



HAL
open science

Integrated optimization of terminal maneuvering area and airport at the macroscopic level

Ji Ma, Daniel Delahaye, Mohammed Sbihi, Paolo Maria Scala, Miguel
Antonio Mujica Mota

► **To cite this version:**

Ji Ma, Daniel Delahaye, Mohammed Sbihi, Paolo Maria Scala, Miguel Antonio Mujica Mota. Integrated optimization of terminal maneuvering area and airport at the macroscopic level. Transportation research. Part C, Emerging technologies, 2019, 98, pp.338-357. 10.1016/j.trc.2018.12.006 . hal-02053694

HAL Id: hal-02053694

<https://enac.hal.science/hal-02053694>

Submitted on 4 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.

Integrated optimization of terminal maneuvering area and airport at the macroscopic level

Ji Ma^{a,*}, Daniel Delahaye^a, Mohammed Sbihi^a, Paolo Scala^b, Miguel Antonio Mujica Mota^b

^a*ENAC, University de Toulouse, 7 Avenue Edouard Belin, Toulouse 31055, France*

^b*Aviation Academy, Amsterdam University of Applied Sciences, Amsterdam, The Netherlands*

Abstract

Airports and surrounding airspaces are limited in terms of capacity and represent the major bottlenecks of the air traffic management system. This paper addresses the problems of terminal airspace management and airport congestion management at the macroscopic level through the integrated control of arrivals and departures. Conflict detection and resolution methods are applied to a predefined terminal route structure. Different airside components are modeled using network abstraction. Speed, arrival and departure times, and runway assignment are managed by using an optimization method. An adapted simulated annealing heuristic combined with a time decomposition approach is proposed to solve the corresponding problem. Computational experiments performed on case studies of Paris Charles De-Gaulle airport show some potential improvements: First, when the airport capacity is decreased, until a certain threshold, the overload can be mitigated properly by adjusting the aircraft entry time in the Terminal Maneuvering Area and the pushback time. Second, landing and take-off runway assignments in peak hours with imbalanced runway throughputs can significantly reduce flight delays. A decrease of 37% arrival delays and 36% departure delays was reached compared to baseline case.

Keywords: Integrated Optimization, Terminal Maneuvering Area, Airport, Simulated Annealing

1. Introduction

With the steady growth of air traffic demand, the current air network is facing capacity problems, leading to delays and congestions. One of the most critical parts is the airport and its surrounding airspaces. Increasing use of

*Corresponding author

Email addresses: ji.ma@recherche.enac.fr (Ji Ma), daniel@recherche.enac.fr (Daniel Delahaye), mohammed.sbihi@enac.fr (Mohammed Sbihi), p.m.scala@hva.nl (Paolo Scala), m.mujica.mota@hva.nl (Miguel Antonio Mujica Mota)

saturated airfield capacity will adversely impact predictability and punctuality. European SESAR (Single European Sky ATM Research) program [1] and FAA's NextGen (Next Generation Air Transportation System) plan [2] aim to increase the network traffic throughput in order to accommodate the forecast demand with a sufficient margin. To achieve this goal, new technologies integrating existing optimization support systems in order to act as holistic decision-support tools for all airport partners are proposed, such as the Total Airport Management Concept (TAM) [3]. Efficient planning and optimization approaches of airport operations are critical to alleviate traffic congestion.

In previous work, segregated problems on runway sequencing and scheduling, and airport ground optimization have been studied extensively. Bennell *et al.* [4] gave a brief review of the techniques and tools for scheduling aircraft landings and take-offs. Atkin *et al.* [5] provided an overview of the research for ground movement and the integration of various airport operations.

The runway sequencing and scheduling problem aims to find the optimal schedule for aircraft in order to reduce delay and to maximize runway throughput, taking into account safety and operational constraints. Arrivals and departures are often considered as separate problems. Past efforts have also been made for mixed operations (simultaneous arrival and departure scheduling on a single runway). Beasley *et al.* [6] presented a mixed-integer, zero-one program to schedule aircraft landings. Atkin *et al.* [7] proposed a hybrid metaheuristic system to improve runway scheduling at the London Heathrow airport. Balakrishnan and Chandran [8] presented dynamic programming algorithms for the problem of mixed operations under constrained position shifting. An interesting rolling horizon approach to the aircraft sequencing problem for arrivals and departures was proposed by Furini *et al.* [9].

Runway sequencing is intimately linked to other airport ground operations. The problem becomes more complicated for finding the best schedules and routes with respect to taxiing separation, route choices, runway wake turbulence separation etc. Gotteland *et al.* [10, 11] presented a hybrid algorithm combining a genetic algorithm, and branch and bound to solve the ground movement and runway sequencing problem. Lee and Balakrishnan [12] introduced a mixed-integer linear programming model to optimize both taxiway and runway schedules. Jung *et al.* [13] presented two decoupled optimization algorithms to provide sequence and timing advisories for pushback and take-off. Ma *et al.* [14] proposed a global optimization approach to solve the surface operations problem and compared different control strategies (controlled pushback time, taxi reroutes and controlled holding time).

Recently, more research focuses on the integrated optimization of TMA (Terminal Maneuvering Area) and airport. Integrating terminal airspace management with existing route network is a more complicated, but more realistic problem than considering only the runway scheduling and sequencing problem in TMA. Khadilkar and Balakrishnan [15] modeled departure operations using a network abstraction, and combined with published arrival routes, used dynamic programming to solve the integrated control problem in order to get the optimal times of departures. Xue and Zelinski [16] modeled terminal airspace

by spatially and temporally segregating arrival and departure routes. Bosson *et al.* [17] extended previous research with surface operations to integrate taxiway and runway operations. Frankovich [18] proposed unified approaches on both strategic and tactical levels to optimize the traffic flowing through an airport.

Preliminary research on merging flows in TMA using a time decomposition approach [19] and reducing airport capacity overload [20] has been presented. This paper studies the integrated problem of terminal airspace management for arrivals and departures, and airport capacity management through the abstraction model of terminal, taxi network, and runway. A fast metaheuristic algorithm combined with a time decomposition approach is proposed. The case studies based on Paris CDG airport show some potential benefits: First, assuming a decrease of airport capacity based on the current level, until a certain threshold, the overload can be mitigated properly by adjusting the flight time decisions. Second, the benefits of landing and take-off runway assignments in peak hours are studied.

The remaining parts of this paper are organized as follows. Section 2 presents the mathematical model of the integrated terminal airspace management and airport congestion management problem. A metaheuristic method combined with a time decomposition approach aiming at minimizing the airspace conflicts, airport overload, and total flight delays is presented in Section 3. Computational experiments conducted with the proposed methodology are presented in Section 4. Conclusions and perspectives are discussed in Section 5.

2. Problem description and model

In the terminal airspace, aircraft from different entry points must be merged and sequenced into an orderly stream, follow the Standard Terminal Arrival Routes (STAR), then prepare to land on the runway. After slowing down the speed and exiting the runway, aircraft taxi towards the assigned gate. Then, after a certain turnaround duration for disembark, embark and other ground-holding operations, aircraft push back, taxi out, depart, and follow the designated Standard Instrument Departure (SID) routes.

Based on different levels of fidelity, the airport models are broadly described as microscopic or macroscopic. In microscopic levels, individual aircraft trajectories with detailed information about taxiway routing, gate occupancy are explicitly considered. However, the simulations can be computationally intensive. In macroscopic models, the airport components (terminals, taxi network) can be globally modeled as resources with specific capacities (as opposed to individual aircraft or taxiway links). This level of abstraction can help better understand the airport congestion situations and integrate into decision support tools.

Our first step is to consider the terminal and airport integration problem at the macroscopic level, in order to be sufficiently flexible to resolve airspace conflicts (which implicitly represents potential workload for air traffic controllers), to mitigate airport congestion, and to reduce delays. We focus on the pre-tactical off-line planning phase, i.e., we assume the planning time to be several

hours, or at least 30 min, prior to actual arrival/departure time. The integrated problem is considered in a moving time frame to reduce computational burden and to account for the frozen flights, which were already optimized in the previous time window and are traveling in the current time window. In future work, the model will be improved to on-line planning taking into account uncertainties. The uncertainty of aircraft arrival time will increase as time passes by, thus a more robust occupancy curve is built and evaluated.

2.1. Network model of TMA and airport surface

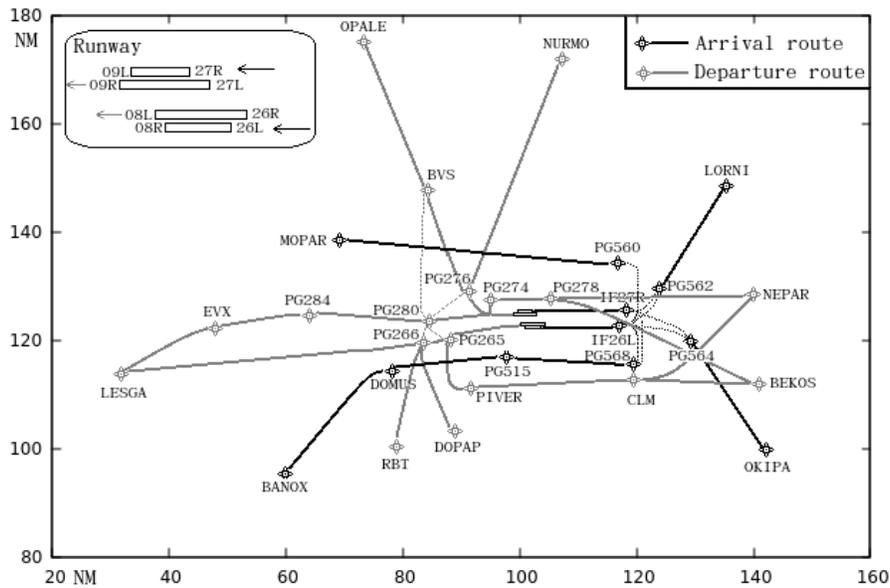


Figure 1: Terminal route network of arrivals and departures in CDG in west configuration

We model the TMA arrival and departure routes by a graph, $\mathcal{G}(\mathcal{X}, \mathcal{L})$, in which the aircraft are allowed to fly in the airspace, where \mathcal{X} is the node set and \mathcal{L} is the link set. Each route is defined by a sequence of nodes and links; the first link starts from an entering point (a so-called Initial Approach Fix (IAF) for arrivals and runway threshold for departures) and the last link ends at the exit point (runway threshold for arrivals and last SID waypoint for departures).

Fig. 1 displays an example model of a route network for the Paris Charles De-Gaulle (CDG) airport. CDG is one of the busiest passenger airports in Europe, it is composed of four parallel runways (two for landings and two for take-offs) and three terminals. The West configuration with runways 26L/26R and 27L/27R is illustrated in Fig. 1, arrival and departure procedures are respectively represented by black and gray colors. In the arrival procedure, four-corner routes fuse into one to each runway. Each of the starting nodes of these four routes is an IAF. The set of entering points here is $\mathcal{X}_e = \{ \text{MOPAR, LORNI,}$

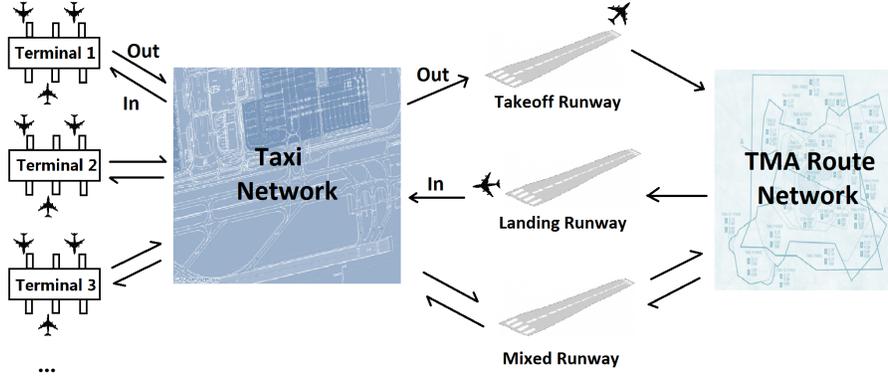


Figure 2: Network model of TMA and airport surface. Each component is considered as resource with a specific capacity.

OKIPA, BANOX}. For the departure procedure, one route starts at the runway threshold and ends with one of the SID waypoints in the set $\mathcal{X}_x = \{\text{OPALE, NURMO, NEPAR, BEKOS, DOPAP, RBT, LESGA}\}$. The set of routes is denoted as $\mathcal{R} = \{r_e | e \in \mathcal{X}_e \cup \mathcal{X}_x\}$. Each aircraft follows exactly one of these routes corresponding to its entering point and landing runway for arrivals, and exit point and take-off runway for departures.

According to real radar data and published routes, departure and arrival trajectories are separated in altitude. The arrival flows from the North-West (South-West) maintain their flight level at about 12,000 ft (13,000 ft) on the route section overlapping with departure flows between MOPAR and PG560 (DOMUS and PG515), and the departure flows to the North (South) pass below them. In the Eastern part, IF27R keeps a flight level of 5000 ft, IF26L keeps 4000 ft, so that the departures are able to fly above the arrivals.

Different airport components are considered using a network abstraction. Runways and terminals are modeled as resources with a specific capacity. We only take into account the overall capacity of a terminal without considering its individual gates. Taxiway is seen as a network with a threshold of total allowed number of taxi-in and taxi-out aircraft. The network model of TMA and airport surface is illustrated in Fig 2.

In the next section, we describe an integrated global optimization model of TMA and airport. We first give the flight input data. Then, decision variables are defined. Lastly, we clarify constraints and introduce the objective function.

2.2. Input data

Assume that we are given a set of flights (or aircraft), $\mathcal{F} = \mathcal{A} \cup \mathcal{AD} \cup \mathcal{D}$, where \mathcal{A} is a set of arrivals, flights that land at the airport and stay until the end of the day; \mathcal{AD} is a set of arrival-departures, flights that arrive at the airport and depart from it after a turnaround duration; \mathcal{D} is a set of departures, flights that are parked at the airport at the beginning of the day and depart later.

For each flight $f \in \mathcal{F}$, the following data is given: wake turbulence category for $f \in \mathcal{F}$, assigned terminal for $f \in \mathcal{F}$, entering waypoint at TMA for $f \in \mathcal{A} \cup \mathcal{AD}$, exit waypoint at TMA for $f \in \mathcal{D} \cup \mathcal{AD}$, taxi-in duration for $f \in \mathcal{A} \cup \mathcal{AD}$, taxi-out duration for $f \in \mathcal{D} \cup \mathcal{AD}$, initial landing runway number for $f \in \mathcal{A} \cup \mathcal{AD}$, initial departure runway number for $f \in \mathcal{D} \cup \mathcal{AD}$, initial off-block time for $f \in \mathcal{D}$, turnaround duration for $f \in \mathcal{AD}$ and initial exit time at the exit SID waypoint for $f \in \mathcal{D} \cup \mathcal{AD}$. Moreover, we know:

- T_f^0 : initial RTA (Required Time of Arrival) at the entering waypoint of TMA ($f \in \mathcal{A} \cup \mathcal{AD}$);
- V_f^0 : initial speed at the entering waypoint of TMA ($f \in \mathcal{A} \cup \mathcal{AD}$);
- P_f^0 : initial off-block time ($f \in \mathcal{D} \cup \mathcal{AD}$), it is the earliest time that an aircraft is ready to depart from its parking position.

Here are the assumptions and simplifications we make for our model:

- Flights are assumed to be able to park at any gates in their assigned terminal;
- We use an average taxi-in and taxi-out duration with regard to terminal and runway for each flight, due to the fact that we do not have information about the gate for the macroscopic model of the airport. Detailed study of airport taxi routings can be found in [14];
- Each aircraft has a constant deceleration or acceleration in the TMA. However, our model can tackle other types of trajectory (real radar data, BADA data) by discretizing the airspace into a space-time grid and detecting conflicts, as done in the work of Chaimatanan *et al.* [21]

2.3. Decision variables

The optimization model we are using has five types of decision variables. For arrivals, we have to attribute the entering time in the TMA, the entering speed in the TMA, and the landing runway:

1. Entering time in the TMA for $f \in \mathcal{A} \cup \mathcal{AD}$: First, we assume that we are given a maximum delay and a maximum advance, denoted respectively ΔT_{\max} and ΔT_{\min} , which define the range of possible entering times in the TMA. We therefore define, for each flight $f \in \mathcal{A} \cup \mathcal{AD}$, a time-slot decision variable $t_f \in \mathcal{T}_f$, where

$$\mathcal{T}_f = \{T_f^0 + j\Delta T \mid \Delta T_{\min}/\Delta T \leq j \leq \Delta T_{\max}/\Delta T, j \in \mathbb{Z}\},$$

where ΔT is a discretized time increment, whose value is to be set by the user. In order to shift an aircraft entering time in the TMA, we can either decrease or increase its speed in the en-route phase. In practice, the latter strategy burns more fuel, and may be far less attractive for the airlines. As a consequence, our time slot interval can be asymmetric, with $|\Delta T_{\max}| \geq |\Delta T_{\min}|$. In the following sections, the notation *delay* is used to indicate the time deviation of a flight.

2. Entering speed in the TMA for $f \in \mathcal{A} \cup \mathcal{AD}$: $v_f \in \mathcal{V}_f$, where

$$\mathcal{V}_f = \{V_f^{\min} + j\Delta_f^v \mid j \in \mathbb{Z}, |j| \leq (V_f^{\max} - V_f^{\min})/\Delta_f^v\},$$

with Δ_f^v is a (user-defined) speed increment, V_f^{\min} and V_f^{\max} are given as input data corresponding to the minimum and maximum allowable speeds for aircraft f .

3. Landing runway for $f \in \mathcal{A} \cup \mathcal{AD}$: r_f^l is the landing runway decision for arrivals. Runway assignment is used to balance the capacity when one runway gets overloaded while another is still able to accommodate more aircraft. Fig. 3 gives an example of how flight delay can be reduced by assigning the landing runways. In Case 1, with a First-Come-First-Served strategy, a total of 470 s is required for all five aircraft to land when all the traffic arrives on southern runway 26L. In Case 2, after optimizing the landing sequence for the same runway, a total of 258 s is required. In Case 3, the total landing time is reduced to 120 s with the possibility of runway assignment. Runway aircraft assignment enables to increase overall throughput with less delay comparing with the case where aircraft have no options to change their landing runway.

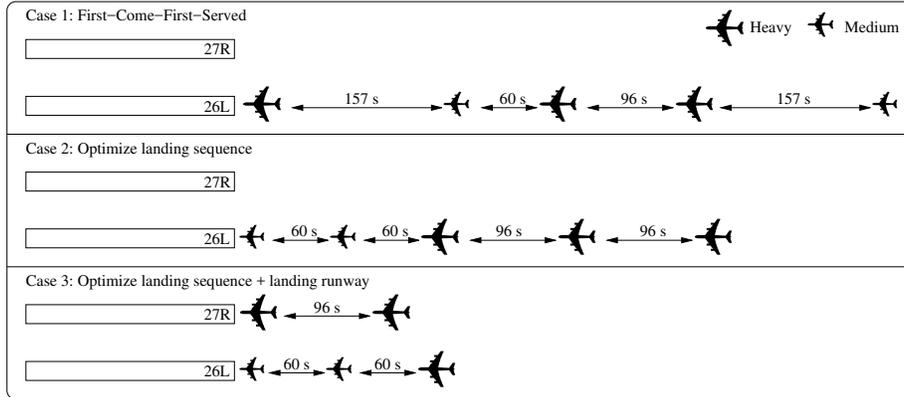


Figure 3: Three landing sequences comparison. In Case 1, First-Come-First-Served strategy is applied; In Case 2, the landing sequence is optimized with regard to wake turbulence separation requirements; In Case 3, by assigning a landing runway and optimize landing sequence, less delay is achieved.

For departures, we have to decide the departure runway and the pushback time:

4. Departure runway for $f \in \mathcal{D} \cup \mathcal{AD}$: r_f^d is the take-off runway decision variable for departures. Similarly, it's possible to yield flights to another take-off runway when the current assigned one is busy.
5. Pushback time for $f \in \mathcal{D} \cup \mathcal{AD}$: $p_f \in \mathcal{P}_f$, where

$$\mathcal{P}_f = \{P_f^0 + j\Delta T \mid 0 \leq j \leq \Delta T_{\max}^p/\Delta T, j \in \mathbb{N}\},$$

where ΔT_{\max}^p is the maximum pushback delay. In contrast to the entering time decision in the TMA for arrival flights, we can only delay a departure with regard to its earliest initial off-block time.

To summarize, our decision vector is $\mathbf{x} = (\mathbf{t}, \mathbf{v}, \mathbf{l}, \mathbf{d}, \mathbf{p})$, where \mathbf{t} is the vector for which the f^{th} component is the decision variable t_f , \mathbf{v} is the vector for which the f^{th} component is the decision variable v_f , \mathbf{l} is the vector for which the f^{th} component is the decision variable r_f^l , \mathbf{d} is the vector for which the f^{th} component is the decision variable r_f^d , and \mathbf{p} is the vector for which the f^{th} component is the decision variable p_f .

2.4. Constraints

We have two main constraints: wake turbulence separation, and single-runway separation for arrivals and departures. Before taking into account these separations, we first calculate the passage times at which the aircraft passes each resource (node, link, runway, taxi network, terminal) based on the decision vector \mathbf{x} . Let us denote respectively the passage time at the resource m , the entering time to the resource m , and the exit time from the resource m by $T_f^m(\mathbf{x})$, $T_{\text{In}}^{f,m}(\mathbf{x})$, and $T_{\text{Out}}^{f,m}(\mathbf{x})$.

2.4.1. Conflicts detection in the TMA

In this paper, we make the assumption that the arrival and departure routes are separated in altitude, which corresponds to real-world TMA operations. Therefore, we detect conflicts separately for arrivals and for departures. Considering the above-described TMA route network structure, the TMA separation violation may happen either in the link or in the node:

Table 1: Distance-based separation on approach and departure according to aircraft categories (in NM). For example, the minimum distance separation between an heavy aircraft followed by a medium aircraft is 5 NM.

Category		Trailing Aircraft		
		Heavy	Medium	Light
Leading Aircraft	Heavy	4	5	6
	Medium	3	3	5
	Light	3	3	3

- Link conflict: As shown in Fig. 4(a), for two consecutive flights f, g that are flying through a link $l = (u, v)$, the minimum separation between these two aircraft, s_{fg} , is obtained based on their respective wake turbulence category as shown in Table 1. Then, the actual separation distance of these aircraft at the entry time, $d_{fg}^u(\mathbf{x})$, and at the exit time of link l , $d_{fg}^v(\mathbf{x})$, are computed and compared with s_{fg} to detect an eventual link conflict.

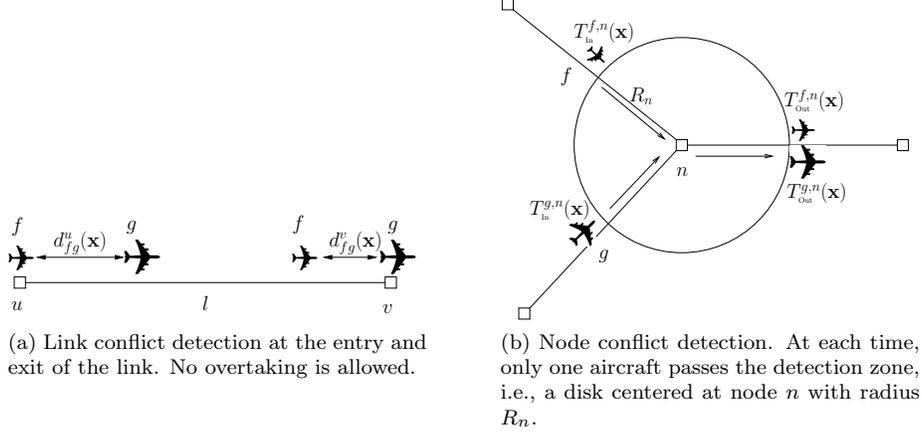


Figure 4: Airspace conflict detection illustration

Let us define, the *link conflict indicator*, $L_{fg}^l(\mathbf{x})$, for aircraft f and g at link l :

$$L_{fg}^l(\mathbf{x}) = \begin{cases} 1, & \text{if } T_f^u(\mathbf{x}) < T_g^u(\mathbf{x}) \text{ and } (d_{fg}^u(\mathbf{x}) < s_{fg} \text{ or } d_{fg}^v(\mathbf{x}) < s_{fg}) \\ & \text{or } T_f^v(\mathbf{x}) > T_g^v(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases}$$

where $T_f^u(\mathbf{x})$ is the passage time of flight f at the entry node u of link l , $T_f^v(\mathbf{x})$ is the passage time of flight f at the exit node v of link l .

- **Node conflict:** If no link conflict is detected, wake-turbulence separation can be guaranteed. However, at the intersection of two successive links, violation of the horizontal separation requirement between any two consecutive aircraft (3 NM in TMA) may still occur. Therefore, we check that when an aircraft flies over a node, the horizontal separation with other aircraft is maintained. Considering a node n and two aircraft f, g that fly over node n , we consider a disk centered at node n with radius R_n , defined as a *detection zone*. R_n can be simply defined as 3 NM for all the nodes, we have refined this value with regard to intersection angles of two links at the common node, more details can be found in [19]. We must ensure that at every moment only one aircraft passes this detection zone. Suppose that aircraft f enters the zone of node n before aircraft g . We denote the entering time to and exit time from this zone for aircraft f (g , respectively) as $T_{\text{In}}^{f,n}(\mathbf{x})$ ($T_{\text{In}}^{g,n}(\mathbf{x})$) and $T_{\text{Out}}^{f,n}(\mathbf{x})$ ($T_{\text{Out}}^{g,n}(\mathbf{x})$). A conflict is detected when $T_{\text{In}}^{g,n}(\mathbf{x}) < T_{\text{Out}}^{f,n}(\mathbf{x})$, which means that aircraft g enters the detection zone before aircraft f exits.

We define the *node conflict indicator* for aircraft f (leading) and g (following) as follows:

$$N_{fg}^n(\mathbf{x}) = \begin{cases} 1, & \text{if } T_{\text{In}}^{f,n}(\mathbf{x}) \leq T_{\text{In}}^{g,n}(\mathbf{x}) < T_{\text{Out}}^{f,n}(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases}$$

2.4.2. Runway conflict evaluation

Table 2: Single-runway separation requirements according to aircraft categories and to operations (in seconds). A refers to Arrival, D refers to Departure, and C refers to Crossing. H refers to Heavy, M refers to Medium, and L refers to Light. For example, the minimum runway separation between an ‘‘A-H’’ (Arrival-Heavy) and a ‘‘D-M’’ (Departure-Medium) is 60 seconds.

Operation-Category	Trailing Aircraft							
	A-H	A-M	A-L	D-H	D-M	D-L	C	
Leading Aircraft	A-H	96	157	207	60	60	60	-
	A-M	60	69	123	60	60	60	-
	A-L	60	69	82	60	60	60	-
	D-H	60	60	60	96	120	120	60
	D-M	60	60	60	60	60	60	60
	D-L	60	60	60	60	60	60	60
	C	-	-	-	40	40	40	10

The landing/take-off time difference of any two consecutive aircraft must respect the time separation. The runway separation rules are calculated by incorporating the different flight speeds and their impact on the final approach segment. Here, the separation requirements are shown in Table 2, where A refers to Arrival, D refers to Departure, and C refers to Crossing. Due to the runway configuration in CDG, arrivals have to cross departure runways to get to the terminal. We consider that the crossing time of an arrival is 40 s.

One runway can be modeled as a specific resource with capacity 1. During high traffic demand periods, the upcoming flights may violate the separation rules and cause runway congestions. Therefore, we note the number of times that the separation is violated and the duration of separation violation for all pairs of aircraft as an indicator for our runway evaluation.

We define the *runway conflict indicator* between two successive aircraft f and g as:

$$P_{fg}(\mathbf{x}) = \begin{cases} 1, & \text{if } 0 \leq T_g^r(\mathbf{x}) - T_f^r(\mathbf{x}) < t_{fg} \\ 0, & \text{otherwise} \end{cases}$$

where $T_f^r(\mathbf{x})$ denotes the time at which aircraft f arrives at the runway threshold, and t_{fg} is the minimum runway separation between flights f and g as shown in Table 2.

One particular case must be considered for the departure runway with a sequence of ‘‘Heavy Departure-Crossing-Small/Medium Departure’’. As shown in Fig. 5, each pair of successive flights satisfies to the separation requirement, however, loss of separation occurs between the aircraft ‘‘D-H’’ and the aircraft

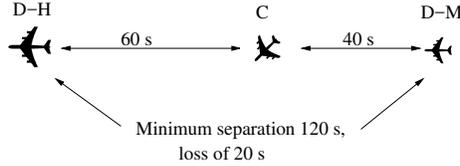


Figure 5: Loss of separation in case “Departure-Heavy (D-H) – Crossing (C)– Departure-Medium (D-M)”. The minimum separation between “D-H” and “C” (“C” and “D-M”) is 60 (40) seconds respectively. However, the minimum separation between “D-H” and “D-M” is 120 seconds, thus it induces 20 seconds loss of separation if only the required separations between successive aircraft are checked.

“D-M”. Therefore, besides detecting the minimum separation between any two successive flights, the loss of triangle inequality as shown in Fig. 5 must be detected too.

The total number of conflicts with regard to decision vector \mathbf{x} is defined as follows:

$$A(\mathbf{x}) = \sum_{\substack{f,g \in \mathcal{F} \\ f \neq g}} \left(\sum_{n \in r_f \cap r_g} N_{fg}^n(\mathbf{x}) + \sum_{l \in r_f \cap r_g} L_{fg}^l(\mathbf{x}) + P_{fg}(\mathbf{x}) \right)$$

The TMA separation and the runway separation are ensured by

$$A(\mathbf{x}) = 0$$

2.5. Objective

Our objective function is a weighted sum of the overloads for terminal and for taxi network and flight delays.

- Terminal and taxiway congestion evaluation:

We have two metrics to measure the terminal congestion. First, the maximum overload number is the maximum value over the period of the difference between the number of aircraft in the terminal and the given terminal capacity. This metric gives us an idea of the time at which severe congestion occurs. However, the maximal overload does not provide sufficient information on the level of congestion. Therefore, another important metric to consider is the average congestion.

Suppose that we have a discretized time window $\mathcal{T} = \{1, 2, \dots, |\mathcal{T}|\}$, let us define the *occupancy indicator* for $i \in \mathcal{T}$:

$$O_m(i) = \text{Card}\{f | T_{\text{In}}^{f,m}(\mathbf{x}) \leq i \leq T_{\text{Out}}^{f,m}(\mathbf{x})\}$$

where $T_{\text{In}}^{f,m}(\mathbf{x})$ and $T_{\text{Out}}^{f,m}(\mathbf{x})$ correspond to the entering time and the exit time of resource m (i.e., terminal or taxi network). It counts the number of aircraft at time i . The overload of resource m at time i is then defined as:

$$G_m(i) = \max\{O_m(i) - O_m, 0\}$$

where O_m is the imposed maximum capacity of the resource m .

The average overload is then defined as $\frac{\sum_{i \in \mathcal{T}} G_m(i)}{|\mathcal{T}|}$.

To conclude, the airside capacity overload is expressed as

$$S(\mathbf{x}) = \frac{\sum_{i \in \mathcal{T}} G_t(i)}{|\mathcal{T}|} + \max_{i \in \mathcal{T}} G_t(i) + \frac{\sum_{i \in \mathcal{T}} G_n(i)}{|\mathcal{T}|} + \max_{i \in \mathcal{T}} G_n(i)$$

where $G_t(i)$ and $G_n(i)$ are respectively the terminal overload and the taxi network overload at time i .

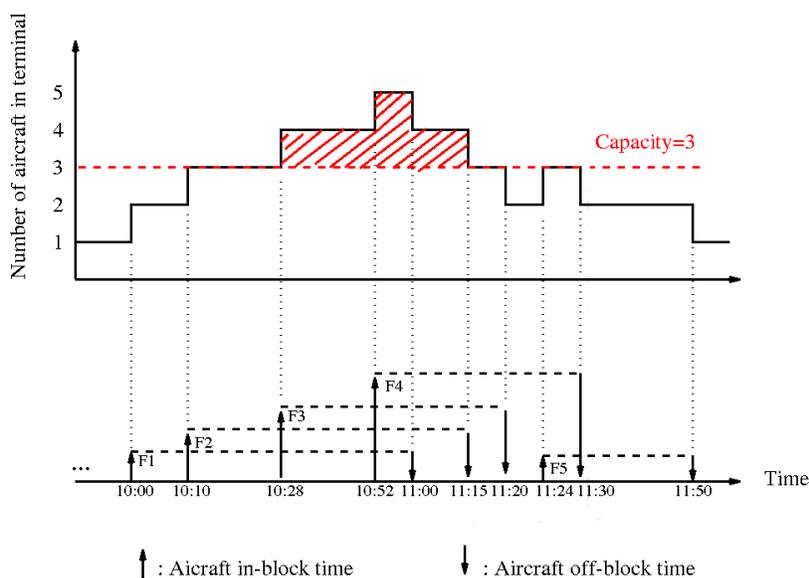


Figure 6: Example of terminal congestion evaluation. Five aircraft turnaround in a terminal with a maximum capacity O_t of 3. The congestion time period is shown in red area.

Let us consider a simple example to show how we propose to measure the terminal congestion level. As illustrated in Fig. 6, suppose that we have one terminal with three gates (i.e., the capacity $O_t = 3$), and 5 flights turnaround in this terminal during a period of two hours, $\mathcal{T} = \{10 : 00, 10 : 01, \dots, 12 : 00\}$. The upward (respectively, downward) arrow represents the

in-block (off-block) time of one aircraft, linked by a dotted line. We count the cumulated number of aircraft in the terminal as time goes by. Here, the maximum terminal occupancy is 5, therefore the maximum overload $\max_{i \in \mathcal{T}} G_t(i)$ is 2. We calculate the total overload $\sum_{i \in \mathcal{T}} G_t(i)$ as well, which is 55 here (the red surface shown in Fig 6). The congestion criterion is $2 + 55/120 \simeq 2.458$.

- Flight delays: The flight delays $D(\mathbf{x})$ are defined as the total time deviation between the optimized and initial values of RTA and pushback time, $D(\mathbf{x}) = \sum_{f \in \mathcal{F}} (p_f - P_f^0) + \sum_{f \in \mathcal{F}} (t_f - T_f^0)$.

The conflict-avoidance constraint is relaxed into the objective function. Thus, our objective function, to be minimized is therefore a weighted sum of these functions:

$$\gamma_a A(\mathbf{x}) + \gamma_s S(\mathbf{x}) + \gamma_d D(\mathbf{x})$$

where γ_a , γ_s , and γ_d are respectively weighting coefficients for the total number of conflicts in airspace, $A(\mathbf{x})$, the airside capacity overload, $S(\mathbf{x})$, and the flight delays $D(\mathbf{x})$.

3. Solution approaches

The complexity of the integrated problem would grow compared to the segregated problem, when in practice the computational time is critical. It is known that the sub-problem of this integrated optimization, aircraft landing scheduling, is NP-hard [22]. Heuristics and hybrid methods may have more potential than exact approaches for tackling this problem [5]. In this paper, we propose a time decomposition approach combined with a simulated annealing algorithm to address the integrated terminal airspace management and airport congestion management. In the following of this section, the time decomposition approach and simulated annealing algorithm are introduced and detailed.

3.1. Time sliding-window decomposition approach

The time sliding-window decomposition approach addresses the original problem by decomposition into several sub-problems using a sliding window in order to reduce the problem size and consequently the computational burden. This specific approach is generic and can be extended and applied to other real-time operation problems.

Suppose that we are given a total time interval, $[t_{\text{INIT}}, t_{\text{FINAL}}]$, over which we want to optimize. Let us introduce some notations:

- W : the time length of the sliding window;
- S : the time shift of the sliding window at each iteration;
- $T_s(k)$: the starting time of the k^{th} sliding window, $T_s(k) = t_{\text{INIT}} + kS$;

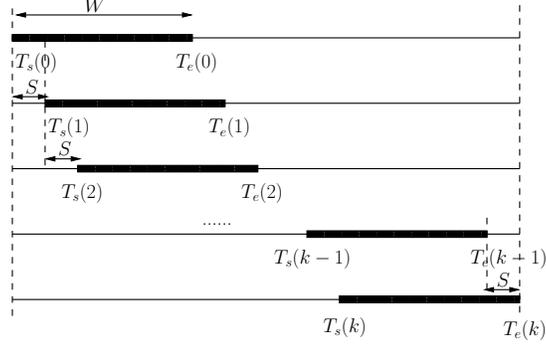


Figure 7: Sliding windows from iteration 0 to iteration k with a time length W and a time shift S at each iteration.

- $T_e(k)$: the ending time of the k^{th} sliding window, $T_e(k) = t_{\text{INIT}} + kS + W$.

Fig. 7 illustrates how the operating window slides along the time axis. The first sliding window begins at t_{INIT} and, the optimization algorithm (to be defined later) is applied in the corresponding time interval $[T_s(0), T_e(0)]$. Next, the sliding window is shifted by time S , and the current optimizing interval becomes $[T_s(1), T_e(1)]$. Then, we repeat the process until we reach the k^{th} sliding window such that $T_e(k) \leq t_{\text{FINAL}} - S$.

Some parameters are needed to describe the sliding-window approach for each flight $f \in \mathcal{F}$:

- t_s^f : initial starting time, i.e.,

$$t_s^f = \begin{cases} T_f^0 & \text{if } f \in \mathcal{A} \cup \mathcal{AD} \\ P_f^0 & \text{if } f \in \mathcal{D} \end{cases}$$

- \underline{t}_s^f : the earliest starting time, i.e.,

$$\underline{t}_s^f = \begin{cases} t_s^f + \Delta t_{\min} & \text{if } f \in \mathcal{A} \cup \mathcal{AD} \\ t_s^f & \text{if } f \in \mathcal{D} \end{cases}$$

- \overline{t}_s^f : the latest starting time, i.e.,

$$\overline{t}_s^f = \begin{cases} t_s^f + \Delta t_{\max} & \text{if } f \in \mathcal{A} \cup \mathcal{AD} \\ t_s^f + \Delta T_{\max}^p & \text{if } f \in \mathcal{D} \end{cases}$$

- t_e^f : initial ending time, i.e.,

- For $f \in \mathcal{A}$, it corresponds to the initial in-block time, which is calculated with regard to initial entry time, STAR route, initial entry speed, average taxi-in duration;

- For $f \in \mathcal{AD}$, it is the exit time of TMA, calculated with regard to initial entry time, STAR route, initial entry speed, average taxi-in duration, turnaround duration, average taxi-out duration, take-off time, and SID route;
 - For $f \in \mathcal{D}$, it is also the exit time of TMA, calculated with regard to earliest off-block time, average taxi-out duration, take-off time, and SID route.
- \underline{t}_e^f : the earliest ending time, i.e.,
 - For $f \in \mathcal{A}$, it corresponds to the earliest in-block time, which is calculated with regard to earliest entry time in TMA, maximum entry speed, STAR route, and average taxi-in duration;
 - For $f \in \mathcal{AD}$, it is the earliest exit time of TMA, calculated with regard to STAR route, earliest entry time in TMA, maximum entry speed, average taxi-in duration, turnaround time, earliest pushback time, average taxi-out duration, take-off time, and SID route;
 - For $f \in \mathcal{D}$, it is also the earliest exit time of TMA, calculated with regard to earliest off-block time, average taxi-out duration, take-off time, and SID route.
 - \overline{t}_e^f : the latest ending time, i.e.,
 - For $f \in \mathcal{A}$, it corresponds to the latest in-block time, which is calculated with regard to latest entry time in TMA, minimum entry speed, STAR route, and average taxi-in duration;

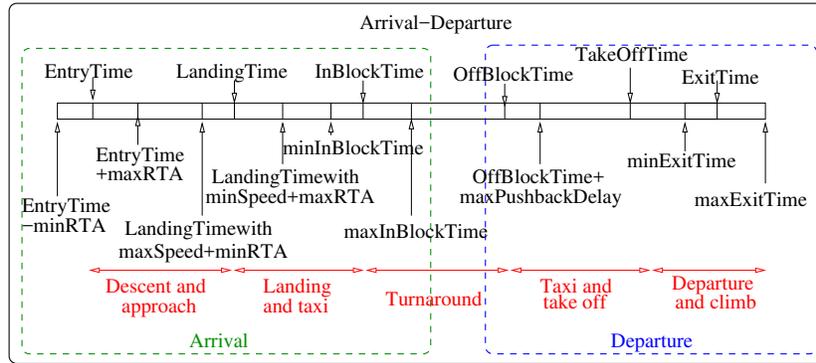


Figure 8: Arrival-Departure operations in TMA. A flight goes through several phases: descent following standard terminal arrival route, landing on the runway and taxiing to the gate, turnaround, push back at the gate, taxiing between the gate and the runway, take-off and initial climb following standard instrument departure procedure.

- For $f \in \mathcal{AD}$, it is the latest exit time of TMA, calculated with regard to STAR route, latest entry time in TMA, minimum entry speed,

- average taxi-in duration, turnaround time, latest pushback time, average taxi-out duration, take-off time, and SID route;
- For $f \in \mathcal{D}$, it is also the latest exit time of TMA, calculated with regard to latest off-block time, average taxi-out duration, take-off time, and SID route.

Fig. 8 gives an overview of the total operations of flight from the entry in TMA until the exit of this TMA.

Each aircraft is classified with one of the following four statuses, based on the positions of the parameters of flight f relative to the starting and ending times of the current sliding window, k :

- **Completed flight:** $\overline{t_e^f} \leq T_s(k)$. The latest ending time for aircraft f , $\overline{t_e^f}$, is lower than the beginning time of the k^{th} sliding window, $T_s(k)$, which means that aircraft f has already finished its operation before the start of the k^{th} sliding window;
- **On-going flight:** $\underline{t_s^f} \leq T_s(k) < \overline{t_e^f}$. The beginning time of the k^{th} sliding window, $T_s(k)$, is between the earliest starting time, $\underline{t_s^f}$, and the latest ending time, $\overline{t_e^f}$, which means that aircraft f has already been assigned, but it may still impact the next aircraft in terms of decision variables;
- **Active flight:** $T_s(k) < \underline{t_s^f} \leq \overline{t_s^f} \leq T_e(k)$. The time decision interval of flight f is included in the sliding window interval $[T_s(k), T_e(k)]$;
- **Planned flight:** $T_e(k) < \overline{t_s^f}$. The latest starting time, $\overline{t_s^f}$, is larger than the ending time of the k^{th} sliding window, $T_e(k)$, which means that the temporal decision variable interval is not totally included in the time window, so that we could not take decision for aircraft f in this interval. The flight will be considered later.

The status of flight f is updated and changed according to the sliding window being considered. Fig. 9 illustrates the four different flight statuses and their positions relative to the sliding window. The different time positions of the aircraft and those of the sliding-window are indicated respectively with blue and red triangles.

At each step, we take into account the active and on-going aircraft in the sliding window interval to be merged and sequenced. Decisions for the on-going flights have already been made, but these flights still have some influence on the decisions to be made for the active flights. On the other hand, the conflicts involving completed flights have already been resolved and they cannot have any impact on the active flights, so they can be cleared out of the decision process and ignored. Then, the optimization window is shifted by the time step S . The aircraft statuses are updated, a new set of flights waiting to be addressed are considered, and the optimization process is repeated, as illustrated in Algorithm 1.

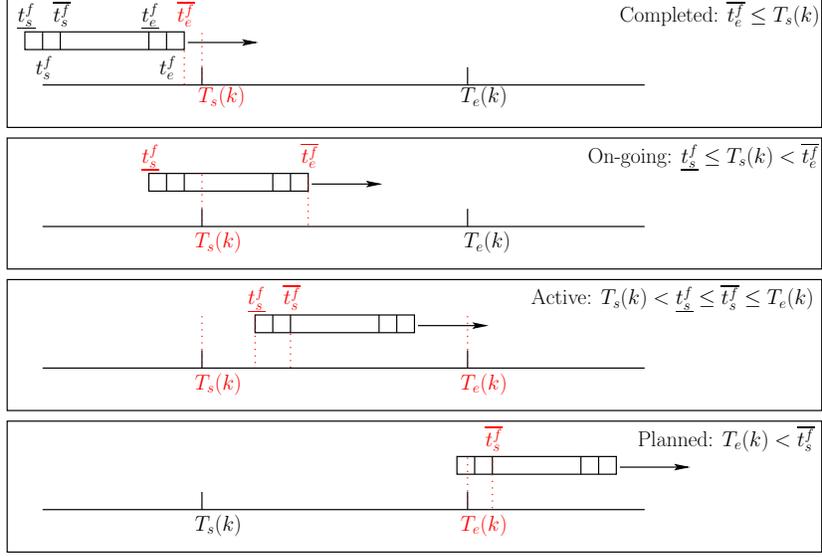


Figure 9: Four flights status, related to the time position of flight f relative to the current sliding window (k) .

Algorithm 1 Sliding-window management

```

1: procedure SLIDINGWINDOW
2:    $k \leftarrow 0$ ;
3:    $T_s(k) \leftarrow t_{\text{INIT}}$ ;
4:    $T_e(k) \leftarrow T_s(k) + W$ ;
5:   Determine each flight status relative to sub-window;
6:    $F_{\text{OPT}} \leftarrow$  Active and on-going flights;
7:   while  $T_e(k) < t_{\text{FINAL}}$  do
8:     if at least one active flight in  $F_{\text{OPT}}$  then
9:       Subproblem: optimize considering  $F_{\text{OPT}}$ ;
10:    end if
11:     $T_s(k) \leftarrow T_s(k) + S$ ;
12:     $T_e(k) \leftarrow T_e(k) + S$ ;
13:     $k \leftarrow k + 1$ ;
14:    Update each flight status relative to sub-window;
15:    Update  $F_{\text{OPT}}$ ;
16:  end while
17: end procedure

```

3.2. Simulated annealing

Simulated Annealing (SA) [23] 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. It can easily be adapted to large-scale problems with continuous or discrete search spaces. In SA, the objective function to be minimized is analogous to the energy of the physical problem. A global parameter T is used to simulate the cooling process. A current solution may be replaced by a random “neighborhood” solution accepted with a probability $e^{-\frac{\Delta E}{T}}$, where ΔE is the difference between corresponding function values. 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 higher temperature, thus the algorithm is able to trap out of local minima. The decrease of temperature may follow different laws, such as linear law, geometric law, etc. At each temperature step, a number of iterations are executed. The probability to accept a degrading solution become smaller and smaller as T decreases. Therefore, at the final stages of the annealing process, the system will converge to a near-global or global optimum.

To generate a neighborhood solution, instead of simply choosing randomly a flight f in the active-flight set, we use a method similar to the so-called roulette-wheel selection. We note for each aircraft the number of conflicts and the time of congestion as its *air* and *ground performance* respectively. Air performance involves link and node conflicts, and ground performance involves runway, taxiway network and terminals congestions. Let us take the example of Fig. 6, in Table 3, we note the total time during which an aircraft is overlapping with other flights. For example, the overlapping time between the flight F_1 and all the other flights is 50 min; for F_5 it is only 6 min.

Table 3: An example of aircraft terminal performance (in minutes)

Flight	F_1	F_2	F_3	F_4	F_5
Terminal performance	50	65	52	34	6

Considering this overload period, it is better to first change the decisions of aircraft which are mostly involved in congestion (F_2, F_3, F_1) than the ones with less impact (F_5) in order to mitigate the terminal congestion. The performance metric can help us to better focus on the most charged and congested periods. The fact that our neighborhood definition is based on the air and ground flight performance increases the likelihood that a flight involved in many conflicts, or experiencing severe congestions, will be chosen. As shown in Fig. 10, in the neighborhood selection, firstly, we record different performance indicators for each aircraft. Then, we choose a flight using a roulette wheel selection method based on the conflict performance. Next, we target this flight to decide which decision variable to be changed. Lastly, we choose randomly a discretized value for the related decision variable. To summarize, a detailed description is shown

Algorithm 2 Neighborhood function

Require: For each flight f , we record its airspace performance, p_f^a , runway performance, p_f^r , ground performance, p_f^g , the total performance is denoted as $p_f^t = p_f^a + p_f^r + p_f^g$.

- 1: The total number of conflicts $P_t = \sum_{f \in \mathcal{F}} p_f^t$;
 - 2: Generate random number, $\nu = \text{random}(0,1)$;
 - 3: **if** $P_t > 0$ **then**
 - 4: $\text{sum} \leftarrow 0$;
 - 5: $\text{target} \leftarrow P_t \times \nu$;
 - 6: $i \leftarrow 1$;
 - 7: **while** $\text{sum} < \text{target}$ **do**
 - 8: $\text{sum} \leftarrow \text{sum} + p_i^t$;
 - 9: $i \leftarrow i + 1$;
 - 10: **end while**
 - 11: **else**
 - 12: Choose randomly one flight i in the flight set;
 - 13: **end if**
 - 14: **if** $i \in \mathcal{A}$ **then**
 - 15: **if** $p_i^a > 0$ **then** choose with equal probability between the entering time and the entering speed in the TMA, then choose randomly one value between 0 and the maximum allowed deviation;
 - 16: **else if** $p_i^r > 0$ **then** choose with equal probability among the entering time in the TMA, the entering speed in the TMA, and the landing runway;
 - 17: **else** choose randomly the entering time in \mathcal{T}_f ;
 - 18: **end if**
 - 19: **else if** $i \in \mathcal{D}$ **then**
 - 20: **if** $p_i^g > 0$ **then**
 - 21: choose randomly the pushback time in \mathcal{P}_f ;
 - 22: **else**
 - 23: choose with equal probability between the pushback time change and the take-off runway change;
 - 24: **end if**
 - 25: **end if**
-

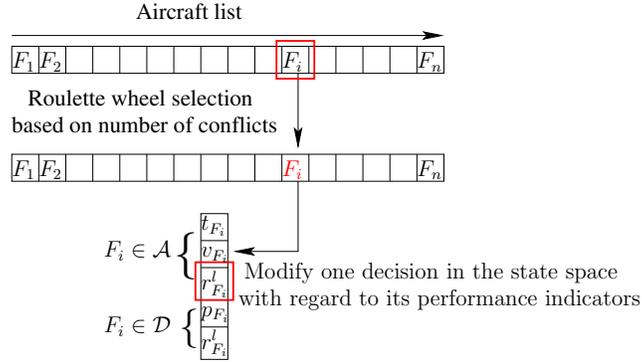


Figure 10: Neighborhood generation. We first target one aircraft using a roulette wheel selection based on number of conflicts. Then, we modify one of its decision variables with regard to the type of aircraft and its performance indicators.

in Algorithm 2.

The SA terminates the execution if either the maximum number of transitions or the minimum temperature are reached.

Fig. 11 summarizes the overall optimization process. The simulation process takes the decision proposed by the optimization algorithm and simulates the associated flight in order to produce the objective function and the vector of performance. The objective function and the performance indicators provided by the simulation process guide the optimization module to search for better solution. The time sliding window manager updates flight status and puts them into the two previous mentioned modules. The optimization and simulation processes are repeated.

In the next section, we apply the simulated annealing algorithm combined with time decomposition approach to resolve the integrated terminal airspace management problem and airport capacity management problem.

4. Results

In this section we present some test problems and analyze the associated results. We tested our methodology on a four-hour real data case at Paris CDG Airport. Numerical results with different settings of (user-defined) algorithm parameters were presented and discussed. The overall process is run on a 2.50GHz core i7 CPU, under Linux operating system PC based on Java code.

4.1. Real data analysis

A busy winter day on February 18th, 2016 was chosen as our data set. Fig. 12 shows the initial terminal and taxi network occupancy over the day, the line color green, blue, orange, and pink respectively represent Terminal 1, Terminal 2, Terminal 3 and taxi network occupancy. Terminal 1 consists of a central circular terminal building and seven satellites with boarding gates, thus

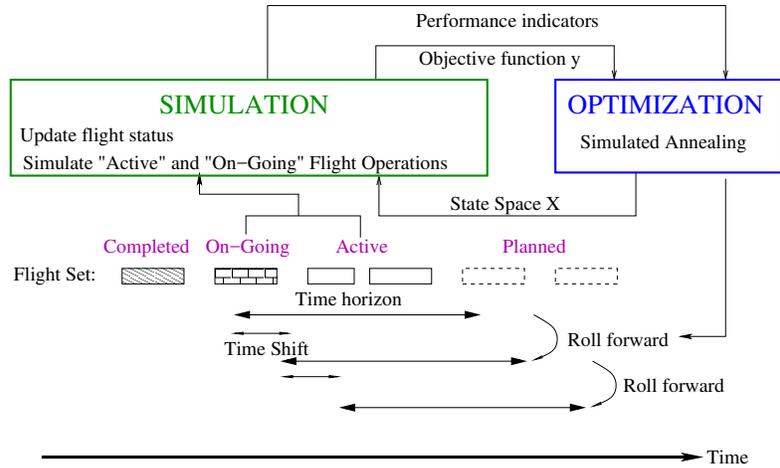


Figure 11: Overall optimization process summary

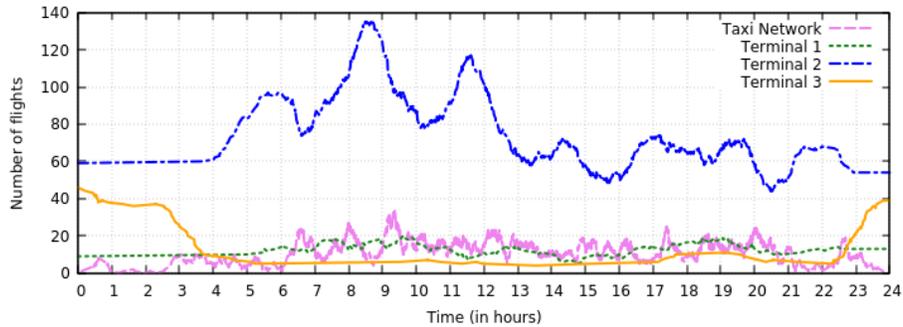


Figure 12: Initial terminals and taxi network occupancy on February 18th, 2016. Terminal 2 is the main terminal in CDG and receives much more traffic flows compared to the other two terminals.

cannot handle many aircraft and keeps a stable low traffic over the day. Air France operates from Terminal 2, and CDG is the principal hub for Air France (hub airport is used by one airline to concentrate passenger traffic and flight operations at a given airport), thus Terminal 2 is the main terminal of CDG that serves the majority of aircraft. Therefore we observed much more traffic flows in Terminal 2 compared to the other two terminals. Terminal 3 mainly hosts charter and low-cost airlines, is mainly composed of hangars for night parking, therefore the departure flights leave the terminal early in the morning and the arrival flights come late at night, forming the curve in orange color shown in Fig. 12. Peak hour with a maximum gate occupancy was reached between 8 am and 10 am in Terminal 2. Then the terminal occupancy decreased sharply, which consecutively led to a peak in the taxi network. Here we extracted the flight data of the most dense time period in the day from 6 am to 10 am as our test

Table 4: User-defined parameter values specifying the optimization problem

Parameter	Value
Discretization time step, ΔT	5 seconds
Discretization speed step, Δv_f	$0.01V_f^0$
Maximum delay of RTA at TMA, ΔT_{\max}	30 minutes
Minimum delay of RTA at TMA, ΔT_{\min}	-5 minutes
Maximum pushback delay, ΔT_{\max}^p	15 minutes
Minimum allowable speed, V_f^{\min}	$0.9V_f^0$
Maximum allowable speed, V_f^{\max}	$1.1V_f^0$
Conflicts weighting coefficient, γ_a	1
Overload weighting coefficient, γ_s	1
Delay weighting coefficient, γ_d	0.001

Table 5: Empirically-set parameter values of the simulated annealing algorithm with time decomposition approach

Parameter	Value
Geometrical temperature reduction coefficient	0.99
Number of iterations at each temperature step	100
Initial rate of accepting degrading solutions	0.15
Final temperature	$10^{-6}T_0$
Time length of the sliding window	2 h
Time shift of the sliding window	0.5 h

problem. A total of 332 flights were operated at CDG, including 177 departures and 155 arrivals, 109 flights were arrival-departures. We have in total 67 Heavy and 265 Medium aircraft. The fleet mix ratio on this period is 20% for Heavy aircraft and 80% for Medium aircraft. The parameters chosen for specifying the optimization problem and the resolution algorithm are respectively given in Table 4 and Table 5.

We tackle the integrated airport and TMA optimization problem at a macroscopic level, the aim is to show that the proposed algorithm can react in the right direction facing airport capacity reduction. Due to lack of data, we cannot apply our method to a historic situation. Moreover, directly comparing the optimized results with the historic situation would somehow be difficult, due to the simplifications and assumptions from the model. In this paper, we build the initial occupancy curve by simulating the process using the initial flight data (initial entering time, initial entering speed, initial pushback time, etc.). We use this curve as baseline case, and impose a reasonable capacity limit, which is fair to compare with the results from the optimization process.

Two major aspects were discussed in the rest of the section: First, different

levels of degradations of the terminal capacity and taxi network capacity were imposed to verify the impact on flight delays and on other airport components. Second, we studied the benefits of runway assignment on reducing flight delays in peak hour when two runways are facing imbalanced throughput.

4.2. Influence of reduced airport capacity to flight delays

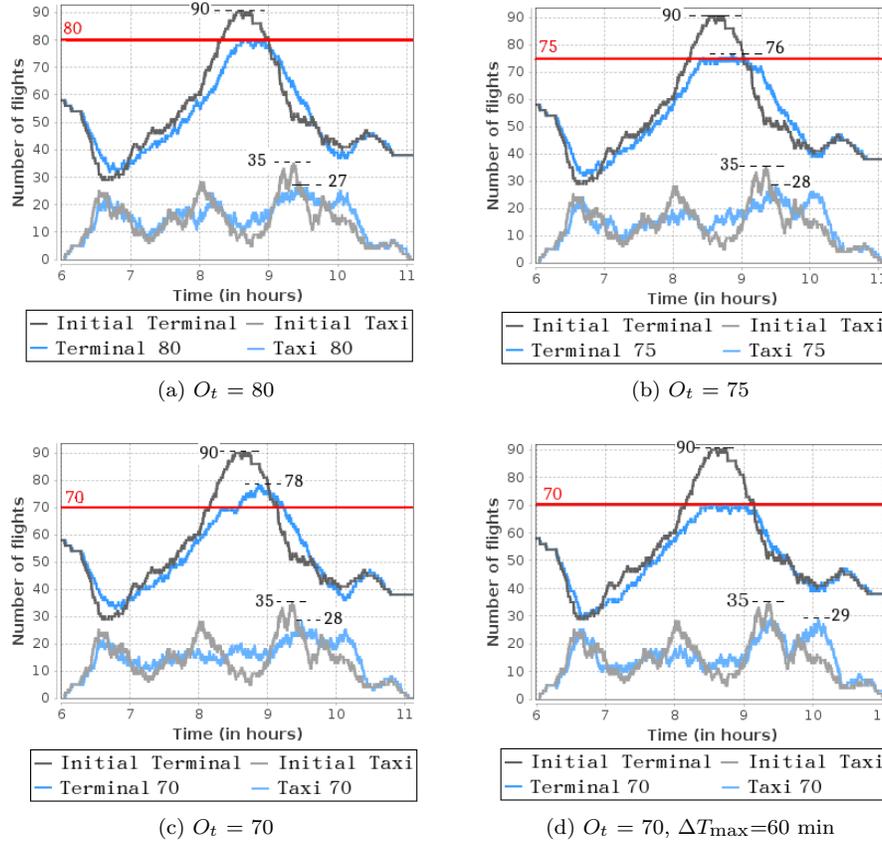
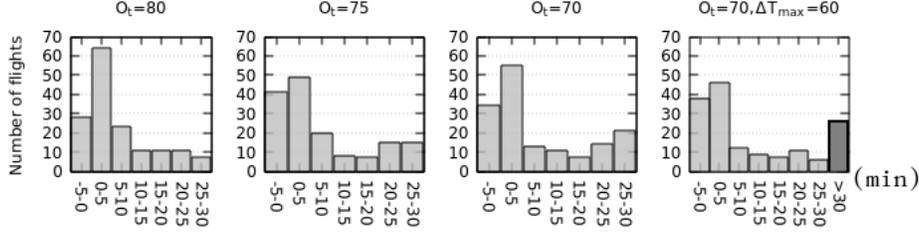
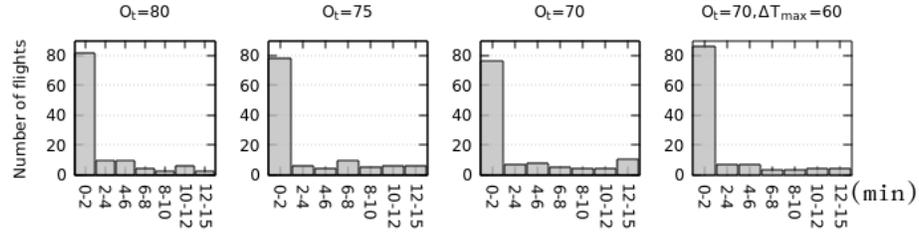


Figure 13: Maximum terminal capacity tests, with $O_t=80, 75, 70,$ and $O_t=70, \Delta T_{\max}=60$ min respectively. Comparison of initial occupancy and optimized occupancy for terminal and taxi network.

First, we investigated how the different levels of degradation of the terminal occupancy and taxi network occupancy would influence the traffic. Two capacity parameters, the imposed maximum terminal capacity, O_t , and the imposed maximum taxi network capacity, O_n , were defined to investigate the airport congestion problem. In the case of terminal overload, we chose to study the traffic on Terminal 2, because it is much more occupied than the other two terminals and has an important peak hour.



(a) RTA delay comparison for arrivals in case $O_t=80$, 75, 70, and $O_t=70$, $\Delta T_{\max}=60$ min respectively. The distribution shifts to the right when O_t decreases. In the fourth case, the dark gray histogram represents the number of flights whose time deviations are greater than 30 minutes.



(b) Pushback delay comparison for departures in case $O_t=80$, 75, 70, and $O_t=70$, $\Delta T_{\max}=60$ min respectively.

Figure 14: Decisions comparison for different maximum terminal capacity O_t

As shown in Fig. 13, the dark gray line and the light gray line respectively represented the initial terminal occupancy and the initial taxi network occupancy. The initial maximum gate occupancy for this period was 90. Therefore, we chose $O_t = 80$, 75, and 70 respectively. We first set a threshold of $O_t = 80$. After running the algorithm, the maximum capacity was reduced and kept below the threshold as illustrated in Fig. 13(a). A decrease of the taxi network occupancy was observed as well. Then we decreased the capacity to $O_t = 75$, in a short period the traffic exceeded this threshold as shown in Fig. 13(b). When the imposed capacity continued decreasing to $O_t = 70$, we encountered a bottleneck and the maximum capacity cannot be reduced anymore. This was due to the inherent maximum allowed RTA delays that we can change. To make a further test, we set the maximum RTA change, ΔT_{\max} , to be 60 minutes instead of 30 minutes. After launching the algorithm, as shown in Fig. 13(d), this terminal overload was totally absorbed with the cost of more than 20 aircraft whose time deviations were more than 30 minutes as shown in Fig. 14(a). We also observed that the RTA distribution shifted to the right as O_t decreases, while pushback delay were not influenced significantly as illustrated in Fig. 14(b).

Similarly, the imposed capacity was applied to taxi network. As shown in Fig. 15, the initial maximum taxi network occupancy on this period was 35. We set a threshold of $O_n = 25$, 20 and 15 to launch three tests separately. In

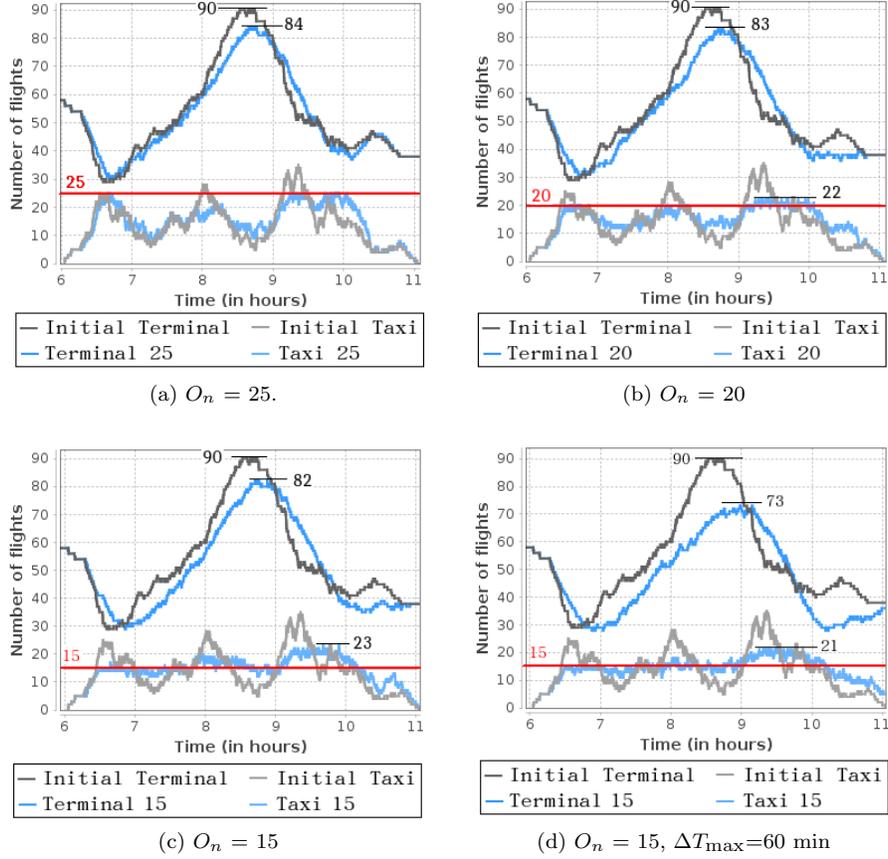
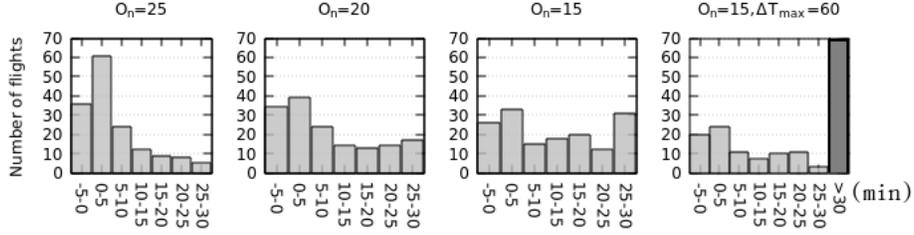
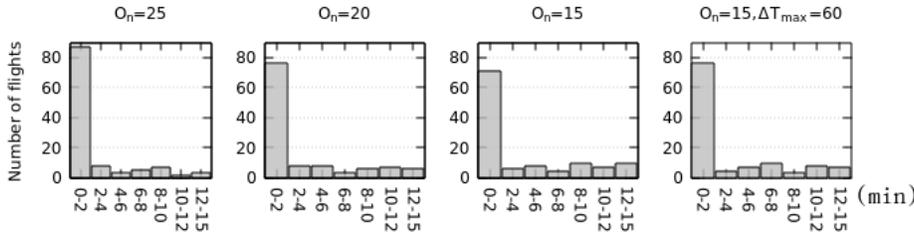


Figure 15: Maximum taxi network capacity tests, with $O_n=25, 20, 15,$ and $O_n=15,$ $\Delta T_{\max}=60$ min respectively. Comparison of initial occupancy and optimized occupancy for terminal and taxi network.

Fig. 15, the dark blue line and the light blue line represented the optimized terminal occupancy and taxi network occupancy respectively. In Fig. 15(a), $O_n = 25$ was easily reached for the whole period after optimization, also a decrease of maximum terminal occupancy was observed, even when we didn't put any constraints on O_t . A sharp increase or decrease of gate occupancy would consecutively increase taxi network capacity as well. As our strategy was to delay the aircraft arrival, the curve was shifted to the right compared to the initial occupancy curve. With $O_n = 20$ in Fig. 15(b), we could see that the traffic overload cannot be absorbed, there was still a maximum taxi network occupancy of 22 around 10 am. With $O_n = 15$ in Fig. 15(c), the limited flight delays cannot absorb the taxi occupancy either, and the maximum value remained almost the same as with $O_n = 20$. To make a further test, we set the maximum RTA change, ΔT_{\max} , to be 60 min instead of 30 min. After



(a) RTA delay comparison for arrivals in case $O_n=25, 20, 15$ respectively. The distribution shifts to the right when O_n decreases.



(b) Pushback delay comparison for departures in case $O_n=25, 20, 15$ respectively. The number of flights whose pushback delay decision change is lower than 2 minutes decreases when O_n decreases.

Figure 16: Decisions comparison for different maximum taxi network capacity O_n

launching the algorithm, as shown in Fig. 15(d), this taxi network overload cannot be absorbed either, in contrast to the terminal overload. This is because only one congestion period was found in the terminal occupancy, while taxi network encountered several levels of congestion in different time periods, thus more unstable and difficult to mitigate under a certain threshold. When we took a look at the decision changes in Fig. 16, the RTA delay and pushback delay distribution shifted to the right when O_n decreased, which indicated that there were more delays for both arrivals and departures. In such case, the algorithm kept aircraft as much as possible at the gate and slowed down arriving aircraft in order to reduce the number of aircraft on the taxi network. Limited capacity of the taxi network caused more flight delays.

4.3. Influence of runway assignment to flight delays

We investigated the benefits of arrival runway assignment and departure runway assignment on reducing flight delays in peak hour when two runways are facing imbalanced throughput.

Paris TMA arrival routes use a four-corner procedure as shown in Fig. 1 in Section 2. In Table 6, southern flows from OKIPA and BANOX mainly use the southern landing runway 26L. Northern flows from MOPAR use more 26L as well, flows from LORNI land more at the northern runway 27R. Moreover, the flows coming from South sometimes land on the northern runway, and vice versa. In practice, landing runway changes can be achieved by controllers' tactical

Table 6: Landing traffic flow distribution with regard to flight entry point in TMA and landing runway on February 18th, 2016.

	27R	26L	Total
MOPAR	39 (6%)	77 (13%)	116 (19%)
LORNI	131 (21%)	73 (12%)	204 (33%)
OKIPA	32 (5%)	165 (27%)	197 (32%)
BANOX	15 (3%)	80 (13%)	95 (16%)
Total	217 (35%)	395 (65%)	612 (100%)

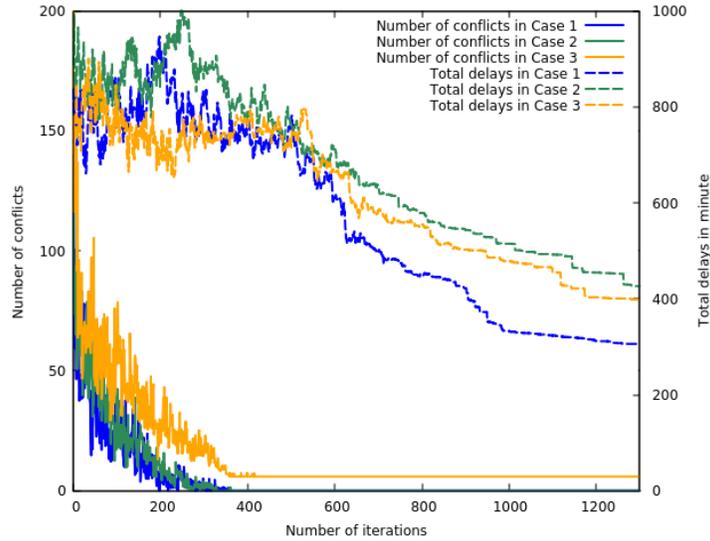


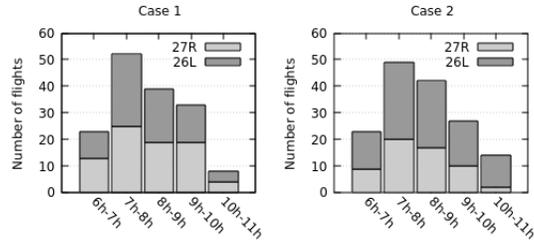
Figure 17: Evolution of the two criteria for different runway decisions

vectoring. The departure runway changes are related to a more detailed level of ground operations, i.e., how alternate taxi routes are assigned. Thus, in this paper, we only focus on the benefits of departure runway changes in imbalanced runway throughput situation.

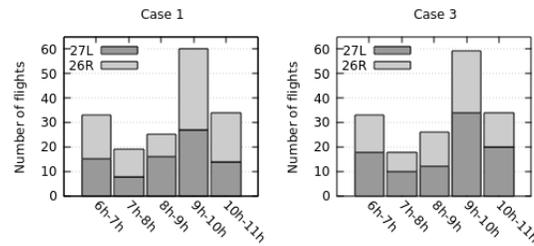
First, we want to investigate how runway changes can bring benefits to reduce flight delays. We set three cases:

- Case 1: Both landing runway and take-off runway are decision variables;
- Case 2: Take-off runway is a decision variable, landing runway is predefined and fixed;
- Case 3: Landing runway is a decision variable, take-off runway is predefined and fixed.

Fig. 17 gave an example of one sliding window optimization evolution; it showed the value of two criteria (number of conflict and total delays) at the end



(a) Landing throughput for runway 27R and 26L. In Case 1, landing runway is assigned; In Case 2, initial landing runway is used.



(b) Take-off throughput for runway 27L and 26R. In Case 1, take-off runway is assigned; In Case 3, initial take-off runway is used.

Figure 18: Landing runways (27R and 26L) throughput comparison for Case 1 and Case 2, and take-off runways (27L and 26R) throughput comparison for Case 1 and Case 3.

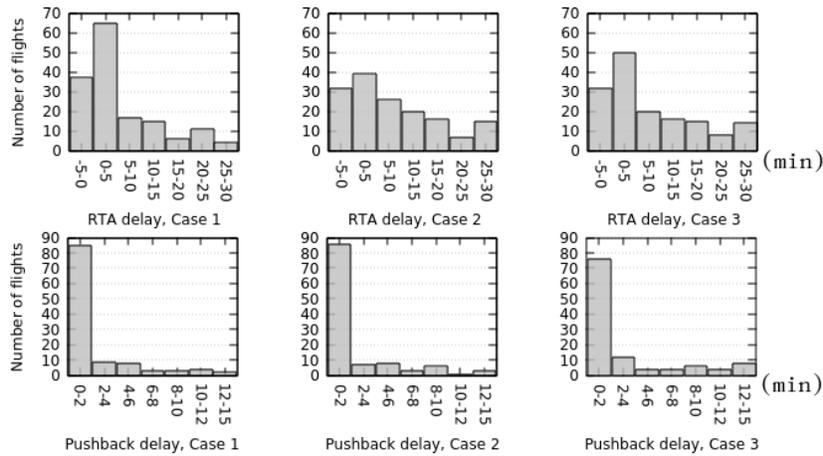


Figure 19: RTA and pushback delay decision changes distribution with take-off and landing runway assignment (Case 1), with only landing runway assignment (Case 2), and with only take-off runway assignment (Case 3).

Table 7: Total RTA delay and pushback delay comparison

	Total RTA delay (in minutes)	Total pushback delay (in minutes)
Case 1	897	443
Case 2	1425	501
Case 3	1293	696

Table 8: Computational time for various problem sizes. The total number of flights are the sum of type “Active” and “On-going”.

Time period	All	Dep.	Arr.	Medium	Heavy	Run time (s)
6:00-8:00	164	101	63	134 (82%)	30 (18%)	51
6:30-8:30	238	137	101	202 (85%)	36 (15%)	85
7:00-9:00	234	123	111	194 (83%)	40 (17%)	88
7:30-9:30	241	119	122	196 (81%)	45 (19%)	93
8:00-10:00	266	127	139	212 (80%)	54 (20%)	103
8:30-10:30	259	121	138	200 (77%)	59 (23%)	104
9:00-11:00	229	106	123	175 (76%)	54 (24%)	85

of each temperature step during the cooling process of SA. Solid lines represented the number of conflicts and dashed lines denoted the total delays in minutes. The number of conflicts in Case 1 converged faster than the other two cases. Case 1 and Case 2 reached conflict-free solution almost at the same time, while in Case 3 conflict-free solution cannot be found. Seven SID conflicts with in total 160 s loss of separation still remained, due to the fact that once the take-off runway was fixed, one can only adjust pushback time to resolve conflicts, thus it was more difficult to find a feasible solution in the given pushback delay period. This result showed that take-off runway assignment did not only balance runway throughput, but may also reduce controllers’ potential maneuvers in peak hour for the SID airspace. As for total delays, before conflict-free solutions were found, delay criteria, for three cases, stayed at a high level to offer more possibilities for the algorithm to search in the state space to establish first a conflict free solution. Then, the delay criterion started decreasing. We can observe that Case 1 reached the lowest delay, while Case 2 remained the highest.

Fig. 18 showed the runway throughput for landings and take-offs. The period 7 am–8 am corresponded to the higher landing throughput, then after the turnaround process, the period 9 am–10 am corresponded to the higher departure throughput. We observed a more balanced traffic for each runway in Case 1 without reducing the throughput. Fig. 19 showed the distribution of flights RTA and pushback delays. In Case 1, a total of 102 arrival flights (66%) modified their RTA within 5 min. While in Case 2 and 3, more RTA delays were required compared to Case 1, only 46% of the flights RTA was less than

5 min, and 54% in Case 3. Without the take-off runway changes, much more pushback delays were requested. Pushback delay didn't change as significantly as RTA delay, because we had a low demand between 6 am and 9 am, the major departure flows occurred between 9 am and 10 am. Regarding the total RTA delay and pushback delay, as shown in Table 7, a decrease of 37 % (from 1425 min to 897 min) RTA delay was reached for Case 1 compared to Case 2. A decrease of 36 % (from 696 min to 443 min) pushback delay was reached for Case 1 compared to Case 3.

The average computational time of our optimization algorithm was 13 min with in total of 10 sliding windows. As shown in Table 8, the average CPU time for each window was less than 2 min, which is very promising for tackling such a complicated problem in practice.

5. Conclusions

To address the connected airport and terminal airspace management problems, this paper proposed a model to manage the arrival, surface and departure problems at the macroscopic level. The objective was to resolve conflicts in the terminal airspace, to reduce airside capacity overload, and to reduce flight delays. First, we proposed a TMA route network structure and a high level airport abstraction model. Then, a time sliding-window approach combined with a simulated annealing algorithm was applied to solve the problem. The approach was implemented in the case of Paris CDG airport and showed some potential benefits: First, reduced terminal capacity until a certain threshold was efficiently mitigated by RTA and pushback time changes. When the imposed capacity was more reduced, the overload could not be mitigated anymore, and the airport could not absorb more demand without imposing delays out of the maximum range. Similarly to terminal occupancy, a decrease of the maximum taxi network capacity could be mitigated by delaying arrivals. Second, landing runway assignment and take-off runway assignment in peak hour with imbalanced runway throughputs could significantly reduce the flight delays. Moreover, the conflicts in the airspace could be resolved also, which may imply that the runway change did not create many more controller's workload.

The next steps for this research would integrate a more precise microscopic level to optimize the ground movements by considering individual flights and gates. Uncertainties about the flight arrival time, pushback time and taxi duration should be taken also into account as well.

Acknowledgments

This work has been partially supported by Civil Aviation University of China (CAUC), by China Scholarship Council (CSC) and by the National Science Foundation of Tianjin through Grant No. 17JCYBJC43100. The authors would like to thank Serge Roux for his assistance with data and technical support. The authors would like to thank SNA-RP/CDG-LB for providing the traffic data. The authors thank the anonymous reviewers for their insightful comments.

References

- [1] Eurocontrol, “European atm master plan,” 2015.
- [2] Federal Aviation Administration, “Nextgen implementation plan 2016,” 2016.
- [3] Y. Günther, A. Inard, B. Werther, M. Bonnier, G. Spies, A. Marsden, M. Temme, D. Böhme, R. Lane, and H. Niederstraßer, “Total Airport Management (operational concept and logical architecture),” Eurocontrol, 2006.
- [4] J. A. Bennell, M. Mesgarpour, and C. N. Potts, “Airport runway scheduling,” *4OR: A Quarterly Journal of Operations Research*, vol. 9, no. 2, pp. 115–138, 2011.
- [5] J. A. Atkin, E. K. Burke, and S. Ravizza, “The airport ground movement problem: Past and current research and future directions,” in *Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT), Budapest, Hungary*, 2010, pp. 131–138.
- [6] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, “Scheduling aircraft landings—the static case,” *Transportation science*, vol. 34, no. 2, pp. 180–197, 2000.
- [7] J. A. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson, “Hybrid metaheuristics to aid runway scheduling at london heathrow airport,” *Transportation Science*, vol. 41, no. 1, pp. 90–106, 2007.
- [8] H. Balakrishnan and B. G. Chandran, “Algorithms for scheduling runway operations under constrained position shifting,” *Operations Research*, vol. 58, no. 6, pp. 1650–1665, 2010.
- [9] F. Furini, M. P. Kidd, C. A. Persiani, and P. Toth, “Improved rolling horizon approaches to the aircraft sequencing problem,” *Journal of Scheduling*, vol. 18, no. 5, pp. 435–447, 2015.
- [10] J.-B. Gotteland, N. Durand, J.-M. Alliot, and E. Page, “Aircraft ground traffic optimization,” in *ATM 2001, 4th USA/Europe Air Traffic Management Research and Development Seminar*, 2001.
- [11] R. Deau, J.-B. Gotteland, and N. Durand, “Airport surface management and runways scheduling,” in *ATM 2009, 8th USA/Europe Air Traffic Management Research and Development Seminar*, 2009.
- [12] H. Lee and H. Balakrishnan, “A comparison of two optimization approaches for airport taxiway and runway scheduling,” in *Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st*. IEEE, 2012, pp. 4E1–1.

- [13] Y. C. Jung, T. Hoang, J. Montoya, G. Gupta, W. Malik, and L. Tobias, “A concept and implementation of optimized operations of airport surface traffic,” 2010.
- [14] J. Ma, D. Delahaye, M. Sbihi, P. Scala, and M. M. Mota, “A study of tradeoffs in airport coordinated surface operations,” in *EIWAC 2017, 5th ENRI international workshop on ATM/CNS*, 2017.
- [15] H. Khadilkar and H. Balakrishnan, “Integrated control of airport and terminal airspace operations,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 216–225, 2016.
- [16] M. Xue and S. Zelinski, “Optimal integration of departures and arrivals in terminal airspace,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 207–213, 2013.
- [17] C. Bosson, M. Xue, and S. Zelinski, “Optimizing integrated arrival, departure and surface operations under uncertainty,” in *10th USA/Europe ATM R&D Seminar (ATM2015), Lisbon, Portugal*, 2015.
- [18] M. J. Frankovich, “Air traffic flow management at airports: A unified optimization approach,” Ph.D. dissertation, Massachusetts Institute of Technology, 2012.
- [19] J. Ma, D. Delahaye, M. Sbihi, and M. Mongeau, “Merging flows in terminal maneuvering area using time decomposition approach,” in *7th International Conference on Research in Air Transportation (ICRAT 2016)*, 2016.
- [20] —, “Integrated optimization of terminal manoeuvring area and airport,” in *6th SESAR Innovation Days (2016).*, 2016, pp. ISSN–0770.
- [21] S. Chaimatanan, D. Delahaye, and M. Mongeau, “A hybrid metaheuristic optimization algorithm for strategic planning of 4d aircraft trajectories at the continental scale,” *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 46–61, 2014.
- [22] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, “Scheduling aircraft landings - The static case,” *Transportation Science*, vol. 34, no. 2, pp. 180–197, 2000.
- [23] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, “Optimization by simulated annealing,” *science*, vol. 220, no. 4598, pp. 671–680, 1983.