

A Study of Tradeoffs in Airport Coordinated Surface Operations

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

▶ **To cite this version:**

Ji Ma, Daniel Delahaye, Mohammed Sbihi, Paolo Maria Scala, M Mujica Mota. A Study of Tradeoffs in Airport Coordinated Surface Operations. EIWAC 2017, 6th ENRI international workshop on ATM/CNS, Nov 2017, Tokyo, Japan. hal-01637959

HAL Id: hal-01637959

https://enac.hal.science/hal-01637959

Submitted on 18 Nov 2017

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.

ENRI Int. Workshop on ATM/CNS. Tokyo, Japan (EIWAC2017)

[EN-A-017] A Study of Tradeoffs in Airport Coordinated Surface Operations

+ J. Ma [*, **] D. Delahaye [*] M. Sbihi [*] P. Scala [***] M. Mujica Mota [***]

[*] ENAC – Université de Toulouse
Toulouse, France
[ji.ma | daniel.delahaye | mohammed.sbihi]@enac.fr

[**] Sino-European Institute of Aviation Engineering
Civil Aviation University of China
Tianjin, China

[***] Aviation Academy
Amsterdam University of Applied Sciences
Amsterdam, The Netherlands
[p.m.scala | m.mujica.mota]@hva.nl

Abstract Airports represent the major bottleneck in the air traffic management system with increasing traffic density. Enhanced levels of automation and coordination of surface operations are imperative to reduce congestion and to improve efficiency. This paper addresses the problem of comparing different control strategies on the airport surface to investigate their impacts and benefits. We propose an optimization approach to solve in a unified manner the coordinated surface operations problem on network models of an actual hub airport. Controlled pushback time, taxi reroutes and controlled holding time (waiting time at runway threshold for departures and time spent in runway crossing queues for arrivals) are considered as decisions to optimize the ground movement problem. Three major aspects are discussed: 1) benefits of incorporating taxi reroutes on the airport performance metrics; 2) priority of arrivals and departures in runway crossings; 3) tradeoffs between controlled pushback and controlled holding time for departures. A preliminary study case is conducted in a model based on operations of Paris Charles De-Gaulle airport under the most frequently used configuration. Airport is modeled using a node-link network structure. Alternate taxi routes are constructed based on surface surveillance records with respect to current procedural factors. A representative peak-hour traffic scenario is generated using historical data. The effectiveness of the proposed optimization methods is investigated.

Keywords Airport surface operations, Global optimization, Taxiways routing, Runways scheduling

1 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. 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 all 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 are critical to alleviate traffic congestions.

Airport operations involve ground movement [3], runway sequencing and scheduling [4], gate assignment [5], etc. Segregated researches on these domains have been conducted in the past years and have been proven to improve safety and efficiency. Recently, integrated study of these sub-problems are in trend since they are intimately linked and affected by one another. Holistic optimization can gain potential benefits and target a better synchronization. More and more, large-scale complex hub airports during peak hours are studied instead of limited toy example. Many works based on deterministic and stochastic optimization approaches were proposed. Mixed Integer Linear Programming (MILP) formulation is a deterministic approach usually used in this problem: In [6], controlled pushback and taxi reroutes concepts are considered to minimize the total taxi time. [7] addresses the integration of runway sequencing problem and ground routing problem including conflicts of the ramp area. As for stochastic approach, Genetic Algorithm (GA) has exclusively used. Gotteland *et al.* [8] presented a hybrid algorithm combining GA and branch and bound to solve ground movement prob-

lem. Deau *et al.* [9] extended their previous work by considering runway sequencing problem and making it more consistent with ground movement. Recently, Weiszer *et al.* [10] introduced a multi-objective genetic algorithm to solve ground operation and departure runway scheduling problem. In general, combining the airside and ground 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 problem of comparing different control strategies on the airport surface to investigate their impacts and benefits. An global optimization approach is used to solve in a unified manner the coordinated surface operations problem on network models of an actual hub airport. Controlled pushback time, taxi reroutes and controlled holding time are considered as main decisions to optimize the ground movement problem and the runway scheduling problem. Our first contribution is to analyze real alternate taxi routes used on a complex airport configuration. Then an integrated global model taking into account pushback, taxi routes, holding point, holding time, runway crossing and departure sequence at the same time is proposed. Several levers give the possibility to compare different scenarios.

The reminder of this paper is structured as follows. Section 2 describes the problem and analyzes the taxi routes using radar traffic data. Section 3 models the airport ground operation problem. Section 4 presents the solution approach. Section 5 performs tests and analyzes the results. Section 6 gives some conclusions and perspectives.

2 PROBLEM DESCRIPTION

Airport ground optimization aims at finding the best schedules and routes in order to minimize delays and maximize airfield capacity taking into account several constraints: taxiing separation, route choices, wake turbulence separation for landings and take-offs etc. A departure flight starts its taxi process with pushback at gate. Then it follows the assigned taxi route to reach the departure runway entry point and takes off. An arrival flight lands and exits the runway, then taxis to the assigned gate. Based on different configurations of airport, an arrival may cross a departure runway, and *vice versa*. In the following section, we give first an introduction about our airport model. Then we explain how to construct alternate taxi routes based on surface surveillance records with respect to current procedural factors. These alternate taxi routes will be used later in the optimization model.

2.1 CDG airport model

We choose to study Paris CDG airport because of its complexity and accessibility to the data. CDG airport is one of the busiest passenger airports in Europe, composed of four parallel runways (two for landings and two for departures) and three terminals. In CDG, Ground controllers handle all intermediary taxiing routes. Local controllers and Apron controllers handle respectively the runway area and parking areas [11]. 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. 1, 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. Ramp area is beyond the scope of this paper. For arrivals, taxiing path starts with runway exit point, and ends in meta-gate. We model CDG airport with a graph $G = (N, L)$, where N and L represent the nodes set and links set respectively. 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.

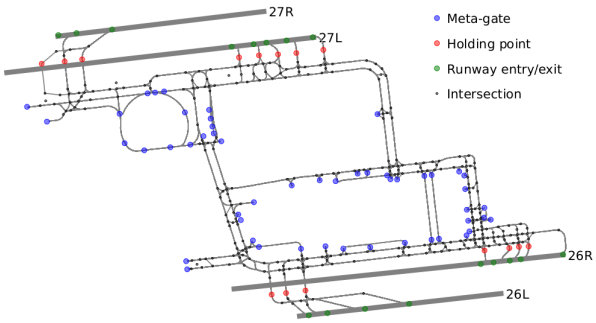
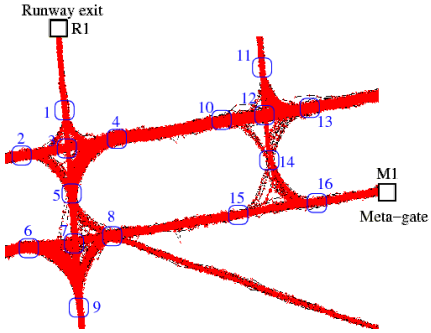


Figure 1 CDG airport model in the west configuration

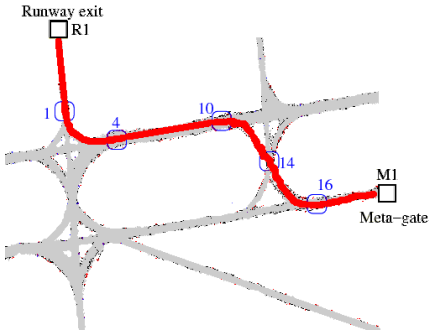
2.2 Ground traffic data preprocessing

In previous literature, three possible routing options are most used in the aircraft taxi problem: single path, alternate path and free path [7]. In the first case, aircraft follow a predetermined taxi route, which is usually the standard route in the airport. In the second case, several routing options are proposed after applying, for instance, the k-shortest path algorithm [12]. In the last case, any routes can be assigned to an aircraft.

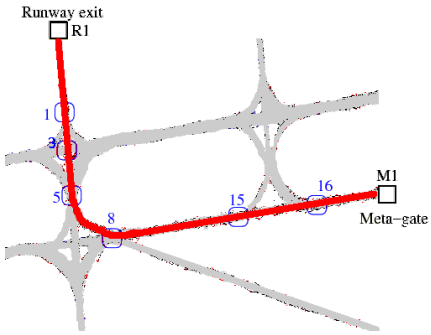
In the operational point of view, most of the aircraft take standard taxi route with a preferential sens specified in the operations manual. Sometimes controllers deviate the predefined taxi route of one aircraft in order to avoid potential conflicts between flights or to forbid blocked or restricted areas. The alternate route choices obey some potential rules (e.g., not taking a long and unnecessary detour). Considering the taxiway configuration of Roissy airport, alternate path seems to be an appropriate option to formulate the problem in order to comply with reality.



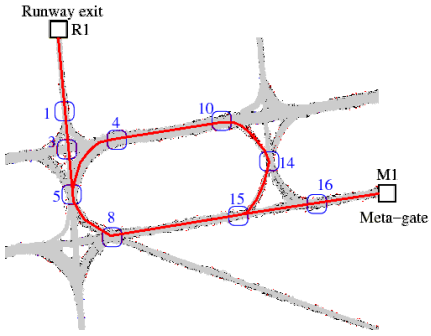
(a) Define intersections on radar track



(b) Route option 1 from $R1$ to $M1$



(c) Route option 2 from $R1$ to $M1$



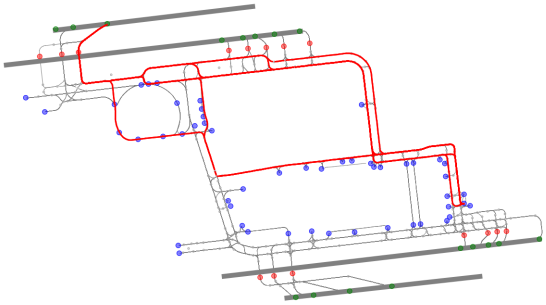
(d) Route option 3 from $R1$ to $M1$

Figure 2 Route generation based on radar tracks and intersections

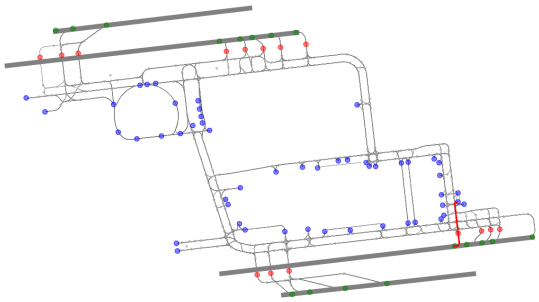
In the previous related work, alternate paths are generated by applying a classical k-shortest path algorithm in the graph network of airport [8]. Cost represents the travel time on one link, possibly augmented to avoid crossing some runway areas. Additional adjustments are applied to avoid passing two times the same node. Thereafter, feasible route options can not be simply defined as a set of validated shortest paths, because the distances from one to another are too small, which is algorithmically correct but not applicable in practice. Distinct alternate routes need to be found. Moreover, the value k should not be the same for each pair of origin and destination. A gate in the north side which is close to a north side runway will have less route options than a gate in the south side which is far from this runway. Based on these practical requirements, we decided to extract alternate routes sets by analyzing airport's flight radar records to find the operationally used potential routes set. To the best of our knowledge, this is the first time that alternate taxi routes are generated with radar data.

First, we define all the possible intersections of the taxi network, each with a circle centered on the intersection and with a radius to cover all the radar tracks. Remark that at the runway entry/exit point, aircraft moves faster, thereby our record radius must be large enough to capture all the aircraft plots. Thereafter, for each radar track, we record its route as a series of nodes, starting with runway exit point, ending with a meta-gate for arrivals and starting with a meta-gate, ending with runway entry point for departures. Fig. 2 shows an simplified example. We have one runway exit $R1$, one meat-gate $M1$ and 16 intersection nodes presented in Fig. 2a, the radar tracks are illustrated in red lines. The possible route options are { $R1$, 1, 4, 10, 14, 16, $M1$ } (Fig. 2b) with 10 aircraft passing, { $R1$, 1, 3, 5, 8, 15, 16, $M1$ } (Fig. 2c) with 5 aircraft passing and { $R1$, 1, 3, 5, 8, 15, 14, 10, 4, 5, 8, 15, 16, $M1$ } (Fig. 2d) with 1 aircraft passing with regard to the positions and time of radar tracks. Then we sort all the possible routes in a descending order by the number of flights going through the set of nodes. It has not escaped our notice that the route traversed by only one aircraft is usually in an abnormal case, see example on Fig. 2d. Therefore, it should not be selected in the potential route set. At last, for pair ($R1$, $M1$), we obtain two alternate routes: { $R1$, 1, 4, 10, 14, 16, $M1$ } and { $R1$, 1, 3, 5, 8, 15, 16, $M1$ }. To summarize, in order to generate alternate taxi routes, we proceed:

- **Step 1: Generate routes of flights**
  - For each flight, find the nodes for which the radar track passes inside the detection zone;
  - Obtain the route by chronologically sorting these nodes;
  - Count the number of flights using the same route, sorted in descending order.



(a) 9 possible taxi-in route options



(b) One unique taxi-out route

Figure 3 Taxiing route set example

- **Step 2: Generate route set for each pair of origin and destination**
  - Collect pairs with same origin and destination, put these pairs in one routes set with the associated origin and destination;
  - For each pair, delete routes used by only one aircraft if another option exists.

After analyzing 13 days of real traffic (February 2016), we generate all the feasible taxiing route sets for the west configuration in CDG. Fig. 3a illustrates an example of 9 possible route options from one north runway exit to a south meta-gate. In Fig. 3b, we found only one route option from the meta-gate to the runway entry, which can be explained by the short distance between origin and destination. However, in total 309 aircraft follow this route.

Table 1 Route options count

Number of route options k	1	2–5	6–9
Number of pairs displaying k options	342	159	9

We have in total 510 combinations of different pairs (runway meta-gate). In most cases we have only one standard route. Besides, other options exist between 2 and 9 routes. Few pairs possess more than 6 options. Table 1 lists the number of pairs admitting k routes option ($k = 1, ..., 9$).

3 MATHEMATICAL FORMULATION OF THE PROBLEM

In this section, we describe an integrated global optimization model for the airport ground operations problem. We first give flights input data. Next, decision variables are defined. Then, we clarify constraints. At last, an objective function is introduced.

3.1 Input data

We have a set of flights $F = A \bigcup D$, where A denotes the set of arrival flights and D denotes the set of departure flights. For each $f \in F$, the following input data are given:

- C_f: wake turbulence category;
- M_f: meta-gate;
- E_f: runway entry point for departure or runway exit point for arrival;
- P_f^0: initial off-block time for departure;
- L_f: initial landing time for arrival;
- H_f: initial holding point at runway threshold;
- R_f: a set of alternate routes knowing the origin and the destination of f.

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. For each link we use the average speed value analyzed with the real data to take into account different taxiway types (e.g., taxiways near parking areas and near runways have significantly different speeds, as do straight taxiway segment and turning segment);
- Ramp area is beyond the scope of this work, instead we use the notion of meta-gate.

3.2 Decision variables

In order to optimize the ground movement, we now consider several potential control points as decisions. For each flight $f \in F$, the decision variables are defined as follows:

- $r_f \in R_f$: taxi-in or taxi-out route;
- t_f^h: holding time (waiting time at runway threshold for departures and time spent in runway crossing queues for arrivals);

- p_f: pushback time;
- h_f: holding point for arrival. CDG south-side runway layout shown in Fig. 4 motivates us to use arrival holding point as decision variable. In reality, simultaneous flight crossings can enhance departure runway throughput.



Figure 4 CDG south side runway layout

Furthermore, the following auxiliary variables are introduced:

- $t_f^{E_f}$: final take-off time for departure or runway crossing time for arrival at E_f. It is calculated based on the route chosen and the associated taxi speed;
- t_f: completion time for flight f: $t_f = t_f^{E_f}$ for departures, t_f is equal to in-block time for arrivals.

These decision variables are discretized considering that in practice, discretized time slots are assigned for the flights:

- $t_f^h \in \{0, \Delta t, 2.\Delta t, ..., N_h.\Delta t\}$, where Δt is a time slot, N_h is the maximum allowed number of holding time slots, $T_h = N_h \times \Delta t$ is the maximum holding time. T_h depends on the type of movement (arrival or departure). We define T_h^a and T_h^d as maximum holding time for arrival and for departure respectively.
- $p_f \in \{P_f^0, P_f^0 + \Delta t, P_f^0 + 2.\Delta t, ..., P_f^0 + N_p.\Delta t\}$, N_p is the maximum allowed number of pushback delay time slots, $T_p = N_p \times \Delta t$ is the maximum pushback delay.

3.3 Constraints

Airport operational constraints are taken into account:

- Minimum taxi separation of $s = 60$ meters [9] between two taxiing aircraft.
- Take-off runway wake turbulence separations shown in Table 2.
- Holding point capacity (the maximum number of lights 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.

Table 2 Single-runway separation requirements, s_{fg}, in seconds.

Category		Leading Aircraft, f		
		Heavy	Medium	Light
Trailing Aircraft, g	Heavy	90	60	60
	Medium	120	60	60
	Light	120	60	60

Based on the route network structure in Fig. 1, and in order to express the previous mentioned separation standards and capacity constraints, given an instantiation of decision variables, we define:

- C_n - the total number of conflicts on nodes. For each node n, we record all the flights f passing this node with the time t_f^n, and sort according to t_f^n. For two successive aircraft f and g passing the node n, we must make sure that $t_g^n - t_f^n > t_s$, where t_s is the minimum time separation calculated based on s and the taxi speed on node n. Otherwise we increase C_n by 1.
- C_l - the total number of conflicts on links. For each given link l, we record all the flights passing this link with the time at the link entry and exit. Then we sort into two lists and compare, if the orders of two aircraft are swapped, then a link conflict is detected and C_l is increased by the rank difference between entry and exit. Besides, if two aircraft use the same link but come from opposite direction, the exit time of the previous aircraft must be earlier than the entry time of the latter one, otherwise C_l is increased by 1. The node-link conflict detection methodology is similar with previous work [13]. Moreover, we add the bi-directional link conflict detection in this work.
- C_r - the total number of conflicts on runways. For each departure runway r and for two successive take-off flights f and g, we have $t_g^r - t_f^r \geq s_{fg}$, where t_f^r and t_g^r are take-off times for flight f and g respectively, otherwise C_r is increased by 1.

- C_h - the total number of conflicts on holding points. For each holding point, we first make sure that the sequence of waiting flights remains the same. If not, C_h is increased by the rank difference. 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 C_h by the exceeded capacity.
- $C = C_n + C_l + C_r + C_h$ - the total number of conflicts

Then the previous separation and capacity constraint is transformed to $C = 0$ (conflict-free).

3.4 Objectives

Remark that one of our objectives in this paper is to investigate the impact of taxi reroutes on airport performance. One of the main roles of taxi reroutes is to avoid aircraft conflicts. Therefore, we decide to relax the conflict-free constraint and put C in our objective function.

The **objective function** that we want to minimize is:

$$C+\alpha \sum_{f\in D}(p_f-P_f^0)+\beta \sum_{f\in F} h_f+\gamma \left[\sum_{f\in D}(t_f - p_f) + \sum_{f\in A}(t_f - L_f)\right],$$

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;
- $\sum_{f\in D}(t_f - p_f)$: Total taxi time for departures;
- $\sum_{f\in A}(t_f - L_f)$: Total taxi time for arrivals.

and α, β and γ are weighting coefficients corresponding to pushback delays, holding time and taxi time respectively.

4 SOLUTIONS APPROACHES

The benefits of integrated airport optimization, such as runway scheduling, taxiway routing and gate assignment are promising. However, 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 [3]. Due to these high combinatorics, we propose a meta-heuristic algorithm – simulated annealing – to address the problem.

Simulated Annealing (SA) is a meta-heuristic that simulates the annealing of a metal, in which the metal is heated up and slowly cooled 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 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 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.

Table 3 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	$0.0001*T_0$

In order to adapt SA to the airport ground optimization problem, several parameters need to be defined: initial temperature and initial acceptance probabilities, cooling schedule, neighborhood function, equilibrium state and termination criterion. For our problem, some parameter values are listed in Table 3. Moreover, to generate a neighborhood solution, instead of simply choosing randomly a flight f, we proceed it by two steps: first resolve conflicts, then minimize time changes. In Algorithm 1 for conflicts resolution, for each aircraft, we use the number of conflicts as its *performance indicator*. For arrivals, the performance involves runway crossing conflicts and ground conflicts, denoted as crossingPerfo and groundPerfo respectively in Algorithm 1. For departures, we record their take-off conflicts (denoted as takeoffPerfo) and ground conflicts. The algorithm targets one flight involved in conflicts and changes its decision with regard to its sub-performances. For example, if a departure has ground conflict with another aircraft, it is clearly useless to change its holding time at the runway threshold in order to solve this type of conflict. Instead, the pushback time or the taxi route should

be changed to generate new neighborhood solution. The fact that our neighborhood definition is based on the total number of conflicts, intensifies the neighborhood generation and accelerates conflicts resolution. Once a conflict-free solution is reached, we change our strategy, to target aircraft with time decision changes (pushback delay or holding time) and try to decrease this value by using the neighborhood function in Algorithm 2.

Algorithm 1 Neighborhood function for resolving conflicts

Require: For each flight, we record its takeoffPerfo, crossingPerfo and groundPerfo, the sum is denoted as totalPerfo;
P_c = crossingPerf/totalPerfo;
P_t = takeoffPerfo/totalPerfo;
P_g = groundPerfo/totalPerfo;

Choose one flight f involved in conflicts based on its performance;
Generate random number, ν = random(0,1);
if $f \in A$ **then**
    if $\nu \leq P_c$ **then** choose with equal probability between holding point and holding time change;
    else choose with equal probability among taxi-in route, holding point and holding time change;
    end if
else if $f \in D$ **then**
    if $\nu \leq P_g$ **then** choose with equal probability between pushback time change and taxi-out route change;
    else choose with equal probability among holding time, pushback time and taxi-out route change
    end if
end if

Algorithm 2 Neighborhood function for minimizing time changes

Choose one flight f with time decision changes;
if $f \in A$ **then** choose a new holding time between 0 and current one;
else if $f \in D$ **then**
    if pushback time changed **then** choose a new pushback time between 0 and current one;
    else if holding time changed **then** choose a new holding time between 0 and current one;
    end if
end if

The SA terminates the execution either if the maximum number of transitions and the minimum temperature are achieved, or if an acceptable solution is obtained (for example, in the conflict-resolution case, SA stops when $C = 0$).

5 RESULTS

We test our methodology on a one-hour real data case at Paris CDG Airport. Numerical results with different settings of (user-defined) algorithm parameters are presented and discussed. The overall process is run on a 2.50 GHz core i7 CPU, under Linux operating system PC based on a Java code.

5.1 Real data analysis

We extracted one-hour dense traffic data from 9:00 a.m. to 10:00 a.m. on February 18, 2016. West configuration is activated over the course of the day. During this peak hour, we observed a long departure queue at runway 26R with radar tracks. A total of 31 arrivals and 69 departures were operated, including 65 Medium and 35 Heavy aircraft. Landing runway 26L and takeoff runway 26R in the south side are more charged with 22 arrivals and 38 departures respectively.

Three major aspects concerning airport ground performances are discussed in this section:

- Benefits of incorporating taxi reroutes on the airport performance metrics;
- Priority of arrivals and departures in runway crossing;
- Tradeoffs between controlled pushback and controlled holding time for departures.

5.2 Taxi reroute

In order to investigate the impact of taxi reroute on ground conflict resolution, we first set our objectives to be only C here. Remind that C includes runway conflicts, ground conflicts (link, node, bidirectional link) and holding conflicts. 30 random tests are launched and results are depicted in Table 4. In the case of "Without taxi reroute", we use standard route for each pair of runway and gate. At the end of algorithm running, we reached conflict-free solution for all the tests in taxi reroute case, while without taxi reroute, we have 2 times unsolved conflicts. Considering the average CPU time, taxi reroutes test is more than twice as fast compared to another case. Therefore, taxi reroutes can help reduce ground conflicts and reach a conflict-free solution faster.

Next, in order to test the taxi reroute influence on flight delays, we reset our objective function to

$$C+\alpha \sum_{f\in D}(p_f-P_f^0)+\beta \sum_{f\in F} h_f+\gamma \left[\sum_{f\in D}(t_f - p_f) + \sum_{f\in A}(t_f - L_f)\right]$$

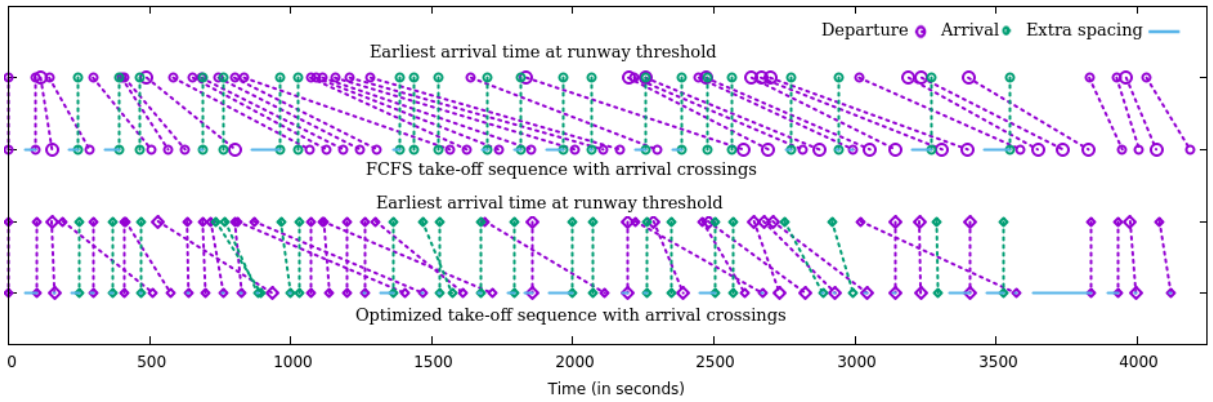


Figure 5 Take-off time comparison between FCFS strategy and optimized case for runway 26R

Table 4 CPU comparison for T_p = 10 min, T_h^d = 10 min, T_h^a = 3 min

Decision Choice	With Taxi Reroute	Without Taxi Reroute
Average CPU (in s)	11	26
Min CPU (in s)	4	4
Max CPU (in s)	25	112
Failed number	0	2

with $\alpha = \beta = \gamma = 0.0001$. Small values of weighting coefficients are chosen in order to ensure that conflict-resolution is the first priority. After reaching a conflict-free solution, the algorithm focuses on minimizing the delay. Even though we use relaxation on conflict constraints, by putting relatively small coefficients on other objectives, all the following results are conflict-free. Combining different objectives into one using a weighted sum approach is a common technique for multi-objective optimization problem. In future research, Pareto ranking-based algorithm [14] can be a good option to target this problem.

After running the algorithm, we observed that for landing runways 27R and 26L, the aircraft average holding times are similar in both situations. However, for take-off runways, a decrease of average holding time for runway 26R from 72 s to 63 s (12 %) and a decrease of average pushback delay for runway 27L from 62 s to 43 s (30 %) is reached after applying the taxi reroute strategy. The result is reasonable because in the case of Without Taxi Reroute, once a ground conflict is detected, the algorithm may modify the pushback time to solve conflicts, therefore more modifications are made compared to taxi reroute case.

5.3 Runway holding

To build a First-Come-First-Serve (FCFS) sequence, we allow a maximum pushback delay T_p = 1 min with taxi reroute. In this way we obtain a ground conflict-free solution with initial time at runway threshold for both departures and arrivals. Thereafter, an arrival is obliged to cross the departure runway immediately as it reaches the holding point. Departures have to wait and take off in a FCFS order. In consequence, we build our FCFS sequence as shown at the top of Fig. 5.

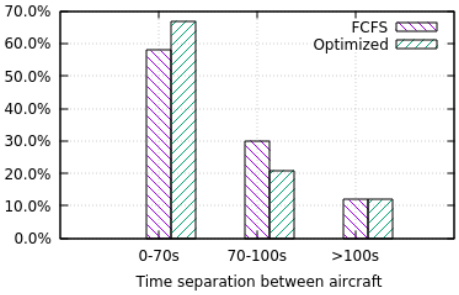
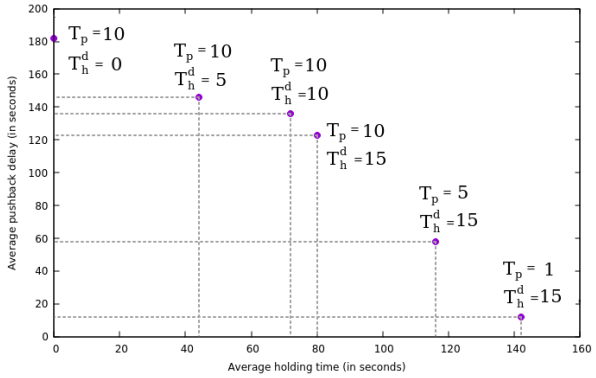


Figure 6 Inter-arrival and inter-departure spacings for south-side runways

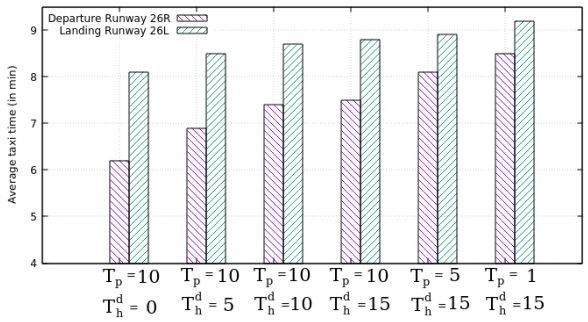
Table 5 Average holding time comparison for south-side runways

Runway	FCFS average holding time	Optimized average holding time
26R	7.1 min	2.6 min
26L	0	0.5 min

To make a comparison, we still set T_p = 1 min with taxi reroutes. Moreover, we choose T_h^d = 10



(a) Trade-offs between pushback time and holding time for runway 26R



(b) Average taxi time

Figure 7 Airport performances

min, $T_h^a = 3$ min to initialize our optimization settings. Fig. 5 illustrates the holding time comparison of runway 26R in the south side. We can observe that much less time deviations are made in the optimized case compared to FCFS strategy. This is mainly thanks to the proper adjustment of arrival holding times. When two arrivals cross almost simultaneously the departure runway after a heavy take-off, it is beneficial to reduce runway waiting time and to increase runway throughput. Controllers give the same order facing high traffic demand in real cases. Blue lines represent the extra spacing between two consecutive aircraft. Even in the end of the second row a large extra spacing appears. This is due to the limit of aircraft earliest arrival time at the runway threshold.

Arrivals do have priority over departures. However, small adjustment of arrival holding time would reduce significantly departure holding time in high traffic demand period. Table 5 show that average departure holding time is reduced from 7 min to 2.6 min at the cost of increasing 0.5 min average arrival holding time. Moreover, the inter-departure spacing is reduced as well. Fig. 6 illustrates spacing reduction of the optimized case compared to the FCFS strategy.

5.4 Trade-offs between pushback time and holding time

We made several tests to see the trade-offs between pushback time and holding time for departures as shown in Fig. 7a. In case $T_p = 1$ min, $T_h^d = 15$ min, aircraft are scheduled to start pushback as soon as possible, causing a large waiting time at the runway threshold with a high runway pressure. When we consider another extreme case $T_p = 10$ min, $T_h^d = 0$, that is aircraft take off smoothly without any departure queue, the delay is transferred to the pushback at gate with engine-off. After testing several combinations of T_p and T_h^d, one can observe that the departure runway waiting time can not be reduced without the cost of pushback delay. The average taxi time shown in Fig. 7b is impacted proportionally by the holding time as well. However, with a proper choice of T_p and T_h^d, a compromise can be found to balance the two criteria. Therefore, controllers can choose the maximum holding time and maximum pushback delay as parameters according to their preference and current traffic demand.

In conclusion, our study shows that taxi reroutes can help reach a conflict-free solution faster and reduce delay for departures. The slight adjustment of arrival holding time can significantly reduce departure runway congestions in peak hour. The departure runway waiting time can not be reduced without the cost of pushback delay. However, the maximum holding time and maximum pushback delay can be used as decision support parameters for controllers to manage departure runway pressure.

6 CONCLUSIONS

This paper addresses the problem of comparing different control strategies (controlled pushback time, taxi reroutes, controlled holding time) on the airport surface to investigate their impacts and benefits. We propose an optimization approach to solve in a unified manner the ground movement problem and runway scheduling problem. A preliminary study case is conducted in a model based on operations of Paris CDG airport under the most frequently used configuration. Alternate taxi routes are constructed based on surface surveillance records with respect to current procedural factors. The preliminary analysis estimates that taxi reroutes can reduce ground conflicts and reach a conflict-free solution faster. The slight adjustment of arrival holding time can significantly reduce departure runway congestions during peak hour. The departure runway waiting time cannot be reduced without the cost of pushback delay. Future research will include uncertainty analysis considering pushback time, taxiing time and ramp area. In addition, multi-objective optimization of ground operations can be considered in a more holistic manner. More high-traffic demand scenarios need to be created for evaluation (e.g., evaluate one day traffic by time decomposition approach).

Integration of airside and airport movements can be investigated as well.

ACKNOWLEDGMENTS

This work has been partially supported by Civil Aviation University of China and by China Scholarship Council (CSC). The authors would like to thank Serge Roux for his assistance with data and technical support. The authors also would like to thank Jean-Baptiste Gotteland and Mathieu Cousy for their helpful discussions.

References

[1] Eurocontrol, "European atm master plan," 2015.

[2] F. A. A. (FAA), "Nextgen implementation plan 2016," 2016.

[3] 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*, pp. 131–138, 2010.

[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] A. Bouras, M. A. Ghaleb, U. S. Suryahatmaja, and A. M. Salem, "The airport gate assignment problem: a survey," *The Scientific World Journal*, vol. 2014, 2014.

[6] H. Balakrishnan and Y. Jung, "A framework for coordinated surface operations planning at dallas-fort worth international airport," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, p. 6553, 2007.

[7] J. Guépet, *Optimization of airport operations: stand allocation, ground routing and runway sequencing*. PhD thesis, Université Grenoble Alpes, 2015.

[8] J.-B. Gotteland, *Optimisation du trafic au sol sur les grands aéroports*. PhD thesis, Université d'Angers, 2004.

[9] 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.

[10] M. Weiszer, J. Chen, and G. Locatelli, "An integrated optimisation approach to airport ground operations to foster sustainability in the aviation sector," *Applied Energy*, vol. 157, pp. 567–582, 2015.

[11] 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.

[12] E. Q. Martins and M. M. Pascoal, "A new implementation of yens ranking loopless paths algorithm," *4OR: A Quarterly Journal of Operations Research*, vol. 1, no. 2, pp. 121–133, 2003.

[13] 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.

[14] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: Amosa," *IEEE transactions on evolutionary computation*, vol. 12, no. 3, pp. 269–283, 2008.