

Integrated Optimization of Arrival, Departure, and Surface Operations

Ji Ma, Daniel Delahaye, Mohammed Sbihi, Paolo Maria Scala

▶ **To cite this version:**

Ji Ma, Daniel Delahaye, Mohammed Sbihi, Paolo Maria Scala. Integrated Optimization of Arrival, Departure, and Surface Operations. ICRAT 2018, 8th International Conference for Research in Air Transportation, Jun 2018, Barcelone, Spain. hal-01823188

HAL Id: hal-01823188

https://enac.hal.science/hal-01823188

Submitted on 26 Jun 2018

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 Arrival, Departure, and Surface Operations

Ji Ma, Daniel Delahaye, Mohammed Sbihi
ENAC – Université de Toulouse
Toulouse, France
[ji.ma | daniel.delahaye | mohammed.sbihi]@enac.fr

Paolo Scala
Amsterdam University of Applied Sciences
Amsterdam, The Netherlands
p.m.scala@hva.nl

***Abstract*—Airports and surrounding airspaces are limited in terms of capacity and represent the major bottleneck in the air traffic management system. This paper proposes a two-level model to tackle the integrated optimization problem of arrival, departure, and surface operations. The macroscopic level considers the terminal airspace management for arrivals and departures and airport capacity management, while the microscopic level optimizes surface operations and departure runway scheduling. An adapted simulated annealing heuristic combined with a time decomposition approach is proposed to solve the corresponding problem. Computational experiments performed on real-world case studies of Paris Charles De-Gaulle airport, show the benefits of this integrated approach.**

Keywords-Airport Operations; Terminal Maneuvering Area; Integrated Optimization; Simulated Annealing

I. INTRODUCTION

With the steady growth of air traffic, 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 saturated airfield capacity will adversely impact predictability and punctuality. The current network traffic throughput need to be increased in order to accommodate the forecast demand with a sufficient margin. To achieve this goal, new operational concepts and techniques need to be developed to support the increased traffic density. Efficient planning and optimization approaches of airport operations and surrounding terminal airspaces are critical to alleviate traffic congestions.

In some previous works, segregated problems on arrival management (AMAN), departure management (DMAN) and surface management (SMAN) have been studied extensively. Bennell *et al.* [1] gave a brief review about the techniques and tools for scheduling aircraft landings and take-offs. Atkin *et al.* [2] provided an overview of the research for ground movement and the integration of various airport operations. Recently, more efforts are made on integrated optimization models for airside (runway, taxiway and terminal) and airspace (terminal airspace) operations.

Integrating terminal airspace management with existing route network is a more complicated, but more realistic problem than considering only the previous-mentioned segregated problems. Khadilkar and Balakrishnan [3] modeled departure operations using a network abstraction, and combined published arrival routes, used dynamic programming to solve the integrated control problem. Bosson *et al.* [4] formulated spatial and temporal separations in terminal airspace and extended with surface operations to integrate taxiway and runway operations. Frankovich [5] proposed unified approaches on both strategic and tactical levels to optimize the traffic flowing through an airport. In general, combining the airside and airspace problem and optimize together can gain more benefit. However, the complexity of the integrated problem would grow significantly as well.

In this work, we propose a methodology to address the integrated problem of AMAN, DMAN, and SMAN. This optimization problem is divided into two levels: a macroscopic level and a microscopic level. At the macroscopic level, we consider the terminal airspace management for arrivals and departures and airport capacity management through the abstraction model of terminal, taxi network, and runway. At the microscopic level, the ground movement and the departure runway scheduling problem with a detailed description of the airport are considered.

The remaining parts of this paper are organized as follows. Section II presents the mathematical model of the macroscopic level optimization and the microscopic level optimization. A metaheuristic method combined with a time decomposition approach for solving the integrated problem is presented in Section III. Computational experiments with the proposed methodology are presented in Section IV. Conclusions and perspectives are discussed in Section V.

II. PROBLEM DESCRIPTION AND MATHEMATICAL FORMULATION

Coordination between the AMAN, DMAN, and SMAN is an important issue to improve the efficiency and the predictability of airports. Current AMAN systems are able to predict the landing times 30 minutes in advance with quite a good accuracy [6], while SMAN and DMAN face more uncertainties of taxi time, start-up time, takeoff time etc. The complex interactions and different anticipation times between AMAN, DMAN and SMAN schedules motivate us to tackle the airport traffic management at two levels: First, at the macroscopic level, a Terminal Maneuvering Area (TMA) route graph for arrival and departure is used for conflict detection and resolution in the airspace. The airside is modeled as an abstraction network: terminal, taxi network, and runway are

seen as specific resources with a defined maximum capacity. Optimization will be carried out from the entry of TMA until the exit of this TMA. Secondly, the optimized results from the macroscopic level will be put into the microscopic level, ground operations and departure runway scheduling with a detailed description of the airport are considered and optimized. Mathematical models of the two-level optimization problem are presented in the remaining parts of this section.

A. Macroscopic model of integrated airport and TMA traffic management

This subsection describes the integrated optimization problem of airport and TMA at the macroscopic level, in order to resolve airspace conflicts (which implicitly represents potential workload of air traffic controllers), to mitigate airport congestions and to reduce delays. We first describe the network model of TMA and airport surface. Then, we present an integrated global optimization model of TMA and airport, giving flight input data, defining decision variables, clarifying constraints, and introducing objective function.

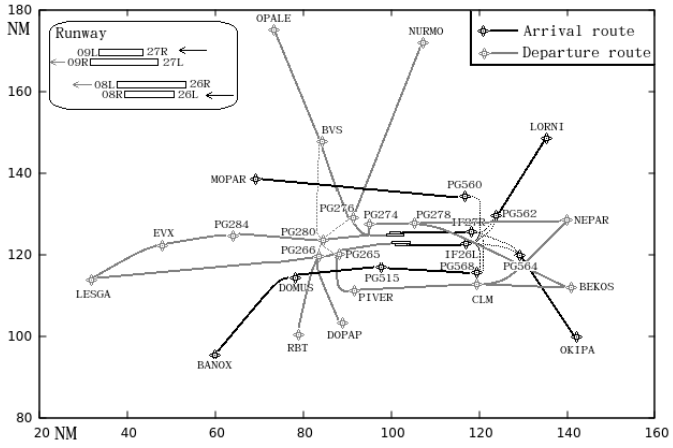


Fig. 1. Terminal route network of arrivals and departures in CDG in west configuration

1) Network model of TMA and airport surface: We model the TMA arrival and departure routes by a graph, $\mathcal{G}(\mathcal{N}, \mathcal{L})$, in which the aircraft are allowed to fly in TMA airspace, where $\mathcal{N}$ is the node set and $\mathcal{L}$ is the link set. Each route is defined by a succession 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 TMA route network of Paris Charles De-Gaulle (CDG) airport. CDG is one of the busiest passenger airports in Europe, composed of four parallel runways (two for landings and two for take-offs) and three terminals. The west configuration with runway 26L/26R and 27L/27R is illustrated in Fig. 1, arrival and departure procedures are represented by black and gray colors respectively. 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 $\mathcal{N}_e$ = {MOPAR, LORNI, OKIPA, BANOX}. For the departure procedure, one route starts at the runway threshold and ends with one of the SID waypoints in the set $\mathcal{N}_x$ = {OPALE, NURMO, NEPAR, BEKOS, DOPAP, RBT, LESGA}.

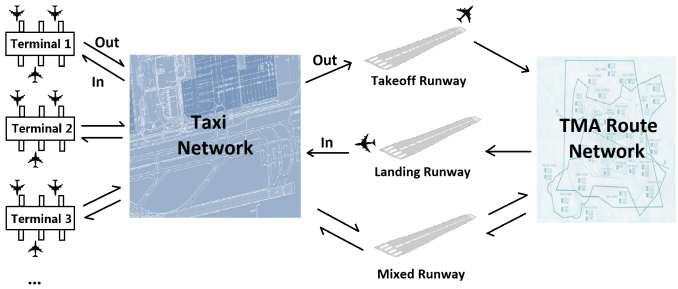


Fig. 2. Network model of TMA and airport surface

Different components of airport 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.

2) Given data: Assume that we are given a set of flights (or aircraft), $\mathcal{F} = \mathcal{A} \cup \mathcal{D} \cup \mathcal{AD}$, where $\mathcal{A}$ is a set of arrivals, $\mathcal{D}$ is a set of departures and $\mathcal{AD}$ is a set of arrival-departures, *i.e.*, aircraft that arrive at the airport and depart again after a turnaround duration.

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 $f \in \mathcal{A} \bigcup \mathcal{AD}$, exit waypoint at TMA for $f \in \mathcal{D} \bigcup \mathcal{AD}$, taxi-in duration for $f \in \mathcal{A} \bigcup \mathcal{AD}$, taxi-out duration for $f \in \mathcal{D} \bigcup \mathcal{AD}$, initial landing runway number for $f \in \mathcal{A} \bigcup \mathcal{AD}$, initial departure runway number for $f \in \mathcal{D} \bigcup \mathcal{AD}$, 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 $f \in \mathcal{D} \bigcup \mathcal{AD}$. Moreover, we know:

- T_f^0: initial RTA (Required Time of Arrival) at the entering waypoint of TMA ($f \in \mathcal{A} \bigcup \mathcal{AD}$);
- V_f^0: initial speed at the entering waypoint of TMA ($f \in \mathcal{A} \bigcup \mathcal{AD}$);
- P_f^0: initial off-block time ($f \in \mathcal{D} \bigcup \mathcal{AD}$), 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 don't have information about gates at the macroscopic level;

- Each aircraft has a constant deceleration or acceleration in TMA.

3) Decision variables: The optimization model we are using has five types of decision variables. First, we have to decide entering time at TMA, entering speed at TMA, and landing runway for arrivals:

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

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

  where ΔT is a discretized time increment, whose value is to be set by the user. In this paper, we choose ΔT = 5 seconds, $\Delta T_{\min}$ = -5 minutes, $\Delta T_{\max}$ = 30 minutes.
- Entering speed in TMA for $f \in \mathcal{A} \bigcup \mathcal{AD}$: We define an entering speed decision variable $v_f \in \mathcal{V}_f$, where

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

  where Δ_f^v is a (user-defined) speed increment, $V_f^{\min}$ and $V_f^{\max}$ are given input data corresponding to the minimum and maximum allowable speeds for aircraft f. In this study, we set $V_f^{\min} = 0.9V_f^0$, $V_f^{\max} = 1.1V_f^0$ and $\Delta_f^v = 0.01V_f^0$.
- Landing runway for $f \in \mathcal{A} \bigcup \mathcal{AD}$: r_f^l is the landing runway decision for arrivals. Runway aircraft assignment is used to balance the capacity when one runway gets overloaded while another is still able to accommodate more aircraft. It enables to increase overall throughput with less delay comparing with the case where aircraft have no options to change their landing runway.

  Secondly, we decide departure runway and pushback time for departures:
- Departure runway for $f \in \mathcal{D} \bigcup \mathcal{AD}$: r_f^d is the departure runway decision for departures. Similarly, it's possible to yield flights to another departure runway when the current assigned one is busy.
- Pushback time for $f \in \mathcal{D} \bigcup \mathcal{AD}$: We define a pushback time decision variable $p_f \in \mathcal{P}_f$, where

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

  In contrast to the entering time decision at TMA for arrival flights, we can only delay departure with regard to its earliest initial off-block time. $\Delta T_{\max}^p = 15$ minutes in this study.

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

4) Constraints and Objectives: We have three main constraints: wake turbulence separation as shown in Table I, single runway separation for arrivals and departures as shown in Table II and III respectively, and maximum capacities for terminal and for taxi network. In order to take into account these constraints, the model is designed to resolve conflicts in the air, to reduce airside capacity overload, and to reduce flight delays. The number of conflicts is evaluated by node and link conflicts detection. The airside capacity overload involves runways, terminals and taxiway network evaluation. The flight delays include total RTA delay and total pushback delay. We make a relaxation of these constraints into our objectives.

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 weighting coefficients for the total number of conflicts in airspace, $A(\mathbf{x})$, the airside capacity overload, $S(\mathbf{x})$, and the flight delays $D(\mathbf{x})$ respectively. $D(\mathbf{x})$ are defined as the total time deviation between the optimized and initial values of RTA and pushback time. Next, we will introduce how to evaluate the airspace conflicts $A(\mathbf{x})$ and the airport congestions $S(\mathbf{x})$.

5) Conflicts detection in the TMA: In this paper, we make an assumption that arrival routes 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, two kinds of conflicts are defined:

- Link conflict: For each given link, we check twice whether a conflict occurs, *i.e.*, the minimum wake turbulence separation (shown in Table I) is violated: at the entry and at the exit of the link. Moreover, we ensure that the order of aircraft sequence remains the same along the link. This enables to detect aircraft passing another one.

TABLE I
DISTANCE-BASED SEPARATION ON APPROACH AND DEPARTURE ACCORDING TO AIRCRAFT CATEGORIES (IN NM).

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

- 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.

Note that once the decision variable values are set, we can calculate the corresponding times at which the aircraft passes

each node and each link. Then, we use these time values to evaluate the number of link and node conflicts. The conflict detection methodology is described in detail in [7].

6) Congestion evaluation in the ground side: The ground side capacity congestion involves runways, terminals and taxi network evaluation. $S(\mathbf{x})$ is evaluated by accumulating the sum of runway separation violation, terminal overload and taxi network overload.

TABLE II
SINGLE-RUNWAY SEPARATION REQUIREMENTS FOR ARRIVALS (IN SECONDS).

Category		Trailing Aircraft		
		Heavy	*Medium*	*Light*
Leading Aircraft	*Heavy*	96	157	207
	Medium	60	69	123
	Light	60	69	82

TABLE III
SINGLE-RUNWAY SEPARATION REQUIREMENTS FOR DEPARTURES AND ARRIVAL CROSSINGS (IN SECONDS).

Category		Trailing Aircraft			
		Heavy	*Medium*	*Light*	*Crossing*
Leading Aircraft	*Heavy*	90	120	120	60
	Medium	60	60	60	60
	Light	60	60	60	60
	Crossing	40	40	40	10

- Runway congestion evaluation: 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 for arrivals (as shown in Table II), for departures and between departures and arrival crossings (shown in Table III) and cause runway congestions. Therefore, we note the sum of accumulated time of separation violation for all pairs of aircraft and the total number of conflicts as an indicator for our runway evaluation.
- Terminal and taxiway congestion evaluation:
  We have two metrics to measure the terminal congestion. First, the maximum overload number is calculated based on the difference between the actual total 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 amount of time during which aircraft experience congestions in the terminal. Let us consider simple example to show how we propose to measure the terminal congestion level. As illustrated in Fig. 3, suppose that we have one terminal with three gates (*i.e.*, the capacity is 3), and 5 flights turnaround in this terminal. The upward (respectively, down) 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 maximal terminal occupancy is 5, therefore the maximal overload is 2. We calculate the total congestion time as well, which is 47+8=55 minutes here (the red surface shown in Fig. 3). The taxiway network congestion can be measured in a similar way. We note as well the maximum overload and the total congestion time for taxi network evaluation. More details about congestion evaluation can be found in our previous work [8].

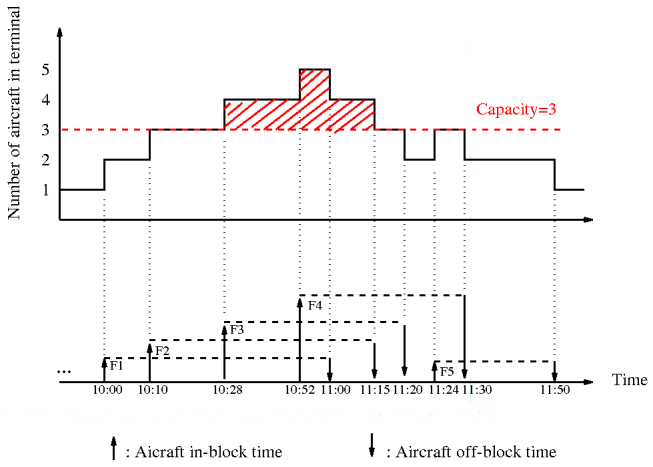


Fig. 3. Example of terminal congestion evaluation

B. Microscopic Model of ground operations and runway scheduling

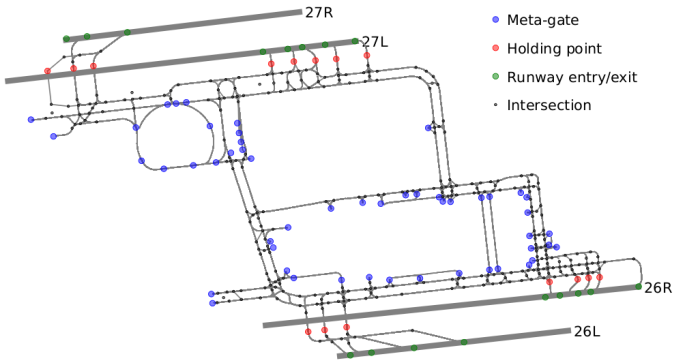


Fig. 4. CDG airport model in the west configuration at the microscopic level

After optimization at the macroscopic level, flight landing time, landing runway and departure runway assignment are fixed, and pushback time is an initial input. We then consider a detailed model of airport including runway holding points, taxiways, gates at the microscopic level to find optimized taxi route, start-up time at gate, runway waiting time, and takeoff time. In CDG, Ground controllers handle all intermediary taxiing routes. Local controllers and Apron controllers handle respectively the runway area and parking areas [9]. Due to

this different area classification and in order to simplify the problem, our model considers that taxiway starts with a defined *meta-gate* shown in Fig. 4, which is the exit point of the ramp area and the entry point of the taxiway area, and ends with runway entry point for departures. For arrivals, taxiing path starts with runway exit point, and ends in meta-gate. We model CDG airport with a node-link graph. Each node can be a runway entry/exit point, a holding point, an intersection or a meta-gate. Each link is composed of two nodes. We have in total 392 nodes and 617 links.

1) Input data: With a short planning horizon at the microscopic level, the previous mentioned arrival-departure is divided into two operations: arrival and departure. For each $f \in F$, the following input data are given: wake turbulence category, meta-gate, runway entry point for departure or runway exit point for arrival, initial holding point at runway threshold, and a set of alternate routes, R_f, connecting the origin and the destination of f. Moreover, the optimized landing time, assigned landing runway for arrival and the optimized off-block time, assigned takeoff runway for departure at the macroscopic level become input data at the microscopic level.

We have some assumptions in order to simplify the problem while keeping some level of reliability.

- Aircraft taxi with a constant speed for a given link;
- Ramp area is beyond the scope of this work, instead we use the notion of meta-gate.

2) Decision variables: For each flight $f \in F$, the decision variables are defined as follows:

- $r_f \in R_f$: taxi-in or taxi-out route. More details about alternate taxiing routes generation can be found in [10];
- t_f^h: holding time (waiting time at runway threshold for departures and time spent in runway crossing queues for arrivals). The maximum holding times for departures and arrivals are 10 minutes and 3 minutes respectively. It is discretized with a time slot of 5 seconds;
- p_f: pushback time. The maximum pushback delay is 15 minutes. Similarly, it is discretized with a time slot of 5 seconds;

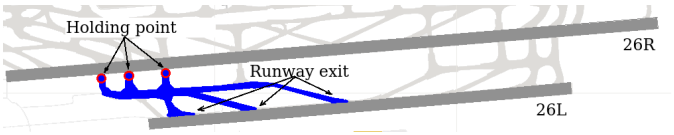


Fig. 5. CDG south side runway layout

- h_f: holding point for arrival. CDG south-side runway layout shown in Fig. 5 motivates us to use arrival holding point as decision variable. In reality, simultaneous flight crossings can enhance departure runway throughput.

3) Constraints: Airport operational constraints are taken into account:

- Minimum taxi separation of s = 60 meters [11] between two taxiing aircraft.
- Departure runway wake turbulence separations considering arrival crossings shown in Table III.
- Holding point capacity (the maximum number of flights waiting at holding point). For arrivals, it is usually one or two due to the fact that a landing flight can not hold too long time to vacate the position for the next landings. For departures, it's a parameter called runway pressure adjusted by controllers considering demand over the period.

Based on the route network structure in Fig. 4, and in order to express the previous mentioned separation standards and capacity constraints, we define four types of conflicts: node conflict, link conflict, runway conflict, and holding conflict depending on the decision variables.

- *Ground node-link conflict:* We apply a conflict detection method similar to the macroscopic level with different separation standards. Moreover, we add the bi-directional link conflict detection on the ground, *i.e.*, two aircraft using a link in opposite directions simultaneously. More details can be found in [10];
- *Runway conflict:* For each runway and for two successive flights (departure or arrival crossing), same method is applied as in the macroscopic model to detect the separation violation.
- *Holding conflict:* For each holding point, we first make sure that the sequence of waiting flights remains the same. Then, by calculating the maximum number of aircraft simultaneously waiting in the queue, we compare it with the maximum holding capacity. If it exceeds the maximum holding capacity, we increase the total number of conflicts by the exceeded capacity.

The total sum of these four types of conflicts is denoted as C. Then the previous separation and capacity constraint is transformed to $C = 0$ (conflict-free).

4) Objectives: The objective function that we want to minimize is:

$$C + \alpha \sum_{f \in D} (p_f - P_f^0) + \beta \sum_{f \in F} h_f,$$

where

- C : Total number of conflicts;
- $\sum_{f \in D} (p_f - P_f^0)$: Total pushback delay;
- $\sum_{f \in F} h_f$: Total holding time.

and α and β are weighting coefficients corresponding to pushback delays and holding time respectively. Even though we use relaxation on conflict constraints, small values of weighting coefficients are chosen in order to ensure that conflict-resolution is the first priority, and conflict-free solution is reached at last.

Next, we present the solution approaches to solve this two-level integrated optimization problem.

III. SOLUTION APPROACHES

It is known that even the sub-problem of this integrated optimization problem at the macroscopic level, aircraft landing scheduling, is NP-hard [12]. Since the complexity of the

integrated problem would grow, when in practice the computational time is critical. Heuristics and hybrid methods may have more potential than exact approaches for tackling this problem [2]. In this paper, we propose a time decomposition approach combined with a simulated annealing algorithm to address this problem. In the following of this section, the time decomposition approach and simulated annealing algorithm are introduced in detail.

A. Time sliding-window decomposition approach

This 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.

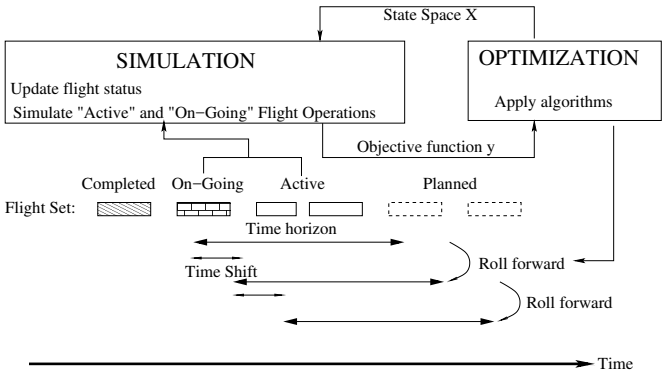


Fig. 6. Overall optimization process summary

Fig. 6 illustrates how sliding window approach works. Each aircraft is classified into four different status: *completed*, *on-going*, *active* and *planned*, based on its operation time interval relative to the sliding window. Completed means that the aircraft has already finished its operations, before the start of the current sliding window. On-going means that a part of the flight trajectory is still in the sliding window, therefore it may impact the assignment of the following aircraft. We can change the decision variables of active aircraft to optimize the operations. Planned flights will be considered in the next sliding windows.

At each step, we take into account the active and on-going aircraft in the sliding window interval to be optimized. 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. Then, the optimization window recedes in the future by a fixed time step. The status of aircraft are updated, a new set of flights waiting to be addressed are considered, and the optimization process is repeated. Detailed description can be found in [7].

Remind that the anticipation time for AMAN and for DMAN-SMAN are different, our sliding window approach are applied with two different time horizons, *i.e.*, two hours of time window with a time shift of thirty minutes at the macroscopic level, and one hour of time window with a time shift of fifteen minutes at the microscopic level. In future work, more study is required on the choices of window length and time shift.

B. Simulated annealing

Simulated Annealing (SA) [13] 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 when T decreases. Therefore, at the final stages of the annealing process, the system will converge to a near-global or global optimum. In our problem, after several empirical tests, some user-defined parameters related to the SA method are listed in Table IV.

TABLE IV
EMPIRICALLY-SET PARAMETER VALUES OF SA

Parameter	Value
Geometrical temperature reduction coefficient	0.99
Number of iterations at each temperature step	100
Initial rate of accepting degrading solutions	0.2
Final temperature	0.0001*T_0

To generate a neighborhood solution, instead of simply choosing randomly a flight in the active-flight set, we note for each aircraft the number of conflicts or 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 at the macroscopic level, or taxiway node-link conflicts and runway holding point conflicts at the microscopic level. It is better to first change the decisions of aircraft which are mostly involved in conflicts than the ones with less impact in order to reduce conflicts. 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 involving many conflicts, or experiencing severe congestions, will be chosen.

The SA terminates the execution either if the maximum number of transitions and the minimum temperature are achieved, or if an acceptable or optimal solution is obtained.

Next, we apply the simulated annealing algorithm combined with time decomposition approach to resolve the integrated optimization problem of arrival, departure, and surface operations.

IV. RESULTS

In this section we present some test problems and analyze the associated results. We test our methodology on a one-day real data case at Paris CDG Airport. Numerical results are presented and discussed. The overall process is run on a 2.50 GHz core i7 CPU, under Linux operating system PC based on a Java code.

A heavy day of traffic on July 11th, 2017 is selected. We select only the flights that have all information necessary for simulation. Thus, we obtained the flight set consisting of 1435 flights, including 719 departures and 716 arrivals. We have a total of 342 Heavy (24%) and 1093 Medium (76%) aircraft.

A. Macroscopic level results

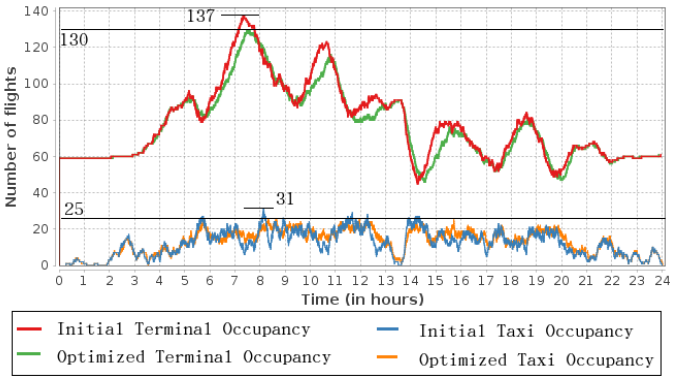


Fig. 7. Comparison of initial occupancy and optimized occupancy for terminal and taxi network

In Fig. 7, red line represents the initial gate occupancy during the day, while blue curve shows the initial taxi network occupancy. The maximum gate occupancy and the maximum taxi network occupancy are 137 and 31 respectively. In order to test the performance of overload mitigation, we set the maximum capacity to be 130 for terminal, and 25 for taxi network. The optimization results show an improvement for mitigating congestions. Both the maximum gate occupancy and the maximum taxi network occupancy are smaller than their capacity limits without any overload compared to the initial situation. The proper adjustment of time decision variables can mitigate both single peak hour (for terminal) and multiple peak hours (for taxi network). The optimized terminal occupancy curve is shifted to the right compared to initial one because the main strategy is to delay arrivals.

Remind that at the macroscopic level, we allow the possibility of runway assignments. In order to investigate how runway assignments can impact flight delays and conflict resolution,

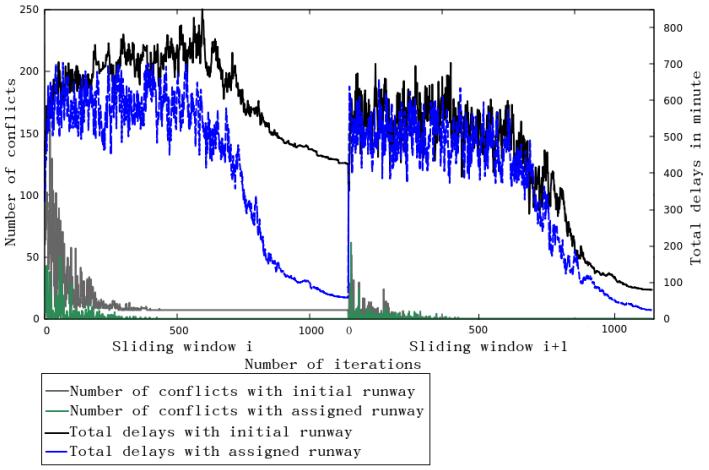


Fig. 8. Evolution of the two criteria for different runway decisions in two sliding windows

we compare the case of using initial landing and departure runway to the case of assigning runway. Fig. 8 gives an example of two sliding windows optimization evolution; it shows the value of two criteria (number of conflict and total delays) during the cooling process of SA. With both landing and departure runway assignments, we achieve a less conflict and less delay solution. When they both reach conflict-free solution as illustrated in the second sliding window, runway assignment case still has less delays. The converging speed of the two criteria is also faster.

TABLE V
COMPARISON OF LANDING RUNWAY THROUGHPUT. 26L AND 27R ARE TWO LANDING RUNWAYS

Period	**Runway throughput from radar data**		**Runway throughput from optimized results**	
	26L	27R	26L	27R
06:00-07:00	32 (48%)	34 (52%)	28 (50%)	28 (50%)
07:00-08:00	16 (53%)	14 (47%)	20 (45%)	24 (55%)
08:00-09:00	25 (60%)	17 (40%)	19 (56%)	15 (44%)
09:00-10:00	**31 (62%)**	**19 (38%)**	**22 (54%)**	**19 (46%)**
10:00-11:00	20 (67%)	10 (33%)	19 (49%)	20 (51%)

Landing runway throughput of 26L and 27R in the morning period is illustrated in Table V. The radar data shows that between 6 am and 7 am, both landing runways are in high traffic demand, while between 7 am and 8 am, runway throughput is largely decreased. Also, an imbalance of landing runway throughput can be observed from 8 am to 11 am. After optimization, the landing flow is more balanced with regard to both the time period and the two runways. Moreover, the traffic shifted to the next time period is also related to the congestion mitigation as shown in Fig. 7. Next, we extract the period between 9 am and 10 am to analyze the results at the microscopic level.

B. Microscopic level results

In order to check the impact of new assigned runway on ground performances, we make a comparison test: First, we launch the optimization process using the initial landing and takeoff runways and the initial start-up time (landing time for arrivals and pushback time for departures) obtained from radar data as input. This case is denoted as "Initial Case". Then, we launch the process using the results from the macroscopic level with assigned runway and optimized start-up time. This case is denoted as "Assigned Case".

TABLE VI
TOTAL HOLDING TIME OF ARRIVAL CROSSINGS (IN SECONDS)

Landing runway	27R	26L	Total
Initial Case	210	485	695
Assigned Case	425	125	550

TABLE VII
TOTAL HOLDING TIME AND PUSHBACK DELAY COMPARISON FOR DEPARTURES (IN SECONDS)

Departure runway	27L	26R	Total
Total holding time in Initial Case	645	645	1290
Total holding time in Assigned Case	570	75	645
Total pushback delay in Initial Case	695	3050	3745
Total pushback delay in Assigned Case	1330	225	1555

Table VI depicts the total holding time of arrival crossings for landing runway 27R and 26L. One can observe a decrease by 21% of waiting time at runway threshold after landing from 695 seconds in Initial Case to 550 seconds in Assigned Case. For departures, in Table VII, a decrease of total holding time from 645 seconds to 75 seconds and a decrease of total pushback delay from 3050 seconds to 225 seconds are achieved for runway 26R. Because in Initial Case, we have 31 landing flights from runway 26L waiting to cross 26L, thus causes a heavy waiting time for departures, while in Assigned Case, we have 22 arrivals. The reduction of delay is of practical significance, because in CDG, runways 26L/26R are usually more charged than another side. Even when we have an increase of total pushback delay from 695 seconds to 1330 seconds for runway 27L, the total pushback delay for both departure runways decreases by 58% compared to Initial Case.

V. CONCLUSIONS

This paper addresses the integrated optimization problem of arrival, departure, and surface operations. A two-level approach is proposed: First, at the macroscopic level, a TMA route graph for arrival and departure is used for conflict detection and resolution in the airspace. The airside is modeled as an abstraction network: terminal, taxi network, and runway are seen as specific resources with a defined maximum capacity. Secondly, at the microscopic level, a detailed description of airport with runway holding points, taxiways, and gates is considered. The system gets the optimized flight information from the macroscopic level, and decides ground control parameters. This two-level problem is solved by decomposing into smaller sub-problems using two time sliding windows.

Results show that at the macroscopic level, proper adjustment of time decisions would efficiently mitigate airport congestion. Runway assignment in peak hour can reduce flight delays. At the microscopic level, we also observe a significant decrease of pushback delay and waiting time at runway threshold after runway assignment.

Future research will include uncertainty analysis with regard to arrival time, pushback time, taxi time etc. More scenarios need to be tested and analyzed with both levels. Moreover, an extension to several coordinated airports will be studied in order to minimize the overall congestion over this set of airports.

ACKNOWLEDGMENTS

This work has been partially supported by CAUC, by China Scholarship Council (CSC) and by the Scientific Research Foundation of CAUC through grant number 2017QD02S. We would like to thank Serge Roux for his assistance with data, technical support and helpful discussions. We would like to thank SNA-RP/CDG-LB for providing the traffic data.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] M. J. Frankovich, "Air traffic flow management at airports: A unified optimization approach," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.

[6] N. Durand, D. Gianazza, J.-M. Alliot, and J.-B. Gotteland, *Metaheuristics for Air Traffic Management*. John Wiley & Sons, 2016, vol. 2.

[7] 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.

[8] ——, "Integrated optimization of terminal manoeuvring area and airport," in *6th SESAR Innovation Days (2016).*, 2016, pp. ISSN–0770.

[9] Z. K. Chua, M. Cousy, F. Andre, and M. Causse, "Simulating air traffic control ground operations: Preliminary results from project modern taxiing," in *4th SESAR Innovation Days 2014*, 2014.

[10] 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.

[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] 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.

[13] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.