Taxi planner optimization: a management tool

Á Marín[1*] and **J Salmerón**[2]
[1]Departamento Matemática Aplicada y Estadística, Universidad Politécnica de Madrid, E. T. S. Ingenieros Aeronáuticos, Madrid, Spain
[2]Department of Operations Research, Naval Postgraduate School, Monterey, California, USA

The manuscript was received on 24 January 2008 and was accepted after revision for publication on 10 July 2008.

DOI: 10.1243/09544100JAERO331

Abstract: This work introduces taxi planning optimization (TPO) as a methodology to guide airport surface management operations. The optimization model represents competing aircraft using limited ground resources. TPO improves aircraft taxiing routes and their schedule in situations of congestion, minimizing overall taxiing time (TT), and helping taxi planners to meet prespecified goals such as compliance with take-off windows, TT limits, and trajectory conflicts. By considering all simultaneous trajectories during a given planning horizon, TPO's estimation of TT from the stand to the runways improves over current planning methods. The operational optimization model is a large-scale space-time multi-commodity network with capacity constraints. In addition to its natural use as a real-time taxi planning tool, a number of TPO variants can be used for design purposes, such as expansion of new infrastructure. TPO is demonstrated using Madrid-Barajas as test airport.

Keywords: taxi planning, airport management, airport design

1 INTRODUCTION

1.1 Airport operations management

Arrival management (AM), departure management (DM), and gate management (GM) are considered as the primary management tasks in the operation of an airport. AM plans the arrival sequence for landing aircraft in a given time horizon. DM gives the push-back orders to departing aircraft at the stands and establishes 'calculated' take-off time windows (CTOTs). GM assigns stands to arriving aircraft. The success of these tasks is closely related to the efficient operation of the airport taxiways, which is commonly known as the 'taxi planning' problem. This work introduces taxi planning optimization (TPO) as a methodology to guide surface management operations.

Aircraft taxiing congestion between stands and runways represents a major challenge for airport architects, aircraft schedule planners, and real-time taxiing operators. Congestion is typically caused by an inadequate ground infrastructure at the airport to meet the needs for flight movement. Major hubs suffer aircraft delays on the ground which are sometimes aggravated by low visibility conditions caused by rain, fog, or other contingencies. Aircraft taxiing operations along with departure and AM are also critical due to security reasons. Being related to ground congestion or not, taxiing errors by pilots or controllers have also been the cause of fatal aircraft accidents such as the collision of Pan Am 1736 with KLM 4805 in Tenerife, Spain [**1**] in 1977, or the August 2006 crash of Flight 5191 [**2**], which took off from the wrong runway at Lexington airport, KY.

1.2 Requirements for a TPO system

In the context of airport operations management, TPO is not intended to operate as a stand-alone tool. In contrast, it must be coordinated with DM tools (for departing traffic), with AM and GM tools (for arriving traffic) and, of course, with the actual taxi planner of the airport, known as the aircraft ground controller (AGC). Currently, each activity is modelled separately and then coordinated with all other activities. Updated

Corresponding author: Departamento Matemática Aplicada y Estadística, Universidad Politécnica de Madrid, E. T. S. Ingenieros Aeronáuticos, Plaza Cardenal Cisneros, 3 Madrid 28040, Spain. email: angel.marin@upm.es

information, such as a delay in the embarkation process, arises frequently during daily airport operations, especially for outbound traffic. Thus, TPO must be flexible in order to accommodate changing inputs, while being consistent regarding routes and schedules already delivered from past executions. In this dynamic context, an AGC's requirement is that TPO's execution time does not exceed a few minutes. Any flight which may use the taxiways within the incumbent planning horizon must be considered in that TPO run. Figure 1 depicts this idea: assume the TPO horizon starts at the time mark labelled as 'current', and consider ten possible flights: five departures (labelled Dn) and five arrivals (labelled An), for $n = 1, \ldots, 5$. Note that flights D1 and A1 have already arrived at the runway and the stand, respectively, so they are disregarded. Flights D2 and A2 left the stand and runway, respectively, before the beginning of the planning horizon, but they are still taxiing, so must be included in the incumbent TPO run. Their current location and immediate direction must be provided to the TPO by the AGC. Flights D3 and D4 have departure times (provided by the DM) within the planning horizon. Similarly, flights A3 and A4 have landing times (provided by the AM) within this period. Thus, these four flights must be considered by the incumbent TPO run, regardless whether or not they arrive at the runway (or stand). Finally, departure for flight D5 and landing for flight A5 will occur after the incumbent planning horizon has ended, so they must be ignored at this iteration.

1.3 Optimization in the air industry

The use of optimization for air transportation problems is predominant in the areas of flight and crew scheduling [**3**], in DM or AM using job scheduling [**4**], and in GM, e.g. to minimize the variance of idle times at the gates [**5**]. TPO is rarely used, instead, simulation models have been proposed [**6**], but they lack the ability to pre-emptively optimize the aircraft routes and schedules, which is the ultimate goal of this study.

Duran and Gotteland [**7**] employs a pattern recognition model based on genetic algorithms to characterize taxiing conflicts. Anagnostakis *et al.* [**8**] presents several formulations for DM, providing a description of runway operations planning. The authors present only initial thoughts on how to solve the problem in a dynamic context. Andersson *et al.* [**9**] proposes two queuing models to capture the 'taxi-out' and 'taxi-in' processes. These topics are also studied by Idris *et al.* [**10**], which estimates the taxi-out time (difference between scheduled gate departure time and take-off time) in terms of factors such as runway and terminal configurations, downstream restrictions and take-off queues. Anagnostakis [**11**] presents a thorough study of the structure and properties of the runway operations planning problem and develops a decomposition-based algorithm to solve it. Stoica [**12**] proposes an adaptative approach for the management of available aircraft routes at the airport.

Marín and Salmerón [**13**] and Marín [**14**] develop the first TPO formulation as a binary multi-commodity network flow model. They solve it by branch and bound and fix and relax, but they are not fast enough for real-time operations. Marín [**15**] develops a computationally efficient Lagrangian decomposition (LD) approach and applies it to the Madrid-Barajas airport (MBA). Marín and Codina [**16**] extends the use of TP to model the airport design, which can be used to support decisions regarding airport configurations.

1.4 Aim and organization

As stated in section 1.1, this work introduces TPO as a tool to guide real-time, operational decisions. TPO (developed as part of the European Commission project 'LEONARDO') improves aircraft taxiing routing and scheduling in situations of congestion, helping

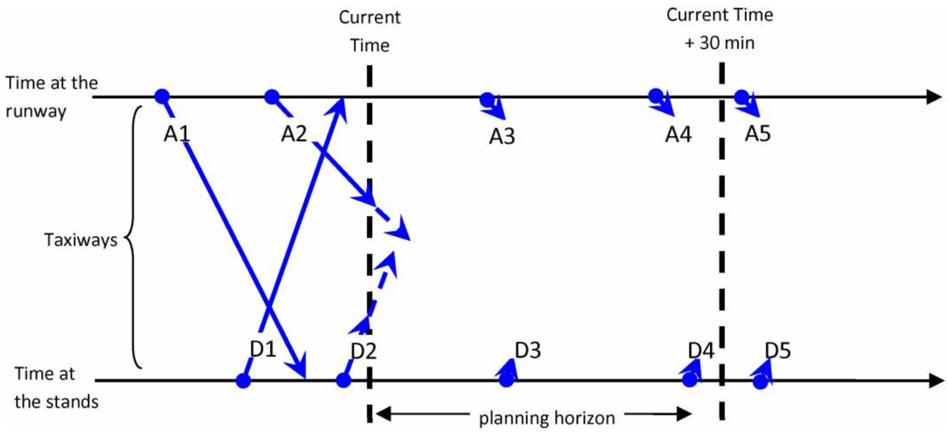

Fig. 1 Flights involved in a 30 min TPO run

taxi planners to reduce taxiing time (TT), meet take-off windows and avoid trajectory and runway conflicts. TPO improves current planning methods which, for example, may employ precalculated tables to estimate TT from the stand to the runways. These tables ignore the location and intended movement of all other aircraft using the airport taxiways. TPO improves current planning methods used by surface manager [**17**].

The remainder of the paper is organized as follows: in section 2, landing, take-off and GM tools which interact with taxi planning are briefly described. Section 3 introduces the mathematical foundation of the TP optimization model. Section 4 describes the computational experience using data from the MBA. Section 5 covers extensions of the basic model to help with airport design. Conclusions are presented in section 6.

2 AIRPORT TRAFFIC MANAGEMENT

This section describes the landing, take-off, and GM tools, and their interaction with the TPO tool.

2.1 Overview: TPO timeline

From here on, 'route' is used to refer to a time-phased route, i.e. a combined physical route and schedule for a given aircraft. TPO uses data from the GM, AM, DM, and AGC modules, and feeds them with updated routes, in a continuous iterative process. The version of TPO developed in this study is efficient with planning horizons of 15–30 min, where updates can be introduced approximately every 1 or 2 min. An example of utilization of the TPO tool presented in this paper would be as follows.

6.00 a.m. – TPO is executed after having received updated input data from:

DM: expected departure (push-back) time, departure runway and desired take-off window, for every departing flight within the next 30 min; AM: arrival runway, expected landing time and exit gate, for every arriving flight within the next 30 min; GM: stand and desired gate arrival window, for every arriving flight within the next 30 min; AGC: location of every aircraft currently en-route (on the taxiways).

6.02 a.m. – TPO execution has finished. The DM, AM, GM, and AGC systems receive updated information based on the new optimized route schedules.

6.05 a.m. – TPO receives the most updated input data from DM, AM, GM, and AGC for the next 30 min (until 6.35 a.m.), and is executed.

6.07 a.m. – TPO execution has finished. DM, AM, GM, and AGC receive new updated route schedules (and the process continues).

2.2 Landing traffic management

Each arriving aircraft requires permission from the runway controller to use a landing runway. The AM estimates the landing time and runway exit gate a few minutes before the aircraft touches ground. This information is communicated to TPO in anticipation of its interaction with the rest of the taxiing operations. Once the aircraft has arrived, its control is handed over to the AGC. If the estimations by the AM are accurate, the AGC will use the route provided by the TPO tool. Otherwise, current arrival parameters will be updated in TPO and a new route for the aircraft must be generated. In any case, the final route should be consistent with ongoing taxiing operations of all other aircraft.

2.3 Take-off traffic management

Departing aircraft's permission to push-back (i.e. leave the stand) is given by the DM. Using the estimated departure times (or actual push-back times) the TPO tool calculates the so-called 'initial take-off time' (ITOT) and provides it to the DM. The route is communicated to the AGC, which takes control of the aircraft routing until it reaches the runway, at which time control is handed over to the DM. In order to calculate the ITOT, the TPO model optimizes the route between the stand and the runway. This includes the warm-up and push-back periods, the outbound taxiing period and the runaway occupancy period.

2.4 Gate management

The terms 'gate' and 'stand' are used interchangeably to designate the parking position used for servicing a single aircraft. The problem of assigning arriving flights to available gates depends on the scheduled flights, their realized schedules, the aircraft requirements, and the capacities of ramp facilities. Gate operations may have a significant impact on the efficiency of taxiing operations.

2.5 Integration of airport management modules

The study of dependencies among runway assignments, landing and take-off traffic sequences, gate assignments, and taxiing routes is important in order to ensure efficient Terminal Manoeuvring Area Management (TMAM). Currently, these operations are integrated 'parametrically' as several modules, where each module attempts to solve its own problem after receiving inputs from other modules, and provides outputs for other modules.

Obviously, from a global viewpoint, this TMAM operation is heuristic, that is, suboptimal because: (a) not all modules operate optimally with respect to their individual objectives, and (b) even if they did, they are still myopic with respect to the interaction their decisions may have in other modules. TPO may help planners to integrate better decisions within current TMAM procedures by improving step (a) for the taxi planning module. A more sophisticated and computationally challenging approach would address (b) by optimizing all TMAM activities simultaneously (i.e. where the interplays between different modules are specified through decision variables, not 'parameters'); however, that is not the focus of this paper.

3 TAXI PLANNER AS AN OPTIMIZATION MODEL

At the core of the TPO tool there is a mathematical formulation which can be described as a multicommodity time-phased network flow model [**18**, p. 737]. This mathematical representation is referred to as the TPO model (TPOM). The TPOM represents the transit of aircraft over time across the airport facilities. The model accounts for features such as capacities on taxiways and at holding points, aircraft length and priority, and congestion in taxiways, and runways. In addition to the physical and logistic constraints imposed on any feasible solution, the optimal routing and scheduling is driven by the objective function: 'to minimize the total TT (including the waiting time in queues due to congestion) plus subjective penalties for take-off delays with respect to prespecified CTOTs'. The CTOT term in the objective gives the planner the flexibility to establish tradeoffs between these two goals. Near-zero penalties indicate indifference in meeting CTOTs as long as overall TT is minimized. Higher CTOT penalties can be used to indicate specific aircraft for which meeting their CTOT is essential.

The complete formulation is not reproduced here. However, in the interest of exposition, a summary of some key aspects of the model, such as the main decision variables and their interplay in select constraints is presented in this section (for the basic model) and in section 5 (for the extended model). A list of notation is included as an appendix for reference.

The basic model can be succinctly described as follows.

1. A directed network $G = (N, A)$ lays the foundation of the TPOM. Here, N represents the set of airport 'nodes', and A the set of 'arcs'. Each node $i \in N$ is either a parking area, a holding area, a junction or intersection of two or more taxiways, or a runway header or exit gate. An arc $(i, j) \in A$ connecting nodes i and j, typically represents a physical taxiway, but the model also uses them to represent entrance- or exit-ways, into or from a stand, respectively, and arcs are used to model intermediate segments of a taxiway where an aircraft may need to stop.
2. The network is replicated over time by considering an ordered set T, where $t \in T$ represents a time period (slot) or, more exactly, the time mark at which the period begins. A 30 min planning horizon with 30 s periods is usually employed. Accordingly, $t = 1$ would be the initial time mark, and $t = 60$ would start 30 s before the end of the planning horizon.
3. Finally, a set of flights $w \in W$ within the planning horizon is considered. W is divided into arriving traffic and departing traffic. Each aircraft is defined by an origin node $o(w)$, a destination node $d(w)$, and either a departure time (given by DM) or a landing time (given by AM), denoted $t(w)$. Assuming that all the trajectories are completed in the planning period, $d(w)$ represents a specific stand for each arriving flight, and a fictitious 'airborne' node for departing flights. Similarly, $o(w)$ is either a stand (for departing flights) or an exit node on the runway (for arriving flights).

With these preliminaries and appropriate data at the above levels (such as travel time for each arc, and departure time and CTOT for departure flights, among others) decisions are represented by the following binary variables:

(a) $E_{it}^{w} = 1$ if flight w is waiting at node i in time period t, and 0 otherwise;
(b) $X_{ijt}^{w} = 1$ if flight w leaves from node i to node j in time period t, and 0 otherwise.

It is important to note that it suffices to know where each aircraft is located at the initial time period, and the value of all of E_{it}^{w} and X_{ijt}^{w}, to completely specify all aircraft routes. The challenge is, of course, to establish a tractable (but consistent) set of mathematical relationships among these variables and the problem data, that lead to an optimal solution on the decision variables.

Two examples of these relationships are as follows.

1. Taxiing time calculation (denoted TT_w, for each aircraft $w \in W$)

$$\mathrm{TT}_w = \sum_{t \geqslant t(w)} \left(1 - E_{\mathrm{d}(w),t}^{w}\right) + \sum_{i \in N} r_i^w E_{i,|T|}^{w} \tag{1}$$

The above equation computes the number of time slots, which translates into taxiing duration (after multiplying by 30 s per slot). The first term on the right-hand side provides the number of time periods before the aircraft arrives to its destination. The second term accounts for aircraft which may not arrive at the destination before the end of the planning period (where $|T|$ denotes the total number of periods). An estimated (precalculated) time, r_i^w, is applied, depending on the intermediate node i where the aircraft is located at the end of the planning period. The total TT is given by

$$\mathrm{TT} = \sum_{w} \mathrm{TT}_w$$

2. Balance equations

$$E_{it}^{w} + \sum_{j \in R(i)} X_{(j,i),t-t_{ji}+1}^{w} = E_{i,t+1}^{w} + \sum_{j \in F(i)} X_{(i,j),t+1}^{w}, \quad \forall w, i, t \geqslant t(w) \qquad (2)$$

Consider t_{ji} as the transit time from node j to an adjacent node i. The above equations establish that, in order for aircraft w to either: (a) wait at node i in period $t+1$, or (b) start going from i to j in period $t+1$, it is required that either w is waiting at node i in period t, or w had left from adjacent node j to node i in period $t - t_{ji} + 1$. $F(i)$ represents the set of arcs with origin node i and $R(i)$ represents the set of arcs with destination in node i(usually known as forward and reverse stars of node i, respectively).

TPOM is defined by the previous objective function and constraints along with other constraints.

1. Objective function: minimize a weighed sum of total TT and penalties for failing to meet CTOTs.
2. Initial conditions: aircraft origin and time at origin.
3. Final conditions: runway capacity and CTOT for departing flights, and stand for arriving flights. These are elastic conditions, because some aircraft will be en-route at the end of the planning period.
4. Balance equations: ensure consistency in aircraft route and schedule.
5. Overtaking: avoid an aircraft passing another aircraft while on the same taxiway.
6. Capacity conditions: enforce limits at holding areas and taxiways.
7. Runways: ensure runway is not used by two aircraft simultaneously, and avoid conflicts if more than one runway header exists.
8. Stands: avoid the arrival of landing traffic to a stand and/or stand exit area if it is still occupied by a departing flight.
9. Other logical conditions and/or objectives, e.g. to encourage persistence on routes for aircraft already on their way at the beginning of the planning period.

The TPOM formulation solves to near optimality in approximately 1 min (using standard personal computers), which is a realistic requirement for the reasons stated in the introduction.

4 CASE STUDY

4.1 Madrid-Barajas airport

All TPOM tests presented in this paper are from the MBA as of year 2005, using actual flight planning data supplied by the Spanish Air Navigation Agency [**19**]. MBA has been experiencing annual increases in traffic of 17.5, 3.5, −0.5, 5.4, 7.9, and 9 per cent during the years 1999–2005, respectively. The greatest congestion problems arise during the summer when the monthly traffic is more than four million passengers.

MBA comprises eight terminals, 166 stands, an arrival runway with four exits, a departure runway with two take-off positions, 56 main junctions (nodes), and roughly 100 main taxiways (arcs). Stands and nearby areas represent the bulk of nodes and arcs which total almost 1000. Figure 2 depicts the notional network for MBA's terminals. Nodes S1, . . . , S5 are the exit nodes of the landing runway. Nodes NEP1, NEP2, and NEP3 are the access nodes of the take-off runway. Terminal areas are denoted ER0, ER1A, . . . , ER7. Nodes NA1, . . . , NA13 correspond to junction nodes on the arrival taxiway and NM1, . . . , NM7, NF, . . . , NL correspond to junction nodes on the departure taxiways. Nodes NY1, NY2, and NY3 are also junction nodes between the main taxiways and the take-off runway.

4.2 Computational results

TPO for MBA is a complex mathematical problem. For example, the network model described in the previous section applied to a scenario with 30 flights and a planning period of 30 min (divided into 60 periods of 30 s each), leads to a TPOM with over 600 000 binary decision variables and over a million constraints. The balance constraints (2) alone represent nearly 200 000 constraints.

TPOM can be solved, in theory, as a mixed-integer optimization problem by using Branch and Bound [**20**, p. 355]. However, due to the model size, this approach is impractical for operational use. Alternative algorithms, such as fix-and-relax [**21**], and LD [**22**] have been employed. Marín [**14**, **15**] describes the specifics on the use of these methods for the TPO problem.

The best computational results, which are reported in this section, have been achieved using LD. Another advantage of the LD approach is that the optimization subproblems derived from the decomposition have special structures for which specialized algorithms exist, so they can be solved efficiently without relying on commercial optimization software.

Steps are being taken to reveal the computational efficiency of TPO as a tool for operational purposes. 'The newly developed taxi planning tool and the gate/airline simulation tools have been verified on LEONARDO CDM (collaborative decision making)

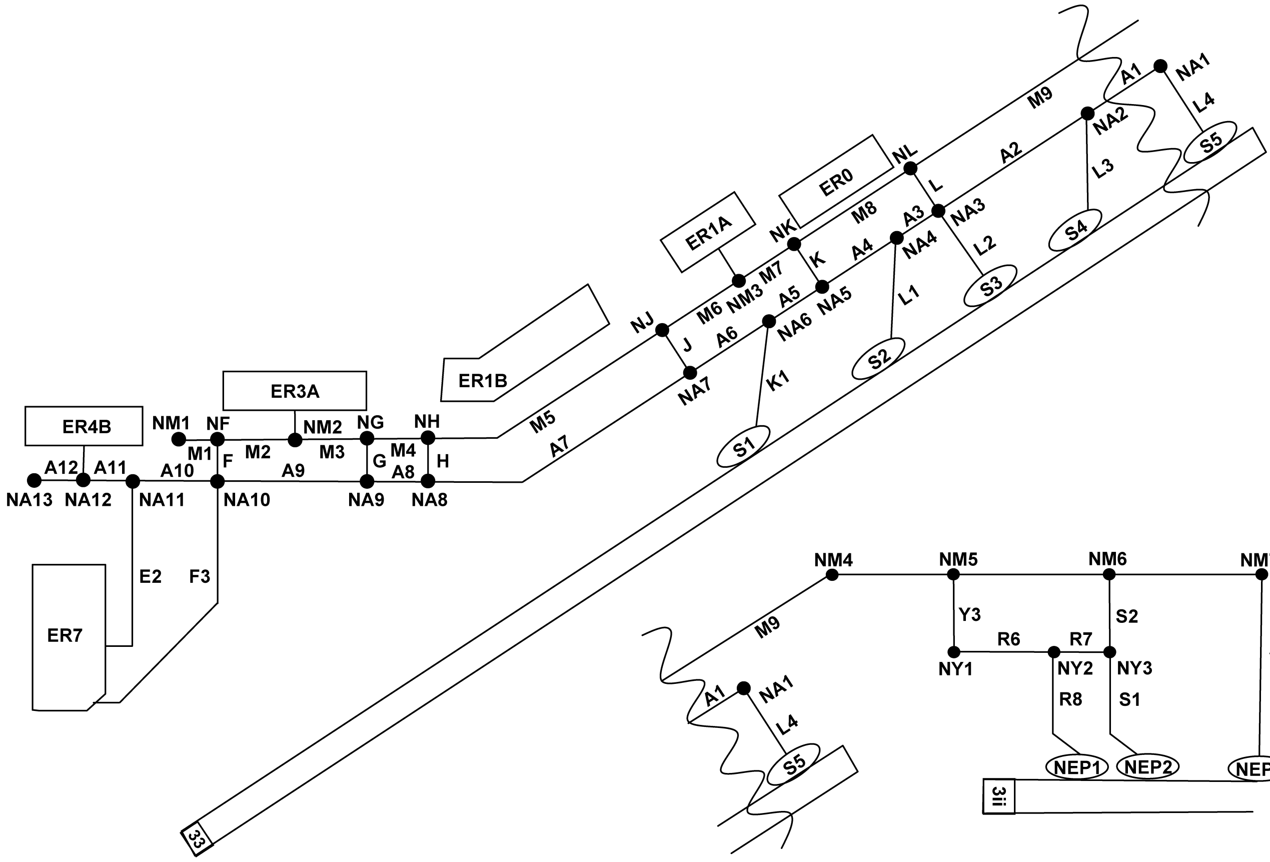

Fig. 2 MBA: TPO network depiction, with landing runway (33) and take-off runway (3ii) (not to scale)

requirement on sufficient high performance before the start of the validation period' [**23**, p. 53].

The TPOM with LD has been implemented in C++ on a Mitac laptop with an AMD-Athon 64-bit processor at 800 MHz and 1 GB of RAM, running under Windows XP 64 bit. Results are summarized in Table 1, where the number of flights considered depends on the planning horizon. The test cases include departing flights in the morning (9.00–11.30) and both arrival and departure flights in the afternoon (12.30–16.00). Overall, there is an increase in the time to solve the problems relative to the total number of flights.

Figure 3 compares the average taxiing time (ATT = $TT/|W|$) for each of the above 13 scenarios. In 11 scenarios, ATT is under 25 min, and in the most complex cases where $|W|$ is 20–32 flights, ATT is approximately 15 min.

Table 1 Computational results

Planning horizon	Arriving flights	Departing flights	CPU time (s)	Objective value
9.00–9.30	0	4	0.4	85
9.30–10.00	0	12	0.8	263
10.00–10.30	0	18	18.1	450
10.30–11.00	0	15	15.1	411
11.00–11.30	0	6	4.0	184
11.30–12.30	Data not available		N/A	
12.30–13.00	16	4	0.7	228
13.00–13.30	18	13	11.1	473
13.30–14.00	14	16	37.0	518
14.00–14.30	0	3	0.3	59
14.30–15.00	15	6	0.9	257
15.00–15.30	12	13	7.3	397
15.30–16.00	16	16	26.0	528

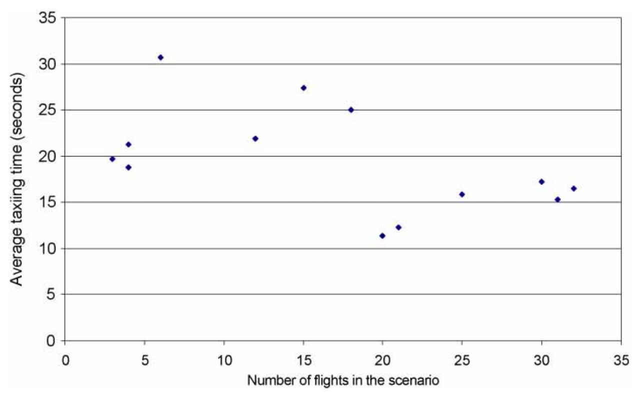

Fig. 3 Comparison of average TTs for the 13 scenarios in Table 1

A customized preprocessing typically reduces the size of the problem by one order of magnitude. Essentially, the preprocess downsizes the original network to a working subnetwork associated with the incumbent flights. Stands and nearby areas not to be used by the incumbent aircraft (those involved in the current planning horizon) are eliminated, among other refinements. As an example, in the 13.30–14.00 case from Table 1, which consists of 14 landing flights and 16 take-off flights, the total number of nodes and arcs (including all stand-related subnetworks) is reduced to 152 and 201, respectively. A 60-period instantiation of this problem can then be solved by LD in less than 2 min, compared to almost 30 min without preprocessing.

4.3 Graphical analysis of the solution

As an example of output provided by the TPO tool, Table 2 shows time-space schedule details for one arriving flight (AZA060) and one departing flight (IBE0548) during the 14.30–15.00 planning period. AZA060, which lands at 14:48:00 (period 38) and uses runway exit S2, arrives at its designated parking at 14:54:00 (period 48). IBE0548, leaves the gate at 14:44:30 (period 29) and initiates its take-off manoeuvre at 14:55:30 (period 51).

Figure 4 represents the TPO trajectories for landing and departing flights in the 14.30–15.00 slot. Marks indicate the aircraft is expected to pass through the node at the specified time. The waiting time for some flights is easily represented as horizontal lines in the time-space routes. 'ERC' symbolizes the arrival at a generic parking platform. (Arrivals occur at different platforms, but are aggregated in the graphical representation for clarity.) 'NP'indicates final parking for arrivals, if reached within the planning period, and 'AN' is a fictitious 'airborne' node for departing flights (remark: flight ANS8542 has been omitted to simplify the graph).

5 TPO EXTENSIONS FOR AIRPORT DESIGN

5.1 Additional objectives

While the main thrust to develop TPO is in order to guide real-time decision making, TPO may be useful to assess the effect of short-term operational decisions,

Table 2 Schedule for two select flights: arriving flight AZA060 and departing flight IBE0548

Period	AZA060	IBE0548	Period	AZA060	IBE0548
28	Not landed	Parked	42	NA7 → NA8	NL-M4_2 → NL-M4_3
29	Not landed	T13 → ER3D	43	NA8 → NA9	NL-M4_3 → NM4
30	Not landed	ER3D → ER3D_NG_1	44	NA9 → NA9-10_1	NM4 → NM4-5_1
31	Not landed	ER3D_NG_1	45	NA9-10_1 → NA10	NM4-5_1 → NM5
32	Not landed	ER3D_NG_1 → ER3D_NG_2	46	NA10 → NF	NM5 → NM5-Y1_1
33	Not landed	ER3D_NG_2 → NG	47	NF → ER3C	NM5-Y1_1 → NY1
34	Not landed	NG → NH	48	ER3C → T8	NY1 → NY2
35	Not landed	NH → NH_NJ_1	49	Parked	NY2 → NY3
36	Not landed	NH_NJ_1 → NJ	50	Parked	NY3 → NEP2 (entering runway)
37	Not landed	NJ → NM3	51	Parked	Take-off manoeuvre
38	S2 → NA4	NM3 → NK	52	Parked	Flight has taken off
39	NA4 → NA5	NK → NL	53	Parked	Flight has taken off
40	NA5 → NA6	NL → NL-M4_1	54	Parked	Flight has taken off
41	NA6 → NA7	NL-M4_1 → NL-M4_2			

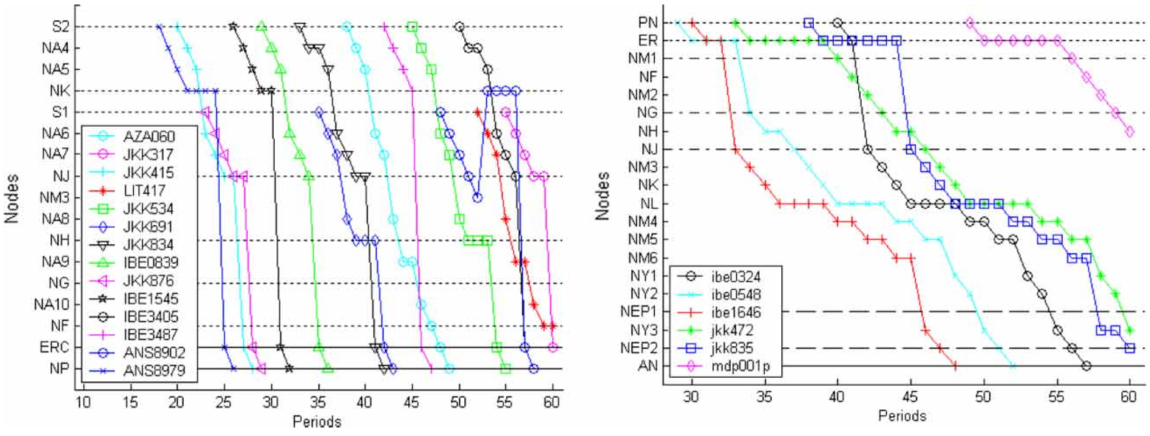

Fig. 4 Landing (left) and departing (right) flight trajectories in the 14.30–15.00 scenario

such as temporarily opening, closing or reversing a runway. In addition, TPO can also be extended to help architects with long-term infrastructure planning, such as the location of a new terminal. All of these extensions are referred to as taxi planning network design (TPND), which was formally introduced in reference [**16**].

As an example, prior to its recent opening, it was anticipated that the 2006 expansion of the MBA with two pairs of parallel runways would add significant variability to TTs: 'since the possible combinations of stand and runway greatly increase, the dispersion of the TTs will also increase. If we want to benefit from the new MBA, the realistic estimation of TTs will become a major issue' [**24**].

In addition to objective of TT, TPO, and TPND may incorporate one or several of the following objectives 1 to 4.

1. Conflict prevention: a conflict arises when two or more aircraft have crossing trajectories within a short time window (e.g. 1 min) from each other. An anticipated conflict requires a close, step-by-step, controller-guided intervention until resolution is ensured. Thus, provisions can be made to limit the number of interventions. Let n be the maximum number of interventions controllers can handle in any given period, γ_{kt} a binary decision variable that takes a value of 1 when two or more aircraft approach node k around period t, and K a subset of nodes where conflicts need to be limited. The following constraint is added to the TPO model

$$\sum_{k\in K}\gamma_{kt} \leqslant n, \forall t \tag{3}$$

Note that an extreme value of $n = 0$ would force all aircraft trajectories to never conflict with each other (thus, longer TTs may occur in order to ensure additional separation).

A term reflecting total controller interventions, I_{C}, is added to the objective function, in order to minimize it (see examples in section 5). I_{C} is defined as

$$I_{\mathrm{C}} = \sum_{t\in T}\sum_{k\in K}\gamma_{kt} \tag{4}$$

Naturally, γ relates to the original $E$and X variables through linear constraints; details [**16**] are omitted for brevity.

2. Worst routing time (WTT), calculated by means of the following constraints

$$\mathrm{WTT} \geqslant \mathrm{TT}_w, \quad \forall w \in W \tag{5}$$

Alternatively, a prespecified maximum time $\mathrm{TT}_w^{\max}$ could be enforced for each flight w

$$\mathrm{TT}_w \leqslant \mathrm{TT}_w^{\max}, \quad \forall w \in W \tag{6}$$

3. Waiting delays for arrival and/or departure traffic, D_{IN} and D_{OUT}, respectively, calculated as

$$D_{\mathrm{IN}} = \sum_{w\in W^{\mathrm{A}}}\sum_{t\in T}\sum_{i\in N^{\mathrm{W}}} E_{i,t}^{w} \quad \text{and}$$
$$D_{\mathrm{OUT}} = \sum_{w\in W^{\mathrm{D}}}\sum_{t\in T}\sum_{i\in N^{\mathrm{W}}} E_{i,t}^{w} \tag{7}$$

where W^{A} and W^{D} are the subsets of arriving and departing flights, respectively, during the planning period, and N^{W} is the set of holding points and other waiting nodes.

4. Number of arrivals at the gate and/or take-offs Γ^+ and Γ^-, respectively, given by

$$\Gamma^+ = \sum_{t\in T}\sum_{w\in W^{\mathrm{A}}}\sum_{j\in N^{\mathrm{P}}}\sum_{i\in R(j)} X_{i,j,t}^{w} \quad \text{and}$$
$$\Gamma^- = \sum_{t\in T}\sum_{w\in W^{\mathrm{D}}}\sum_{i\in N^{\mathrm{AR}}}\sum_{j\in F(i)} X_{i,j,t}^{w} \tag{8}$$

where N^{P} and N^{AR} are the sets of parking and access runway nodes.

Priorities among all objectives can be established through weights, hierarchical optimization of objectives or by setting individual target levels as hard constraints. Whichever strategy is used to incorporate all of these goals simultaneously, the objective function must be calibrated. For example, TT offers opportunities to analyze tradeoffs with trajectory conflicts (I_{C}): in presence of congestion, reducing the number of conflicts may lead to an increase in TTs.

5.2 Design decision modelling

Airport design modelling concerns the airport topology (i.e. its physical infrastructure or its configuration at a given point in time). The TPO model can incorporate design variables, represented by a binary vector $\boldsymbol{Y}$, to become TPND. For example, component Y_i may take a value of one if a certain airport facility (e.g. a runway accessible through node i) is available, and zero otherwise.

Design possibilities considered in TPND comprise the following decisions.

1. Opening or closing individual facilities: nodes or links.
2. Sizing the capacity of waiting nodes to hold one or more aircraft simultaneously.
3. Enabling one or several runway headers and exit nodes.

For example, the utilization and capacity of a particular node (including runway access or exits) can be modelled as

$$\sum_{w\in W} e_w E^w_{i,t} + \sum_{w\in W}\sum_{j\in R(i)} e_w X^w_{j,i,t-t_{ji}+1} \leqslant q_i\, Y_i, \forall i \in \hat{N}, t \in T \tag{9}$$

where $\hat{N} \subset N$ is the subset of nodes whose availability needs to be decided. Clearly, if $Y_i = 0$, no aircraft is allowed to use (wait at or leave from) node i. This would indicate that the node is not needed for the given scenario. On the other hand, $Y_i = 1$ would allow a maximum capacity of q_i, where e_w is the space required by each aircraft w. Based on a sufficiently representative sample of potential scenarios, an airport designer may determine the convenience of adding or suppressing nodes.

The following example of TPND analysis comprises several runs motivated by two possible runway configurations: two separated runways used exclusively for take-off and landing, respectively (Fig. 5), and mixed runways (Fig. 6). In both cases, NW1 and NW2 are waiting areas to access the take-off runway at headway points NAR1 and NAR2, respectively. NW3 is a waiting area for both arriving and departing traffic.

Since the number of combinations for the design variable $\boldsymbol{Y}$ is small, the headway nodes NAR1 and NAR2 have been set to closed or open manually (i.e. as input data, rather than as optimization variables),

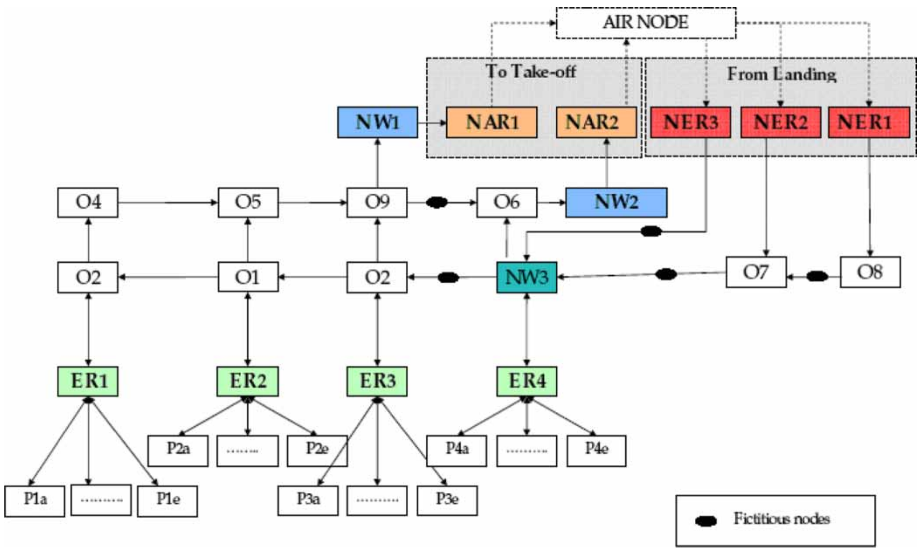

Fig. 5 Example used for aircraft network design: take-off and landing runways are separated

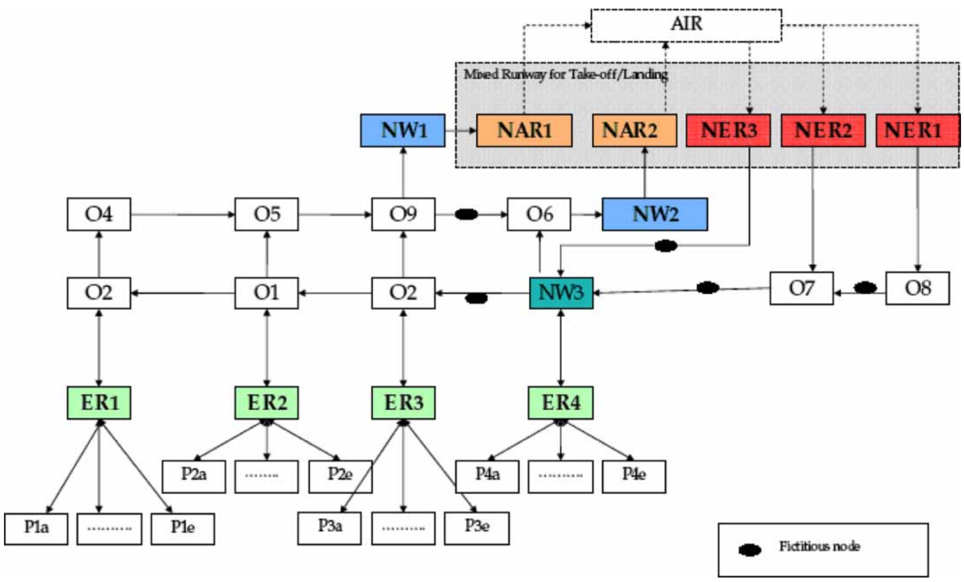

Fig. 6 Example used for aircraft network design: mixed runway

Table 3 Design runs for MBA with mixed and separate runways and unlimited-capacity terminal nodes

Run	NAR1	NAR2	Y	I_C	TT	D_{IN}	D_{OUT}	Γ^-	Γ^+	T_{CPU}
Mixed runways										
1	1	0	$(\infty, 0, 0)$	4	309	1	116	10	7	4.4
2	0	1	$(0, \infty, 0)$	15	308	1	89	10	7	20.7
Separate runways										
3	1	0	$(\infty, 0, 0)$	4	250	1	61	14	7	0.9
4	0	1	$(0, \infty, 0)$	11	267	1	58	13	7	89.0

which allow to compare the optimal headway choice to a suboptimal design. Thus, in these examples, $\boldsymbol{Y}$ will only determine whether or not waiting nodes NW1, NW2, and NW3 are necessary, and their capacity (if applicable).

In the first set of examples (Table 3) it is assumed that: only one waiting node (indicated by vector $\boldsymbol{Y}$) of unlimited capacity can be chosen and the capacity of terminal nodes ER1, ER2, ER3, and ER4 is also unlimited.

For the mixed-runway runs (one and two), total throughput is 17 (seven arrivals and ten take-offs) regardless of the available headway node. It appears that the use of NAR1 as the only headway point significantly reduces conflicts. This may be due to the fact that node O6 and waiting nodes NW2 and NW3 form a 'control zone', where resolution is strictly enforced. The use of NAR2 may reduce delays for take-off traffic. Similarly, for separate runways (runs three and four), NAR1 also decreases the number of conflicts and the routing time, and improves the number of arriving aircraft. Overall, the advantage of separated runways is clear in terms of routing time, delays and total throughput.

In the second set of examples (Table 4) it is assumed that: both NAR1 and NAR2 are allowed; waiting nodes, of different capacity, can be chosen; allocating capacity at waiting nodes incurs a cost whose weight is α_L, being $1-\alpha_L$ the weight for all the other terms (number of conflicts, routing time, delays for arriving and departing traffic, and number of departing and arriving flights). For example, for $\alpha_L = 0.5$, these weight are: $\alpha_{Ic} = 0.2, \alpha_{TT} = 0.1$, and $\alpha_{Din} = \alpha_{Dout} = \alpha_{\Gamma-} = \alpha_{\Gamma+} = 0.05$; and the capacities of terminal nodes ER1, ER2, ER3, and ER4 are given data (q).

The $\boldsymbol{Y} = (Y_1, Y_2, Y_3)$ vector reflects optimal capacity (number of aircraft) allocated at waiting nodes NW1, NW2, and NW3, respectively, assuming the airport is configured with mixed runways.

Results show the effect of prioritizing capacity versus routing costs by altering the weighting factor. When the location cost factor is increased on the objective function (runs one to three) then total capacity at the design wait nodes is reduced and a progressive degradation of the routing time is detected. A slight decrease in controller intervention is also observed, due to an increase in the delays experienced by aircraft waiting at other locations with less capacity, far from the conflicting area. Runs four and five where the capacity of the terminal nodes is increased from one to two and five, respectively, confirm the previous recommendation to locate the capacity at the wait node NW2.

Table 4 Design runs for MBA with mixed runways and capacitated terminal nodes

Run	α_L	q	Y	I_C	TT	D_{IN}	D_{OUT}	Γ^-	Γ^+	T_{CPU}
1	0.1	1	$(1, 6, 0)$	15	302	1	95	10	7	18.5
2	0.5	1	$(1, 4, 0)$	15	303	1	104	10	7	23.6
3	0.9	1	$(1, 3, 0)$	14	324	8	108	9	7	22.8
4	0.5	2	$(0, 3, 0)$	12	305	1	106	10	7	73.5
5	0.5	5	$(0, 2, 0)$	12	302	1	101	10	7	351.0

6 DISCUSSION AND CONCLUSIONS

Motivated by the existing lack of formal optimization in the process of aircraft routing and scheduling through the airport taxiways, this work has introduced TPO. The underlying model, a space-time, multi-commodity network with capacity constraints, represents conflicts among aircraft competing for limited ground resources while trying to optimize the taxi planner's goals, all of which had not been considered simultaneously in any previous work. The conceptual optimization model has been demonstrated using Madrid-Barajas as the test airport.

Optimization here distinguishes itself from current approaches which may overlook better trajectories considered by the airport's taxi planner. At their discretion, TPO allows them to establish tradeoffs among different goals such as aircraft routing time, number of anticipated trajectory conflicts, traffic delays, and total number of aircraft completing their routes. For example, a taxi planner may be interested in minimizing the worst-routing time during periods of higher congestion, and switching to a minimizing average routing time at other periods.

TPO's potential to improve overall airport operations stems from its ability to quickly find optimal decisions (within the limitations of its mathematical modelling): TPO may guide the AGC in better decision making, which is the ultimate goal. In addition, fast optimized space-time trajectories for taxiing aircraft allows its real-time integration with other airport management tools such as DM and AM and AGM. This integration would make such a system one step closer from an ideal integrated optimization framework, that is, a system where all those management tools are optimized simultaneously. TPO or other enhanced optimization models may become one day the basis of fully-automated guidance systems for airport operations. In addition to its main use as a real-time taxi planning tool, TPO provides insights into congestion levels, bottlenecks, and airport capacity utilization, which may help flight schedulers to make adequate changes to timetables.

Lastly, TPO can be used pre-emptively to estimate how infrastructure changes (such as a new runway, waiting area, parking, or taxiways) would affect taxiing operations, allowing airport architects to assess the need for such infrastructure.

Future research may seek the integration of several or all of the airport management modules into a consolidated optimization planning tool.

ACKNOWLEDGEMENTS

This research was supported by Aeropuertos Españoles y Navegación Aérea and Spain's Ministry of Education and Science through grant TRA2005-09068-C03-01. The authors also wish to thank two anonymous reviewers for their helpful comments on this paper.

REFERENCES

1 National Transportation Safety Board (NTSC), 2008, available from http://www.ntsb.gov/ntsb/brief.asp?ev_id=55834&key=0, accessed June 2008.

2 National Transportation Safety Board (NTSC), 2008, available from http://www.ntsb.gov/ntsb/brief.asp?ev_id=20060828X01244&key=1, accessed June 2008.

3 **Etschmaier, M. M.** and **Mathaisel, D. F.** Airline scheduling: an overview. *Transp. Sci.*, 1985, **19**, 127–138.

4 **Bianco, L., Dell'Olmo, P.,** and **Giordani, S.** Scheduling models and algorithms for traffic management. In *Modelling and simulation in air traffic management* (Eds L. Bianco, P. Dell'Olmo, and A. R. Odoni), 1997, pp. 139–167 (Springer-Verlag, Berlin, Heidelberg, New York).

5 **Bolat, A.** Models and a genetic algorithm for static aircraft-gate assignment problem. *J. Oper. Res. Soc.*, 2001, **52**, 1107–1120.

6 **ATAC Corporation.** Airport and airspace simulation model, 2006, available from http://www.atac.com, accessed June 2008.

7 **Duran, N.** and **Gotteland, J.** Genetic algorithms applied to air traffic management. In *Metaheuristics for hard optimization*, 2006, pp. 277–3066 (Springer, Berlin).

8 **Anagnostakis, I., Clarke, J. P., Böhme, D.,** and **Völckers, U.** Runway operations planning and control: sequences and scheduling. *AIAA J. Aircr.*, 2001, **38**(6), 988–996.

9 **Andersson, K., Carr, F., Feron, E.,** and **Hall, W. D.** Analysis and modelling of ground operations at hub airports. In the 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, 13–16 June 2000.

10 **Idris, H., Clarke, J. P., Bhuva, R.,** and **Kang, L.** Queuing model for taxi-out time estimation. *Air Traffic Control Q.*, 2002, **10**(1), 1–22.

11 **Anagnostakis, I.** *A multi-objective, decomposition-based algorithm design methodology and its application to runway operations planning*. PhD Dissertation, Massachusetts Institute of Technology, 2003.

12 **Stoica, D.** *Analyse, represèntation et optimisation de la circulation des avions surune plate-forme aèroportuaire*. PhD Thesis, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, Toulouse, France, 2004.

13 **Marín, A.** and **Salmerón, J.** Aircraft routing and scheduling on the airport ground. In the Joint Euro-Informs Conference, Istanbul, Turkey, 6–10 July 2003.

14 **Marín, A.** Airport management: taxi planning. *Ann. Oper. Res.*, 2006, **143**, 189–200.

15 **Marín, A.** Decomposition methodology to solve taxi planning. In the 11th Meeting of the Euro Working Group on Transportation, Bari, 27–29 September 2006, available from http://www.poliba.it/ewgt2006/11th%20EWGT%20Meeting.htm, accessed June 2008.

16 **Marín, A.** and **Codina E.** Network design: taxi planning. *Ann. Oper. Res.*, 2008, **157**, 135–151.

17 Leonardo. Technical verification report for Barajas. Document 41D01IND02R, Leonardo Project (European Commission), 2004, available from http://leonardo.aena.es/, accessed June 2008.

18 **Ahuja, R. K., Magnanti, T. L.,** and **Orlin, J. B.** *Network flows*, 1993 (Prentice-Hall, Inc., Upper Saddle River, New Jersey).

19 AENA (Aeropuertos Españoles y Navegación Aérea), 2006, available from http://www.aena.es/csee/Satellite?pagename=Home&lang=EN_GB, accessed June 2008.

20 **Wolsey, L. A.** and **Nemhauser, G. L.** *Integer and combinatorial optimization*, 1999 (John Wiley & Sons, Inc., New York, NY).

21 **Escudero, L.** and **Salmeron, J.** On a fix-and-relax framework for a class of project scheduling problems. *Ann. Oper. Res.*, 2005, **140**, 163–188.

22 **Guignard, M.** and **Kim, S.** Lagrangean decomposition: a model yielding stronger Lagrangean bounds. *Math. Program.*, 1987, **39**, 215–228.

23 Leonardo. D 5.1.1 Validation specification. Document 51D01SIC10R, Leonardo Project (European Commission), 2004, available from http://leonardo.aena.es/, accessed November 2006.

24 **Pina, P.** and **De Pablo, J.** Benefits obtained from the estimation and distribution of realistic taxi times. In the 6th USA/Europe ATM2005, R&D Seminar, 2005, available from http://atmseminar.eurocontrol.fr/all-seminars/6th-usa-europe-atm-2005-r-d-seminar / papers _ calendar, accessed June 2008.

APPENDIX

Notation

A	set of arcs
ATT	decision variable equal to average TT
$d(w), o(w)$	destination and origin nodes, respectively, for flight w
$D_{\mathrm{IN}}, D_{\mathrm{OUT}}$	decision variable equal to waiting delays of arrival and departure traffic, respectively
e_w	area occupied by aircraft for flight w
E_{it}^{w}	decision variable equal to 1 if flight w is waiting at node i in time period t, and 0 otherwise
$F(i), R(i)$	sets of arcs with origin and destination node i, respectively
i, j, k	node indices. (i, j) refers to an arc with tail node i and head node j
I_{C}	decision variable equal to total controller interventions during the planning time
K	subset of nodes where conflicts need to be limited
n	maximum number of interventions controllers can handle in any given period
N	set of nodes
$\hat{N}$	subset of nodes whose availability needs to be decided
$N^{\mathrm{P}}, N^{\mathrm{AR}}$	sets of parking and access runway nodes, respectively
N^{W}	subset of holding points and other waiting nodes
q_i	maximum capacity of facility or node i
r_i^{w}	estimated time for aircraft w from node i to its runway or parking destination
t	time period index
t_{ji}	transit time from node j to an adjacent node i
$t(w)$	departure or arrival time for flight w
T	set of time periods
TT	decision variable equal to total TT
TT_w	decision variable equal to TT for aircraft w
$\mathrm{TT}_w^{\max}$	maximum travel time allowed for flight w
w	flight index
W	set of flights
$W^{\mathrm{A}}, W^{\mathrm{D}}$	subsets of arriving and departing flights, respectively
WTT	decision variable equal to the worst (longest) routing time among all aircraft during a given planning time
X_{ijt}^{w}	decision variable equal to 1 if flight w is waiting at node i in time period t, and 0 otherwise
Y_i	(vector form $\boldsymbol{Y}$) decision variable equal to one if airport facility i is available, and zero otherwise
$\alpha_{\mathrm{L}}, \alpha_{\mathrm{Ic}}, \alpha_{\mathrm{TT}}, \alpha_{\mathrm{Din}}, \alpha_{\mathrm{Dout}}, \alpha_{\Gamma-}, \alpha_{\Gamma+}$	weights for allocating capacity to potential waiting nodes, number of conflicts, routing time, delays for arriving and departing traffic, and number of departing and arriving flights, respectively
γ_{kt}	decision variable equal to 1 when two or more aircraft approach node k around period t
Γ^{+}, Γ^{-}	decision variable equal to the achieved number of arrivals at the gate and take-offs, respectively, during a given planning period