \geq s$, $\alpha_{is'}$ has to have a value of 1. Constraints (14) and Constraints (15) link the two binary variables to the runway usage time and the arrival time at runway threshold for departures. Constraints (16) ensure that $\beta_{ihs} = 0$ if flight i does not use holding point h . Constraints (17) is similar to Constraints (14), that link the binary variable β_{ihs} to the runway usage time for arrivals. Constraints (18) ensure that at every time the number of aircraft waiting at the departure runway threshold will not exceed the departure capacity. The term $\beta_{is} + \alpha_{is} - 1$ is equal to 1 if and only if flight i is waiting at runway threshold during the time interval s . Likewise, Constraints (19) ensure that at every time the number of crossing aircraft waiting at the holding point will not exceed the holding capacity. Notice that, contrary to the departures, we do not need to introduce variables α_{ihs} for arrivals since the arrival time at runway threshold is known as soon as the holding point is given.

Considering the LP relaxation of the problem, we would like M to be as small as possible, and this can be achieved by tailoring M for each constraint. For example in Constraints (6), M can be set to $\max(0, T_i^0 + R_i^p - T_j^0)$. Moreover, as we know the time range where each aircraft can use the runway, we can strengthen the formulation by fixing the value of some variables:

$$\begin{aligned} \alpha_{is} &= \begin{cases} 1, & \text{for } s \in \{T_{iR_i^p}, \dots, T\}, \\ 0, & \text{for } s \in \{1, \dots, T_i^0 - 1\}; \end{cases} \quad \forall i \in \mathcal{D}; \\ \beta_{is} &= \begin{cases} 1, & \text{for } s \in \{1, \dots, T_i^0\}, \\ 0, & \text{for } s \in \{T_{iR_i} + 1, \dots, T\}; \end{cases} \quad \forall i \in \mathcal{D}; \\ \beta_{ihs} &= 0, \text{ for } s \in \{0, \dots, T_{ih}^0 - 1\}; \quad \forall i \in \mathcal{A}, h \in \mathcal{H}. \end{aligned}$$

3.2. Model B: delay-indexed formulation.

The Decision Variables. We define the following binary decision variables:

$$\begin{aligned} x_{ir} &= \begin{cases} 1, & \text{if the take-off time of flight } i \text{ is delayed by } r \text{ slots,} \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{D}, r \in \mathcal{R}_i \\ x_{ihr} &= \begin{cases} 1, & \text{if } i \text{ crosses via holding point } h \text{ and is delayed by } r \text{ slots,} \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{A}, h \in \mathcal{H}, r \in \mathcal{R}_i^a \\ y_{ir} &= \begin{cases} 1, & \text{if the pushback time of flight } i \text{ is delayed by } r \text{ slots,} \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{D}, r \in \mathcal{R}_i^p \\ \delta_{ij} &= \begin{cases} 1, & \text{if } i \text{ uses runway before } j, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{F}, j \in \mathcal{F}, i \neq j \end{aligned}$$

$$\begin{aligned}
\alpha_{is} &= \begin{cases} 1, & \text{if } i \text{ arrives at the runway threshold after time } s, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{D}, s \in \mathcal{T} \\
\beta_{is} &= \begin{cases} 1, & \text{if } i \text{ takes off before time } s, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{D}, s \in \mathcal{T} \\
\beta_{ihs} &= \begin{cases} 1, & \text{if } i \text{ crosses via holding point } h \text{ before time } s, \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{A}, h \in \mathcal{H}, s \in \mathcal{T}
\end{aligned}$$

The Objective Function. Our objective is expressed by:

$$\text{Min} \sum_{i \in \mathcal{D}} \sum_{r \in \mathcal{R}_i} r x_{ir} + \sum_{i \in \mathcal{A}} \sum_{h \in \mathcal{H}} \sum_{r \in \mathcal{R}_i^a} r x_{ihr}.$$

The Constraints. In order to introduce the constraints, we define the following sets to represent some infeasible assignments pairs of delays (and crossing point for arrivals). Given two distinct flights $i, j \in \mathcal{F}$, we introduce:

$$\mathcal{C}_{ij} = \{(r, q) \in \mathcal{R}_i \times \mathcal{R}_j | T_{jq} - T_{ir} < s_{ij} \text{ and } T_{ir} - T_{jr} < s_{ji}\} \text{ if } (i, j) \in \mathcal{D} \times \mathcal{D}$$

the set of delay assignments which violates the separation requirements between two departure flights;

$$\mathcal{C}_{ijh} = \begin{cases} \{(r, q) \in \mathcal{R}_i \times \mathcal{R}_j^a | T_{jq} - T_{ir} < s_{ij} \text{ and } T_{ir} - T_{jhq} < s_{ji}\} & \text{if } (i, j) \in \mathcal{D} \times \mathcal{A} \\ \{(r, q) \in \mathcal{R}_i^a \times \mathcal{R}_j^a | (T_{ih}^0 - T_{jh}^0)(T_{ihr} - T_{jhq}) < 0 \text{ or } (T_{jhq} - T_{ihr} < s_{ij} \text{ and } T_{ihr} - T_{jhq} < s_{ji})\} & \text{if } (i, j) \in \mathcal{A} \times \mathcal{A} \end{cases}$$

the set of delay assignments which violates the separation requirements between departure and arrival or between two arrivals using the same holding point. The first condition in the set definition expresses the violation of the FCFS order for arrival holding point. Notice that it is not possible to express the FCFS order for departure queue directly in the set definition. It is ensured by Constraints (38) and (39).

$$\sum_{r \in \mathcal{R}_i} x_{ir} = 1, \quad \forall i \in \mathcal{D}, \quad (30)$$

$$\sum_{h \in \mathcal{H}} \sum_{r \in \mathcal{R}_i^a} x_{ihr} = 1, \quad \forall i \in \mathcal{A}, \quad (31)$$

$$\sum_{r \in \mathcal{R}_i} r x_{ir} - \sum_{r \in \mathcal{R}_i^p} r y_{ir} \geq 0, \quad \forall i \in \mathcal{D}, \quad (32)$$

$$\sum_{r \in \mathcal{R}_i} r x_{ir} - \sum_{r \in \mathcal{R}_i^p} r y_{ir} \leq R_i^w, \quad \forall i \in \mathcal{D}, \quad (33)$$

$$x_{ir} + x_{jq} \leq 1, \quad \forall (r, q) \in \mathcal{C}_{ij}, \forall i, j \in \mathcal{D}, i < j, \quad (34)$$

$$x_{ir} + x_{jhq} \leq 1, \quad \forall (r, q) \in \mathcal{C}_{ijh}, \forall h \in \mathcal{H}, \forall i \in \mathcal{D}, \forall j \in \mathcal{A}, \quad (35)$$

$$x_{ihr} + x_{jhq} \leq 1, \quad \forall (r, q) \in \mathcal{C}_{ijh}, \forall h \in \mathcal{H}, \forall i, j \in \mathcal{A}, i < j, \quad (36)$$

$$\delta_{ij} + \delta_{ji} = 1, \quad \forall i, j \in \mathcal{D}, i < j, \quad (37)$$

$$\sum_{r \in \mathcal{R}_j} T_{jr} x_{jr} - \sum_{r \in \mathcal{R}_i} T_{ir} x_{ir} \geq -M \delta_{ij}, \quad \forall i, j \in \mathcal{D}, i \neq j, \quad (38)$$

$$\sum_{r \in \mathcal{R}_j^p} T_{jr} y_{jr} - \sum_{r \in \mathcal{R}_i^p} T_{ir} y_{ir} \geq -M \delta_{ij}, \quad \forall i, j \in \mathcal{D}, i \neq j, \quad (39)$$

$$\alpha_{is} \leq \alpha_{i(s+1)}, \quad \forall i \in \mathcal{D}, \forall s \in \mathcal{T} \setminus \{T\}, \quad (40)$$

$$\beta_{is} \geq \beta_{i(s+1)}, \quad \forall i \in \mathcal{D}, \forall s \in \mathcal{T} \setminus \{T\}, \quad (41)$$

$$\beta_{ihs} \geq \beta_{ih(s+1)}, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \forall s \in \mathcal{T} \setminus \{T\}, \quad (42)$$

$$\sum_{r \in \mathcal{R}_i} T_{ir} x_{ir} - \sum_{s \in \mathcal{T}} \beta_{is} = 0, \quad \forall i \in \mathcal{D}, \quad (43)$$

$$\sum_{r \in \mathcal{R}_i^p} T_{ir} y_{ir} - \sum_{s \in \mathcal{T}} (1 - \alpha_{is}) = 0, \quad \forall i \in \mathcal{D}, \quad (44)$$

$$\sum_{r \in \mathcal{R}_i^a} T_{ih} x_{ih} - \sum_{s \in \mathcal{T}} \beta_{ihs} = 0, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \quad (45)$$

$$\sum_{i \in \mathcal{D}} (\beta_{is} + \alpha_{is} - 1) \leq C_d, \quad \forall s \in \mathcal{T}, \quad (46)$$

$$\sum_{i \in \mathcal{A}} \beta_{ihs} \leq C_a, \quad \forall s \in \mathcal{T}, \forall h \in \mathcal{H}, \quad (47)$$

$$x_{ir} \in \{0, 1\}, \quad \forall i \in \mathcal{D}, \forall r \in \mathcal{R}_i, \quad (48)$$

$$x_{ih} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_i^a \quad (49)$$

$$y_{ir} \in \{0, 1\}, \quad \forall i \in \mathcal{D}, \forall r \in \mathcal{R}_i^p, \quad (50)$$

$$\delta_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{D}, i \neq j, \quad (51)$$

$$\alpha_{is} \in \{0, 1\}, \quad \forall i \in \mathcal{D}, \forall s \in \mathcal{T}, \quad (52)$$

$$\beta_{is} \in \{0, 1\}, \quad \forall i \in \mathcal{D}, \forall s \in \mathcal{T}, \quad (53)$$

$$\beta_{ihs} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \forall s \in \mathcal{T}, \quad (54)$$

Constraints (30) and (31) require exactly one time period of runway usage for departure and arrival flights. Constraints (32) guarantee that for departures, the arrival time at the runway threshold is always smaller than the take-off time. Constraints (33) state the maximum holding time for departures. Constraints (34), Constraints (35), and Constraints (36) ensure the separation criteria between two departures, arrival and departure, and two arrival crossings respectively. Constraints (37) guarantee that given any two aircraft, one leads the other. Constraints (38) and Constraints (39) guarantee the same order waiting at the runway threshold and for take-off for departures. Constraints (40–42) are the same as Constraints (11–13) in Model A. Constraints (43) and Constraints (44) link the binary variables for departures. Constraints (45) is similar to Constraints (43), that link the binary variables for arrivals. Constraints (46–47) are the same as Constraints (18–19) in Model A.

One disadvantage of this discrete time formulation is a relatively large number of variables (in particular, Constraints (34–36)). In order to reduce the number of variables, we reformulate Constraints (34) and Constraints (35) to Constraints (55), and Constraints (36) to Constraints (56),

$$x_{ir} + \sum_{j \in \mathcal{D}, j \neq i} \sum_{q | (r,q) \in \mathcal{C}_{ij}} x_{jq} + \sum_{h \in \mathcal{H}} \sum_{j \in \mathcal{A}} \sum_{q | (r,q) \in \mathcal{C}_{ijh}} x_{jhq} \leq 1 + M(1 - x_{ir}), \quad \forall r \in \mathcal{R}_i, \forall i \in \mathcal{D}, \quad (55)$$

$$x_{ihr} + \sum_{j \in \mathcal{A}, j \neq i} \sum_{q | (r,q) \in \mathcal{C}_{ijh}} x_{jhq} \leq 1 + M(1 - x_{ihr}), \quad \forall r \in \mathcal{R}_i^a, \forall h \in \mathcal{H}, \forall i \in \mathcal{A}. \quad (56)$$

There are two cases to consider: if $x_{ir} = 1$, then the take-off time of flight i is delayed by r slots, Constraints (55) become $\sum_{j \in \mathcal{D}, j \neq i} \sum_{q | (r,q) \in \mathcal{C}_{ij}} x_{jq} + \sum_{h \in \mathcal{H}} \sum_{j \in \mathcal{A}} \sum_{q | (r,q) \in \mathcal{C}_{ijh}} x_{jhq} \leq 0$, ensuring that i is separated with respect to all the other flights. If $x_{ir} = 0$, then Constraints (55) become $\sum_{j \in \mathcal{D}, j \neq i} \sum_{q | (r,q) \in \mathcal{C}_{ij}} x_{jq} + \sum_{h \in \mathcal{H}} \sum_{j \in \mathcal{A}} \sum_{q | (r,q) \in \mathcal{C}_{ijh}} x_{jhq} \leq 1 + M$, which inactivates the constraints. The same reasoning applies to Constraints (56).

The decision variables of Model A can be linked to the decision variables of Model B using the following equations (57–60):

$$w_i = \sum_{r \in \mathcal{R}_i} i x_{ir}, \quad \forall i \in \mathcal{D}, \quad (57)$$

$$d_i = \sum_{r \in \mathcal{R}_i^p} i y_{ir}, \quad \forall i \in \mathcal{D}, \quad (58)$$

$$w_i = \sum_{h \in \mathcal{H}} \sum_{r \in \mathcal{R}_i^a} i x_{ihr}, \quad \forall i \in \mathcal{A}, \quad (59)$$

$$z_{ih} = \sum_{r \in \mathcal{R}_i^a} x_{ihr}, \quad \forall i \in \mathcal{A}, \forall h \in \mathcal{H}, \quad (60)$$

It is possible to remove all the “big-M” constraints in Model B by introducing the decision variables x_{irq} instead of x_{ir} and y_{ir} , where

$$x_{irq} = \begin{cases} 1, & \text{if the pushback time of flight } i \text{ is delayed by } r \text{ slots} \\ & \text{and flight } i \text{ holds } q \text{ slots at the runway threshold,} \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } i \in \mathcal{D}, r \in \mathcal{R}_i^p, q \in \mathcal{R}_i^w$$

However, experimental tests showed that the benefit is counterbalanced by the large increase of the number of variables and constraints.

After describing our two ILP formulations, in the next section we consider also the metaheuristic method to tackle the aircraft runway scheduling problem.

4. Simulated Annealing

In this section we present a simulated annealing algorithm to solve our aircraft runway scheduling problem.

Simulated Annealing (Kirkpatrick et al., 1983) is a meta-heuristic that simulates the annealing of a metal, in which the metal is heated up and slowly cooled down to move towards an optimal energy state. SA is applied to many optimization problems, in particular the job shop scheduling problem (Cruz-

Algorithm 1 Simulated Annealing

Require: Initial state S_0, T_0 ;
 $S_c \leftarrow S_0, T \leftarrow T_0, i \leftarrow 0$;
while $T > T_f$ **do**
 for $i = 0$ to N_T **do**
 Generate a neighborhood solution, S_N ;
 if $E(S_N) \leq E(S_C)$ **then**
 $S_C \leftarrow S_N$;
 else if $\text{random}(0,1) < \exp \frac{-(E(S_N)-E(S_C))}{T}$ **then**
 $S_C \leftarrow S_N$;
 end if
 end for
 $T \leftarrow \beta T$;
end while
return S_c

Chávez et al., 2017). In SA, the objective function to be minimized is analogous to the energy of the physical problem. Algorithm 1 illustrates the SA procedure, where the parameters are

- S_0 : the initial solution;
- S_C : the current solution;
- S_N : the neighborhood solution generated from S_C ;
- T_0 : the initial temperature;
- T : the current temperature;
- T_f : the final temperature;
- E : the SA fitness function;
- N_T : the total number of transitions in each temperature step;
- β : the geometrical temperature reduction coefficient.

A global parameter T is used to simulate the cooling process. A current solution may be replaced by a random “neighborhood” solution with a probability. We start the cooling process from a high initial temperature T_0 (which can be determined by a heating process or defined by user), the current solution changes almost randomly at a high temperature, thus the algorithm is able to escape from local minima. The decrease of temperature may follow different laws, such as linear law, geometric law, etc. At each temperature step, a number of transitions are executed. The probability to accept a degrading solution become smaller and smaller when T decreases. Therefore, at the final stages of the annealing process, the system will converge to a near-global or global optimum.

In order to apply SA to our problem, the neighborhood generation is critical and needs to be defined precisely. Our neighborhood choice is similar to a Traveling Salesman Problem with a constrained search space. The algorithm starts with an initial sequence sorted by the earliest runway usage time. First, we randomly choose one flight, and we search for a list of sequence positions that can be used by the current chosen flight without exceeding the maximum delay. Then, we choose one flight in the list and apply three exchange strategies with the current one: *swap*, *inversion*, and *circular shift*. As shown in Fig. 3,

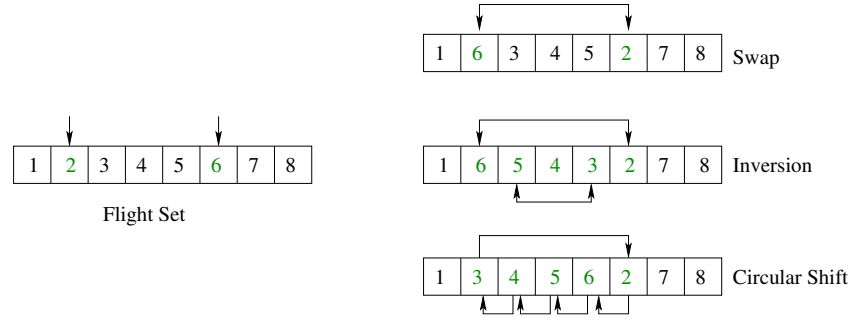


Fig. 3: Illustration of three neighborhood generation methods: swap, i.e., exchanging the positions of two aircraft; inversion, i.e., inverting the order of aircraft between two positions; circular shift, i.e., moving the first aircraft to the final position, while shifting all other aircraft to the previous position.

swap move exchanges the positions of two aircraft, inversion move inverts the order of aircraft between two positions, and circular shift move changes the first aircraft to the final position, while shifting all other aircraft to the previous position. Note that it is necessary to relax the maximum pushback delay and maximum holding time constraints in order to allow SA to search more freely in the state space. Nevertheless, a penalty coefficient is added if one aircraft cannot meet the maximum delay constraints. At the end of the procedure, SA can always find a solution satisfying the maximum delay requirement. A comparison test with one hour traffic is made to find the most efficient neighborhood generation method. As indicated in Fig. 4, the strategy combining circular shift move and swap move can achieve less delay than other strategies. Thus, in the remaining part of the paper, we choose circular shift move and swap move as our neighborhood generation method.

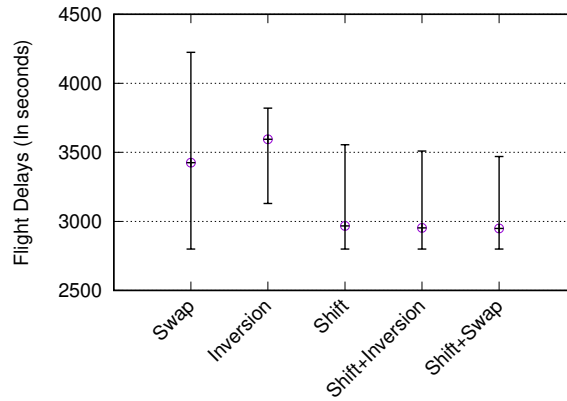


Fig. 4: Comparison of different neighborhood methods on solution qualities. We run the proposed SA algorithm 10 times with the same input parameters for each method. The error bar represents the minimum delay, the average delay, and the maximum delay respectively.

After the previous step, we obtained a new sequence, and we now have to update the flight schedules: First, we set the earliest runway usage time of the first flight in the sequence as its actual runway usage time. Then, for each aircraft, its runway usage time is set with respect to its earliest available time and the minimum separation with its precedent aircraft. Moreover, the best holding point for arrivals is determined. More precisely, we calculate for each holding point the required waiting time for the flight and choose the holding point with the shortest waiting time. Table 3 lists some SA parameters related to our problem.

Table 3: Empirically-set parameter values of SA

Parameter	Value
Geometrical temperature reduction coefficient, δ	0.95
Number of transitions at each temperature step, N_T	100
Initial temperature, T_0	0.1
Final temperature, T_f	$0.0001 * T_0$

5. Computational results

In this section we present the results of some computational experiments. We test our methodology on both real data cases at Paris CDG Airport and randomly generated data. Numerical results with different settings of (user-defined) algorithm parameters are presented and discussed. The overall process is run on a 2.50 GHz core i7 CPU, PC under Linux operating system, implemented in the Java programming language. The ILP models were solved with GUROBI solver (Optimization, 2014) with the default parameters settings. Table 4 lists some parameters related to our mathematical formulation. The actual pushback time from historical data is considered as the EOBT in this paper. In our test cases, all the flights are without CFMU slots due to data unavailability.

In the current system, controllers schedule aircraft take-offs in a FCFS order with regard to their earliest arrival time at runway threshold. In this study, we set a FCFS sequence as the baseline based on the earliest arrival time at runway threshold. We assume that all the arrival flights use the closest holding point after they land, i.e., S3. There is no limit on the maximum holding number and the maximum waiting time at the runway threshold. Moreover, a landing aircraft is assumed to cross the departure runway without any holding time. In a real-world case, arrivals do have priority on departures, as departures can be held on the ground. The arrival holding area is limited in terms of capacity and should be vacated as early as possible.

To illustrate how we calculate the FCFS sequence and compare with the optimized sequence, we give an example by generating a random instance consisting of 15 aircraft (5 arrivals and 10 departures). As shown in Table 5, given the initial landing time or initial pushback time, we calculate the earliest arrival time at runway threshold, t_i^T . For departures, $t_i^T = P_i + X_i$; For arrivals, $t_i^T = L_i + O_i + T_{e,h}$. We

Table 4: Chosen parameter values specifying the optimization problem.

Parameter	Value
Departure capacity at runway threshold, C_d	5
Arrival capacity at holding points, C_a	2
Time slot, ρ	5 seconds
Maximum pushback delay (in number of slots), R_i^p	120
Maximum holding time (in number of slots) for departures, R_i^w	120
Maximum holding time (in number of slots) for arrivals, R_i^a	36

Table 5: Example of FCFS sequence and optimized sequence with a random instance of 5 arrivals and 10 departures. H refers to Heavy, M refers to Medium, C refers to Crossing. All the times are in seconds.

Data	D01	D02	D03	D04	D05	D06	D07	D08	D09	D10	A01	A02	A03	A04	A05
Category	H	M	M	H	M	M	M	M	M	M	C	C	C	C	C
P_i	0	10	20	30	40	50	60	70	80	90	–	–	–	–	–
X_i	300	300	300	300	300	300	300	300	300	300	–	–	–	–	–
L_i	–	–	–	–	–	–	–	–	–	–	0	100	200	300	400
O_i	–	–	–	–	–	–	–	–	–	–	60	60	60	60	60
e	–	–	–	–	–	–	–	–	–	–	V4	V4	V4	V4	V4
FCFS	A01	A02	D01	A03	A04	D02	A05	D03	D04	D05	D06	D07	D08	D09	D10
t_i^T	175	275	300	375	475	310	575	320	330	340	350	360	370	380	390
ρw_i	0	0	15	0	0	205	0	295	345	455	505	555	605	655	705
t_i	175	275	315	375	475	515	575	615	675	795	855	915	975	1035	1095
h	S3	S3	–	S3	S3	–	S3	–	–	–	–	–	–	–	–
$T_{e,h}$	115	115	–	115	115	–	115	–	–	–	–	–	–	–	–
Total delay	4340														
Optimized	A01	A02	D02	D08	D10	D01	A03	A04	A05	D09	D05	D06	D07	D03	D04
ρd_i	–	–	0	0	30	190	–	–	–	230	330	380	430	535	580
$P_i + \rho d_i$	–	–	10	70	120	190	–	–	–	310	370	430	490	555	610
t_i^T	175	275	310	370	420	490	419	519	575	610	670	730	790	855	910
ρw_i	0	0	5	5	15	5	140	50	0	5	5	5	5	0	5
t_i	175	275	315	375	435	495	559	569	575	615	675	735	795	855	915
h	S3	S3	–	–	–	–	S1	S1	S3	–	–	–	–	–	–
$T_{e,h}$	115	115	–	–	–	–	158	158	115	–	–	–	–	–	–
Total delay	2950														

assume that all the arrival flights use the closest holding point after they land, i.e., S3. Then, we build a FCFS sequence by sorting t_i^T and we obtain the runway usage time t_i with regard to the separation

requirements. After that, the optimized sequence is presented in Table 5. We observed a decrease of total delay from 4340 s in the baseline case to 2950 s in the optimized case.

Table 6: Comparison of heuristics for solving one-hour real traffic on July 11th, 2017. Best results are in bold.

Time Window	Number of flights			Gap (%)				Computational time (s)			
	All	Dep.	Arr.	FCFS	Model A	Model B	SA	Model B Optimum	Model A	Model B	SA
8:00–9:00	54	31	23	69.59	–	–	2.35	113.13	10.00	10.00	3.35
9:00–10:00	49	23	26	84.43	3.92	–	6.34	43.98	10.00	10.00	3.60
10:00–11:00	34	16	18	66.16	0.00	19.44	0.00	12.31	2.02	10.00	1.57
11:00–12:00	55	30	25	82.64	12.08	–	2.15	58.35	10.00	10.00	3.82
12:00–13:00	40	22	18	84.04	0.00	–	4.62	20.15	4.37	10.00	1.94
13:00–14:00	25	20	5	33.31	4.89	–	4.37	22.47	10.00	10.00	0.41
14:00–15:00	46	16	30	79.18	7.32	–	5.04	27.51	10.00	10.00	3.31
15:00–16:00	41	18	23	67.31	0.00	18.68	0.00	13.51	2.22	10.00	2.44
16:00–17:00	39	24	15	68.82	0.00	–	7.81	26.82	10.00	10.00	1.52
17:00–18:00	45	19	26	81.02	10.84	–	2.06	29.34	10.00	10.00	2.94
18:00–19:00	47	23	24	43.12	10.25	45.27	0.12	64.87	10.00	10.00	3.00
19:00–20:00	54	32	22	76.23	28.87	–	5.55	156.80	10.00	10.00	3.21

We chose a heavy traffic summer day, July 11th, 2017 as our test case. An optimization time window of one hour is applied from 8:00 to 20:00; the rest of the periods involve only a few flights. As illustrated in Table 6, real traffic data provide various mixes of arrivals and departures: three fully loaded periods of 8:00-9:00, 11:00-12:00, and 19:00-20:00 with more than 30 departures per hour per runway, and 14:00-15:00 with 30 arrivals per hour per runway are observed.

Table 6 shows the comparison of optimization results for four cases: FCFS, Model A, Model B, and SA. We list the gap, i.e., the percent increase in total delay over the optimal solution. The computation time in seconds is listed in the last four columns. For SA, the gap and the computation time are averaged over 10 times random runs. In order to be fair and practical in the real world application, we set the computational time limit to be 10 seconds for all the algorithms. As shown in Table 6, first we can see a significant reduction of delays from Model A and SA compared to FCFS sequence. Model B cannot find a feasible solution in most of cases within the time limit. Model A can find a near-optimal solution. Nevertheless, it is difficult to prove the optimality due to the poor lower bound when the number of aircraft is large and the traffic scenario is dense. Therefore, we still use Model B to find the optimal solution with a reasonable time from 12 seconds up to 156 seconds, as shown in the first column of computational times in Table 6. In contrast, SA can find good quality solutions with most of the gaps less than 10 % within 5 seconds.

In order to test the performance of algorithms facing large instances, we set our time window to be two hours and the algorithm running time limit to be 20 seconds. The number of flights ranges from 65 to 103, which is challenging to find a good solution in a short period of time. As shown in Table 7, first we observe a large gap between FCFS solution and optimal solution. After optimization, SA can still find a

Table 7: Comparison of heuristics for solving two-hour real traffic on July 11th, 2017. Best results are in bold.

Time Window	Number of flights			Gap (%)				Computation time (s)			
	All	Dep.	Arr.	FCFS	Model A	Model B	SA	Model B Optimum	Model A	Model B	SA
8:00–10:00	103	54	49	84.15	–	–	3.92	378.90	20.00	20.00	15.68
10:00–12:00	89	46	43	81.51	11.35	–	0.46	99.67	20.00	20.00	10.35
12:00–14:00	65	42	23	68.74	12.26	–	2.65	83.40	20.00	20.00	3.91
14:00–16:00	87	34	53	77.13	6.97	–	4.21	53.78	20.00	20.00	11.22
16:00–18:00	84	43	41	77.56	22.13	–	3.42	83.79	20.00	20.00	8.81
18:00–20:00	101	55	46	81.88	39.31	–	5.12	383.93	20.00	20.00	13.14

near-optimal solution within the time limit compared to other models. In 18:00-20:00, the computational time is 384 seconds for Model B, even Model A has a gap of 39 % compared to optimality. This proves that SA is more suitable and more practical for dealing with large instances.

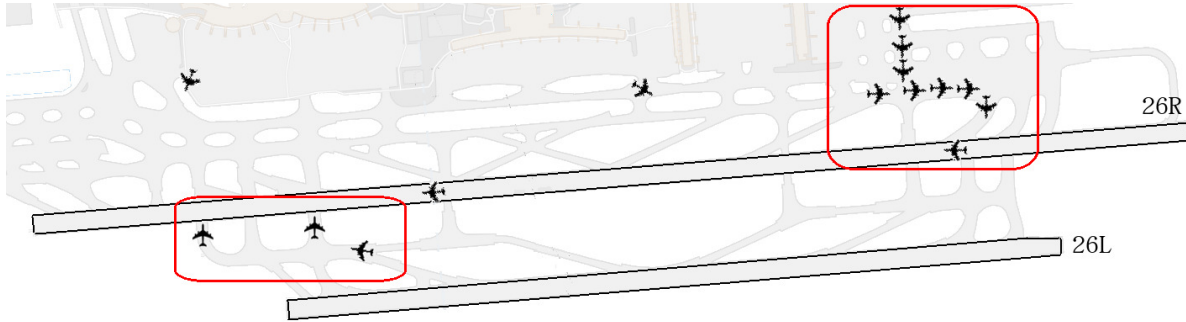
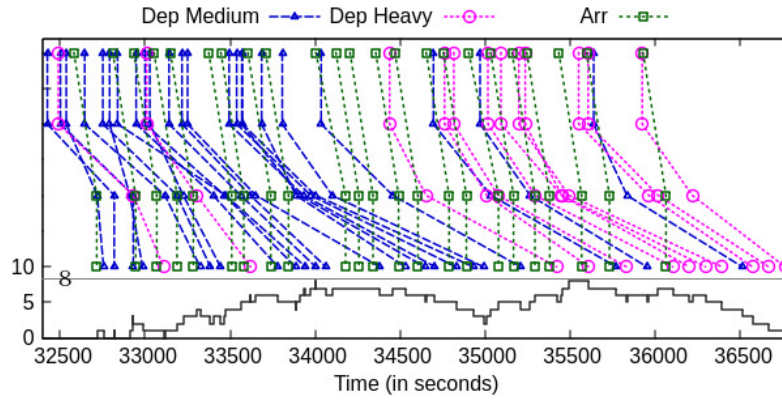


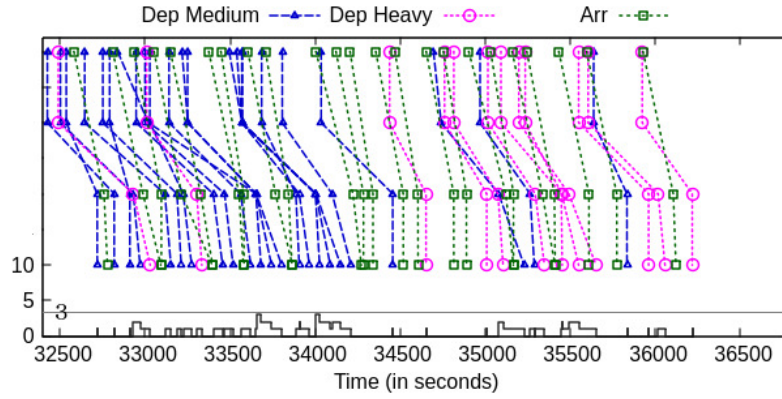
Fig. 5: Departure queues at 9:30 at runway 26R at CDG Airport on February 18th, 2016. Visualization of actual surface surveillance data.

We test another heavy traffic day on February 18th, 2016. A departure queue of 9 aircraft with 3 arrivals waiting at the holding point at 9:30 is illustrated in Fig. 5. As shown in Table 8, in 9:00-10:00, there are 58 aircraft with 35 departures and 23 arrivals.

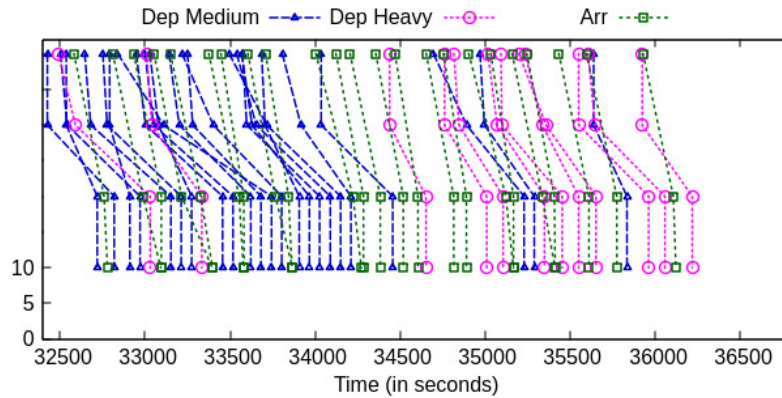
The optimization results are listed in Table 8. SA still keeps a similar performance compared to the first test. Model A also shows very good performance to find the optimal solution within 10 seconds, except in 9:00-10:00 with a gap of 41%. Note that 9:00-10:00 is the period in highest demand of the day. Below, we analyze in detail the optimization results of time window 9:00-10:00. Fig. 6 shows the sequencing results in 9:00-10:00 for three cases: the FCFS sequence, the optimal sequence with a preference to wait at the holding point, and the optimal sequence with a preference to wait at the gate. One can observe a large departure delay as well as a long departure queue for FCFS case illustrated at the bottom of the Figure 6a. While in Fig. 6b, much less departure delays are achieved with slight modification of arrival



(a) FCFS sequence. Each line represents one flight, the four points from top to bottom for departures indicate respectively initial pushback time, actual pushback time, arriving time at the runway threshold, and take-off time; the three points from top to bottom for arrivals indicate respectively landing time, arriving time at the runway threshold, and crossing time. The departure waiting queue is shown at the bottom of the figure.



(b) Optimal sequence with a preference to wait at the holding point.



(c) Optimal sequence with preference waiting at the gate.

Fig. 6: Comparison of the FCFS sequence, optimal sequence with a preference to wait at the holding point, and optimal sequence with preference waiting at the gate.

Table 8: Comparison of heuristics for solving one-hour real traffic on February 18th, 2016. Best results are in bold.

Time Window	Number of flights			Gap (%)				Computation time (s)			
	All	Dep.	Arr.	FCFS	Model A	Model B	SA	Model B Optimum	Model A	Model B	SA
6:00–7:00	41	19	22	61.23	11.79	–	0.64	86.73	10.00	10.00	1.51
7:00–8:00	48	13	35	71.16	0.00	53.67	3.81	14.57	5.35	10.00	2.31
8:00–9:00	34	17	17	57.49	0.00	0.00	0.00	8.20	1.25	8.20	1.05
9:00–10:00	58	35	23	85.31	41.09	–	6.55	353.74	10.00	10.00	3.28
10:00–11:00	48	21	27	79.81	8.46	–	1.81	106.17	10.00	10.00	2.40
11:00–12:00	42	20	22	26.98	1.57	–	0.00	51.01	10.00	10.00	1.68
12:00–13:00	42	30	12	33.09	7.31	–	8.48	124.70	10.00	10.00	1.35
13:00–14:00	37	14	23	89.82	0.00	0.00	1.96	6.22	1.36	6.22	1.33
14:00–15:00	41	23	18	47.05	0.00	69.09	2.54	24.38	6.80	10.00	1.40
15:00–16:00	40	20	20	45.77	0.00	–	0.99	16.02	5.39	10.00	1.47
16:00–17:00	38	10	28	58.86	0.00	4.21	0.00	11.51	4.76	10.00	1.49
17:00–18:00	46	25	21	84.14	9.09	–	0.00	25.20	10.00	10.00	1.99
18:00–19:00	45	19	26	69.45	0.47	–	0.61	80.24	10.00	10.00	2.02
19:00–20:00	46	26	20	73.77	5.26	–	2.17	34.19	10.00	10.00	1.89
20:00–21:00	41	21	20	74.54	0.00	–	2.36	28.71	10.00	10.00	1.54

holding time. The departure queue is reduced to 3 aircraft. In Fig. 6c, with a preference of holding at gate, aircraft taxi to the runway threshold and take off smoothly.

Next, we generate some random data set to check if our methods can be suitable for dense traffic scenarios. The mix of arrivals and departures, and the fleet mix of Medium and Heavy aircraft are two important factors in departure scheduling. Thus, we generate 15 random high traffic demand scenarios in a time window of 30 minutes. The number of arrivals/departures are set to be 10/20, 15/15, and 15/20. A mix of Heavy aircraft from 0% to 40% is set. Results are shown in Table 9: the solution quality decreases when the number of departures increases; SA keeps the shortest computational time; Model A is more suitable for cases with low departure rate.

In conclusion, Model B is suitable for finding an optimal solution while Model A is suitable for finding a feasible solution within a short computation time. In fact, by imposing a time limit, Model A solves most instances and sometimes reaches optimality. However, when the runway is in high demand, Model A is not able to find a good quality solution. The SA is suitable for finding a near-optimal solution in a short computational time.

6. Conclusions

The runway system is a major source of delay in the departure process. It is critical to achieve efficient scheduling of aircraft taking into account specific operational constraints. In this paper, we have developed two ILP models and one metaheuristic algorithm for departure runway scheduling incorporating

Table 9: Comparison of heuristics for solving random data with different fleet mix. A time window of 30 minutes is considered, “10–20–0–100” means 10 arrivals and 20 departures with a fleet mix of 0% Heavy and 100% Medium. Best results are in bold.

Arr-Dep-H%-M%	Gap (%)				Computation time (s)			
	FCFS	Model A	Model B	SA	Model B Optimum	Model A	Model B	SA
10–20–0–100	68.36	15.14	–	8.01	97.07	10.00	10.00	0.58
10–20–10–90	74.54	27.99	–	5.99	80.35	10.00	10.00	0.57
10–20–20–80	81.77	0.00	–	1.41	31.93	10.00	10.00	0.56
10–20–30–70	64.55	28.23	–	8.80	601.98	10.00	10.00	0.59
10–20–40–60	85.78	3.57	–	3.57	36.80	10.00	10.00	0.57
15–15–0–100	59.00	0.00	–	1.51	25.68	10.00	10.00	0.63
15–15–10–90	91.28	0.00	–	11.45	13.03	4.06	10.00	0.70
15–15–20–80	59.55	19.15	–	4.78	146.53	10.00	10.00	0.67
15–15–30–70	93.73	0.00	–	7.53	15.40	8.59	10.00	0.73
15–15–40–60	89.47	0.00	–	5.99	13.60	10.00	10.00	0.74
15–20–0–100	70.74	40.16	–	15.54	52.63	10.00	10.00	0.96
15–20–10–90	36.28	15.40	–	6.21	86.23	10.00	10.00	1.01
15–20–20–80	25.94	5.71	–	0.32	154.06	10.00	10.00	1.02
15–20–30–70	32.39	0.64	–	1.77	48.41	10.00	10.00	1.11
15–20–40–60	54.74	22.68	–	9.03	2194.23	10.00	10.00	1.05

arrival crossings. Specific constraints such as wake turbulence separations, flight time window restrictions, and holding queue capacity at runway threshold are explicitly considered. Different comparison tests on real and random data were launched and demonstrate that: the time slot based formulation can find an optimal or near-optimal solution quickly, but it is hard to prove optimality facing high demand periods; the delay-indexed formulation can prove optimality in a reasonable time; SA is a good candidate for having a near-optimal solution to reduce flight delays, and the run times are sufficiently small for real-time application. The runway sequencing algorithm can be implemented as a decision support tool for controllers in actual operations. However, the benefits of the runway sequencing algorithm may be limited by the uncertainty arising from real operations. For real-time application, flight information needs to be updated frequently and accurately as time passes. For example, the taxi time to the runway threshold can be varied with regard to the current aircraft position at the airport after pushback. A proper prediction of the travel time on the surface can help the tactical runway sequencing algorithms better organize the sequence and decrease flight delays. Future extensions of the model could develop a robust algorithm facing uncertain pushback time and taxi time changes.

Acknowledgments

This work has been partially supported by CAUC, by China Scholarship Council (CSC) and by the Natural Science Foundation of Tianjin grant 17JCYBJC43100. We would like to thank Serge Roux

and Andrea Hakala for helpful discussions and feedback on the manuscript. We would like to thank SNA-RP/CDG-LB for providing the traffic data. The authors thank the anonymous reviewers for their insightful comments.

References

- Anagnostakis, I., 2004. A multi-objective, decomposition-based algorithm design methodology and its application to runway operations planning. Ph.D. thesis, Massachusetts Institute of Technology.
- Anagnostakis, I., Clarke, J.P., 2003. Runway operations planning: a two-stage solution methodology. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, IEEE.
- Anagnostakis, I., Idris, H.R., Clarke, J.P., Feron, E., Hansman, R.J., Odoni, A.R., Hall, W.D., 2000. A conceptual design of a departure planner decision aid.
- Atkin, J.A., Burke, E.K., Greenwood, J.S., Reeson, D., 2007. Hybrid metaheuristics to aid runway scheduling at london heathrow airport. *Transportation Science* 41, 1, 90–106.
- Atkin, J.A., Burke, E.K., Greenwood, J.S., Reeson, D., 2008a. A metaheuristic approach to aircraft departure scheduling at london heathrow airport. In *Computer-aided Systems in Public Transport*. Springer, pp. 235–252.
- Atkin, J.A., Burke, E.K., Greenwood, J.S., Reeson, D., 2008b. On-line decision support for take-off runway scheduling with uncertain taxi times at london heathrow airport. *Journal of Scheduling* 11, 5, 323.
- Atkin, J.A., Burke, E.K., Greenwood, J.S., Reeson, D., 2009. An examination of take-off scheduling constraints at london heathrow airport. *Public Transport* 1, 3, 169.
- Bae, S.W., Park, J., Clarke, J.P., 2013. Modified mixed integer linear program for airport departure scheduling. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, p. 4885.
- Balakrishnan, H., Chandran, B., 2006. Scheduling aircraft landings under constrained position shifting. In *AIAA guidance, navigation, and control conference and exhibit*, p. 6320.
- Balakrishnan, H., Chandran, B., 2007. Efficient and equitable departure scheduling in real-time: new approaches to old problems. In *7th USA-Europe Air Traffic Management Research and Development Seminar*, pp. 02–05.
- Balakrishnan, H., Chandran, B.G., 2010. Algorithms for scheduling runway operations under constrained position shifting. *Operations Research* 58, 6, 1650–1665.
- Bennell, J.A., Mesgarpour, M., Potts, C.N., 2011. Airport runway scheduling. *4OR* 9, 2, 115.
- Böhme, D., 2005. Tactical departure management with the eurocontrol/dlr dman. In *6th USA/Europe Air Traffic Management Research and Development Seminar, Baltimore, MD*.
- Chaudhry, I.A., Khan, A.A., 2016. A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operational Research* 23, 3, 551–591.
- Cruz-Chávez, M.A., Martínez-Rangel, M.G., Cruz-Rosales, M.H., 2017. Accelerated simulated annealing algorithm applied to the flexible job shop scheduling problem. *International Transactions in Operational Research* 24, 5, 1119–1137.
- D’Ariano, A., D’Urgolo, P., Pacciarelli, D., Pranzo, M., 2010. Optimal sequencing of aircrafts take-off and landing at a busy airport. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, IEEE, pp. 1569–1574.
- Deau, R., Gotteland, J.B., Durand, N., 2009. Airport surface management and runways scheduling. In *ATM 2009, 8th USA/Europe Air Traffic Management Research and Development Seminar*.
- Du, J., Leung, J.Y.T., 1990. Minimizing total tardiness on one machine is np-hard. *Mathematics of operations research* 15, 3, 483–495.
- Furini, F., Kidd, M.P., Persiani, C.A., Toth, P., 2015. Improved rolling horizon approaches to the aircraft sequencing problem. *Journal of Scheduling* 18, 5, 435–447.
- Gupta, G., Malik, W., Jung, Y., 2009. A mixed integer linear program for airport departure scheduling. In *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, p. 6933.
- Gupta, G., Malik, W., Jung, Y., 2010. Incorporating active runway crossings in airport departure scheduling. In *AIAA Guidance, Navigation, and Control Conference*, p. 7695.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *science* 220, 4598, 671–680.

- Lieder, A., Stolletz, R., 2016. Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways. *Transportation research part E: logistics and transportation review* 88, 167–188.
- Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling-salesman problem. *Operations research* 21, 2, 498–516.
- Malik, W., Jung, Y.C., 2016. Exact and heuristic algorithms for runway scheduling. In *16th AIAA Aviation Technology, Integration, and Operations Conference*, p. 4072.
- Montoya, J., Rathinam, S., Wood, Z., 2014. Multiobjective departure runway scheduling using dynamic programming. *IEEE Transactions on Intelligent Transportation Systems* 15, 1, 399–413.
- Mori, R., 2017. Development of a pushback time assignment algorithm considering uncertainty. *Journal of Air Transportation* pp. 1–10.
- Optimization, G., 2014. Gurobi optimizer reference manual.
- Van Leeuwen, P., Hesselink, H., Rohling, J., 2002. Scheduling aircraft using constraint satisfaction. *Electronic notes in theoretical computer science* 76, 252–268.