

Airport taxi planning: Lagrangian decomposition

Ángel G. Marín*

Departamento Matemática Aplicada y Estadística, Universidad Politécnica de Madrid, E. T. S. Ingenieros Aeronáuticos, Plaza Cardenal Cisneros, 3, Madrid 28040, Spain

SUMMARY

The airport taxi planning (TP) module is a decision tool intended to guide airport surface management operations. TP is defined by a flow network optimization model that represents flight ground movements and improves aircraft taxiing routes and schedules during periods of aircraft congestion. TP is not intended to operate as a stand-alone tool for airport operations management: on the contrary, it must be used in conjunction with existing departing and arriving traffic tools and overseen by the taxi planner of the airport, also known as the aircraft ground controller. TP must be flexible in order to accommodate changing inputs while maintaining consistent routes and schedules already delivered from past executions. Within this dynamic environment, the execution time of TP may not exceed a few minutes. Classic methods for solving binary multi-commodity flow networks with side constraints are not efficient enough; therefore, a Lagrangian decomposition methodology has been adapted to solve it. We demonstrate TP Lagrangian decomposition using actual data from the Madrid-Barajas Airport.

KEY WORDS: taxi planning; airport management; binary multi-commodity flow network; Lagrangian decomposition

1. INTRODUCTION

The annual increase in average flight delays is in large part due to increased traffic in the airport terminal area and in aircraft ground operations. This situation is even more complicated during peak hours or periods of low visibility or when flight schedules become irregular. During these situations, appropriate handling of flight ground movements is crucial to maintain adequate airport capacity.

Arrival management (AM), departure management (DM), and gate management (GM) are the primary tasks in operating an airport. AM plans the arrival sequence for landing aircraft in a given time horizon, DM gives push-back orders to departing aircraft at the stands, and 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 (TP) problem.

Taxi planning is a dynamic problem that considers flight movement decisions during a given planning time; however, the data change every few minutes. Thus, continuous solution updates are needed. Every minute, an aircraft requests a takeoff or a landing. Thus, the methods proposed to solve TP must run in real time in practice and be integrated with other airport management planning tools to generate updated outputs when new data become available. This manuscript examines the TP module; however, the inter-relation among airport management modules is not explicitly formulated.

Taxi planning attempts to determine the optimal route and scheduling of an aircraft in a given planning horizon in which the current and previous outputs of other airport management modules are delivered as inputs to the TP module. We use the most recent flight schedule to maintain consistency while re-optimizing according to updated conditions.

*Correspondence to: Ángel G. Marín, Departamento Matemática Aplicada y Estadística, Universidad Politécnica de Madrid, E. T. S. Ingenieros Aeronáuticos, Plaza Cardenal Cisneros, 3, Madrid 28040, Spain. E-mail: angel.marin@upm.es

To illustrate airport planning module integration, we assume that any flight that may use the taxiways during the incumbent planning horizon must be considered in the given TP run. Figure 1 depicts this idea; assume that the TP horizon starts at the time mark labeled “current,” and consider 10 possible flights: five departures (labeled D_n) and five arrivals (labeled A_n), for $n=1\dots5$. 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 the runway, respectively, before the beginning of the planning horizon, but they are still taxiing and therefore must be included in the incumbent TP run. Their current location and immediate direction must be provided to the TP by the aircraft ground controller. Flights $D3$ and $D4$ have departure times (provided by the DM) within the planning horizon. Likewise, flights $A3$ and $A4$ have landing times (provided by the AM) within this period. Thus, these four flights must be considered by the incumbent TP run, regardless of whether or not they arrive at the runway (or the stand).

1.1 State of the art

In TP, optimization models are rarely used; instead, simulation models such as SIMMOD [1] have been proposed, but they lack the ability to preemptively optimize aircraft routes and schedules, which is the ultimate goal of the TP model.

Discrete optimization techniques for airport planning have been used by Bianco *et al.* [2] to study static and dynamic AM. For the static case, they modeled the problem as a cumulative traveling salesman problem with ready time and proposed two lower bounds to define heuristics. GM has been studied by Bolat [3]; he minimizes the variance of gate idle times by making the initial assignments insensitive to variations in flight schedules. Yan *et al.* [4,5] studied a heuristic approach to airport gate reassignments for stochastic flight delays and considered the gate assignment recovery problem as a consequence of temporary airport closures.

Durand and Gotteland [6] employed a pattern recognition model based on genetic algorithms to characterize taxiing conflicts. Anagnostakis *et al.* [7] presented several formulations for DM and provided a description of runway operations planning; they presented only preliminary approaches to solve the problem in a dynamic context. Andersson *et al.* [8] proposed two queuing models to capture the “taxi-out” and “taxi-in” processes. These topics have also been studied by Idris *et al.* [9] to estimate the taxi-out time (difference between scheduled gate departure time and takeoff time) in terms of parameters such as runway and terminal configurations, downstream restrictions, and takeoff queues.

Anagnostakis [10] has presented a study of the structure and properties of the runway operations planning problem and developed a decomposition-based algorithm. Stoica [11] proposed an adaptive approach to manage available aircraft routes. Stamatoopoulos *et al.* [12] described an integrated set of models for the estimation of the capacity of an airfield and the associated delays.

Roling and Visser [13] developed a planning support tool to optimize the routing and scheduling of airport surface traffic using robust solutions to uncertainty data. The tool is designed to adapt to perturbations in these input conditions and account for failure in the actual execution of surface trajectories.

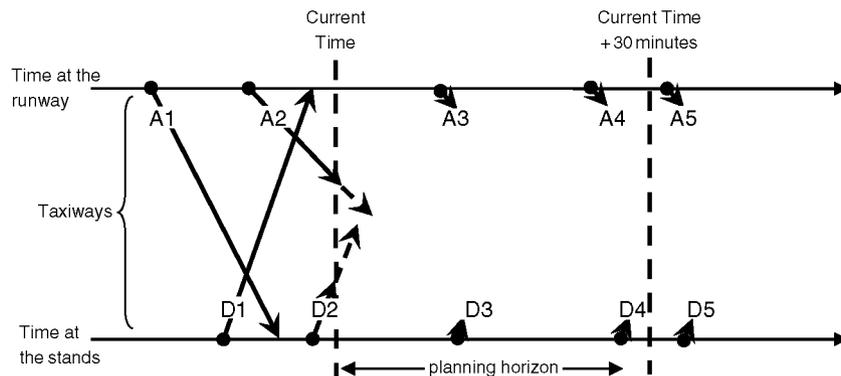


Figure 1. Flights involved in a 30-minute taxi planning run.

Some authors, such as Balakrishna *et al.* [14–16], have investigated the taxi-out prediction by using a nonparametric reinforcement learning-based method set in the probabilistic framework of stochastic dynamic programming.

Marín [17] developed the first TP model as a binary multi-commodity network flow model and solved it using the branch and bound (B&B) and fix and relax (F&R) approaches; however, these methods are not fast enough for real-time operations. Marín and Codina [18] extended the use of TP to model airport design, which can be used to support decisions regarding airport configurations. Marín and Salmerón [19] studied TP as an integrated airport tool and introduced extensions to the model.

1.2 Contributions

Taxi planning allows airport taxi planners to consider tradeoffs among various goals, such as aircraft routing time, traffic delays, and number of aircraft completing their routes in a horizon planning. TP optimizes space–time aircraft trajectories by considering congestion, bottlenecks, and airport capacities, essentially providing a guide for the aircraft ground controller in making decisions to reduce taxiing time, in meeting takeoff windows, and in avoiding runway conflicts.

Taxi planning is a complex mathematical problem that must be integrated with other management tools (AM, DM, and GM) and generates updated route sequences in a continuous, iterative process; moreover, each update must be realized every 1 to 2 minutes. The TP model must be solved very efficiently, which has led to the development of an efficient computational tool to optimize airport ground traffic via the Lagrangian decomposition (LD) method.

The LD method is based on heuristics defined to solve the sub-models and implement the sub-gradient methodology. The efficiency of the LD method has allowed TP to be integrated with other airport management modules, where it must be able to execute multiple times within a short period. Computational experiments have been validated with actual data from the Madrid-Barajas Airport (MBA).

1.3 Synopsis

This manuscript is organized as follows: Section 2 describes the optimization model of TP as defined by multi-commodity network flow formulation; Section 3 describes how the LD method is adapted to TP (an algorithm is defined, and two basic routines are described in detail); and Section 4 describes how the computational experience is validated using data from the MBA; finally, Section 5 discusses the main contributions of TP and outlines future research that may promote new developments in the field.

2. TAXI PLANNING MODEL

Taxi planning is modeled as a multi-commodity binary network flow model using a space–time network to represent the conflicts between aircraft and the congestion generated when several flights try to use simultaneously the limited resource capacity of an airport.

A directed graph $G(N,A)$ is used to define the TP model. N represents the set of airport “nodes” and $A = \{(i,j), \forall i,j \in N\}$ the set of airport “links.” A node $i \in N$ can be a stand, a holding area, a junction or intersection of two or more taxiways, a node to represent aircraft queues in taxiways, an access runway header, or an exit runway.

A link $(i,j) \in A$ connecting nodes i and j typically represents a physical taxiway, but in TP, they are also used to represent entrance and exit ways, to or from a stand–runway or intermediate segments of a taxiway where an aircraft may need to stop, for example.

The links can be classified as “moving” or “waiting.” During moving links, the flight movements are characterized by a fixed average velocity, which is used to define the link time t_{ij} that is used by any flight moving from node i to node j . During waiting links, the flights may stop, and t_{ij} has a value of 1.

To account for the time within the planning horizon, the network is replicated over time by considering the set $T = \{0, 1, \dots, |T|\}$. Each $t \in T$ represents a time mark or “period” of equal length tp . Typically, tp is taken as a step of 30 seconds. In this case, $t=0$ would represent the current time, and $t=60$ would represent 30 minutes after the current time.

The TP model is based on the definition of a space–time graph, $G^*(N^*,A^*)$, where N^* is obtained by replicating $|T|$ times the previous node set and $\{(n, t) : \forall n \in N, \forall t \in T\}$ and A^* are the links defined between them: $\{(i, t), (j, t'), \forall i, j \in N, \forall t, t' \in T\}$.

A set of flights W is considered within the planning horizon. Each flight w is defined by an origin node $o(w)$, a destination node $d(w)$, and an origin time $t(w)$ that may be a departure time (given by DM) for departing traffic or a landing time (given by AM) for arriving traffic. On the space–time graph, the origin is a single node $\{o(w), t(w)\}$, but the destination is a set of space–time nodes that is defined by the set of destination nodes $d(w)$ during different periods or intermediate nodes between the origin and the destination at the end of the planning horizon; the flight may or may not arrive at its destination during this period. In this second group of flights, we must estimate the time needed (r_i^w) for aircraft w to arrive at its destination from node i , that is, the node where the aircraft is located at the end of the planning horizon. r_i^w is obtained using an external, shortest-path algorithm that is run prior to TP.

The TP optimal variables are defined as follows:

$$E_{it}^w = 1, \text{ if flight } w \text{ is waiting at node } i \text{ in period } t; \text{ and } 0, \text{ otherwise.}$$

$$X_{ijt}^w = 1, \text{ if flight } w \text{ leaves from node } i \text{ to node } j \text{ at period } t; \text{ and } 0, \text{ otherwise.}$$

Taxi planning minimizes the total routing time for all flights. The objective function is defined by the time periods spent to route all aircraft.

$$z = \sum_{w \in W} \sum_{i \geq t(w)} \lambda^w \left(\sum_{ij \in A} t_{ij} X_{ij,t}^w + \sum_{i \in N^w} E_{i,t}^w \right) + \sum_{w \in W} \sum_{i \in N^w} \lambda^w r_i^w E_{i,|T|}^w \quad (1)$$

where λ^w is the priority of aircraft w . Priority is used so that some aircraft have preference in event of conflict or congestion. λ^w is a real value greater than 1 if w is a preference flight.

The following constraints describe the TP feasible set: the nodal flow conservation constraints.

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

Consider t_{ji} as the transit time from node j to an adjacent node i . The above equations establish that in order for an aircraft w to either (i) wait at node i in period $t+1$ or (ii) start moving from i to j in period $t+1$, it is required that either w is waiting at node i in period t or w has departed from adjacent node j for 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 of node i , typically known as the forward and reverse stars of node i , respectively.

The flow node conservation constraints need to account for the aircraft at the origin node, $o(w)$. Aircraft w may wait or move at $(o(w), t(w))$.

$$E_{o(w),t(w)}^w + \sum_{j \in F(o(w))} X_{o(w)j,t(w)}^w = 1, \forall w \in W \quad (3)$$

For practical reasons, we force all aircraft to be located at some node (instead of traveling on an arc) at the end of the planning horizon $t=|T|$. (This includes the ‘‘air node’’ if the aircraft has taken off.)

$$\sum_{i \in N} E_{i,|T|}^w = 1, \forall w \in W \quad (4)$$

A visual illustration of the roles played by variables E and X of the previous flow conservation constraints (2), (3), and (4) for aircraft $w \in W$ is shown in Figure 2.

The formulation of the previous model may be shortened by setting a equal to the space–time links so that $a = i,j,t$ for moving arcs and $a = i,t$ for waiting arcs and by setting x_a^w equal to the binary flow variables (these variables are equivalent to the previously defined $X_{ij,t}^w$ or E_{it}^w). The following variable vectors are defined by $x^w = \{x_a^w\}$ and $x = \{x^w\}$. With this new notation, the TP model may be concisely defined by

$$\text{TP} : z_{\text{TP}}(x) = \min_{x_a^w \in \{0,1\}} \sum_{w \in W} \sum_{a \in A^*} c_a x_a^w \quad (5)$$

$$A^w x^w = h^w, \quad \forall w \in W \quad (\text{node flow conservation}) \quad (6)$$

$$\sum_{w \in W} x_a^w \leq q_a, \quad \forall a \in A^* \quad (\text{link capacity}) \quad (7)$$

where the “Node flow conservation” constraints (in matrix format) are equivalent to the previous flow conservation constraints and the “Link capacity” constraints are the remaining constraints relative to different link capacities. The coefficients of the objective function are c_a . The link capacity are q_a .

The “Link capacity constraints” (details of them have been considered in my previous paper about TP, [17–19]) are the following:

- Taxiway holding node capacities.
- Ordinary and exit runway node capacities.
- Parking, access runway and fictitious node capacities.
- Conflict between flights, etc.

2.1 Extended taxi planning

The TP may be solved by decomposing the constraints into easy and difficult ones. The easy constraints are the node flow conservation constraints, which are separable by nodes and flights. The link capacity constraints are the difficult ones as they are only separable by links.

The extended TP (ETP) is formulated from the previous TP when we added some additional constraints that are not link separable (side constraints, SCs).

Several SCs may be defined. As an example of an SC, constraints concerning the separation between takeoffs at departure runways (DRs) are considered, takeoff separation constraints (TSCs). Assuming that the departure and arrival runways are independent, we define a TSC for DR p and the access runway point set of each runway by N_p^{AR} .

The security takeoff time ($t_{sto}(w)$) for elements of the departure flight set (W^{D}) is equal to the separation time needed for the takeoff of other flights following the takeoff of flight w . This time must

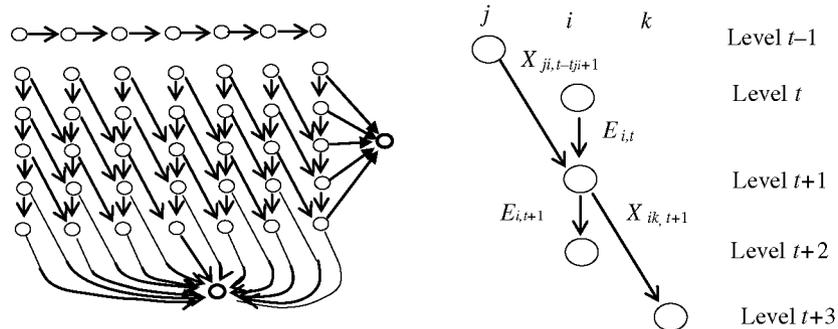


Figure 2. Space-temporal graphic for a unique taxiway with $T=5$, $t_{ji}=2$, and nomenclature for the decision variables X and E .

be equal to or greater than the takeoff time of the flight because $t_sto(w)$ includes additional security time along the takeoff time. This time prevents the wake vortex separation of one flight from affecting the takeoff of the next flight.

The TSC is formulated as follows:

$$\sum_{j \in N_p^{AR}} \left(\sum_{\substack{w \in W^D \\ j=d(w)}} \left\{ X_{j,an_p,t}^w + \sum_{t' \in [t, t+t_sto(w)]} X_{j,an_p,t'}^w \right\} \right) \leq 1, \forall p \in DR, \forall t \in T, \quad (8)$$

where an_p is the air node of DR p . This node models the final destination of any aircraft using the runway. ETP is defined by the previous TP objective function and constraints, as well as the TSC constraints.

3. TAXI PLANNING: LAGRANGIAN DECOMPOSITION

Extended TP may be solved by Branch and Bound B&B, but B&B is inefficient in terms of computational time as a result of the large number of binary variables of the model and the need to solve the problem very quickly. As a consequence of this inefficiency, other methodologies have been developed. Fix and Relax F&R is another methodology that has been considered for solving ETP. It is approximately 20% more efficient than B&B, but its speed is still not fast enough to solve ETP [17].

Various Lagrangian relaxation and decomposition approaches have been analyzed to solve ETP. As a result of this analysis (excluded here because of space limitations), LD has been selected to solve ETP. A central argument for this choice is that “the lower bound for the Lagrangian dual of the new approach is at least as strong as that of the best of the traditional Lagrangian relaxation approaches,” as proven by Jörnsten and Näsberg [20] and Guignard and Kim [21].

Lagrangian decomposition introduces a new variable that is a copy of the existing variable and the corresponding constraint that establishes equality between both copies of the variable (also known as variable splitting). The body of literature on LD and its extensions and applications is enormous. Therefore, we have not attempted to discuss this literature. We have previously used LD to solve electric capacity expansion for uncertain demand problems [22] and to solve maximal covering problems in code multiplexing access telecommunications networks [23].

Lagrangian decomposition variable splitting is used to group the constraints into two groups, depending on the copy of the variable used to define each group. ETP may be described by copies (x and y) of the original optimization constraints in the following way:

$$\text{ETP} : z_{\text{ETP}} = \min_{\substack{x \in \{0,1\} \\ y \in \{0,1\}}} \gamma_{\text{xsm}} \sum_{w \in W} \sum_{a \in A^*} c_a x_a^w + (1 - \gamma_{\text{xsm}}) \sum_{w \in W} \sum_{a \in A^*} c_a y_a^w \quad (9)$$

subject to

$$A^w x^w = b^w, \quad \forall w \in W \quad (10)$$

$$\sum_{w \in W} y_a^w \leq q_a, \quad \forall a \in A^* \quad (11)$$

$$\sum_{j \in N_p^{AR}} \left(\sum_{\substack{w \in W^D \\ j=d(w)}} \left\{ y_{jan_p,t}^w + \sum_{t' \in [t, t+t_sto(w)]} y_{jan_p,t'}^w \right\} \right) \leq 1 \forall p \in DR, \forall t \in T \quad (12)$$

$$x_a^w = y_a^w \quad \forall a \in A^*, \forall w \in W \quad (13)$$

where γ_{xsm} is the weight of the sub-model X (XSM) objective function term, γ_{xsm} can take values in the interval $(0, 1)$, and the weight of the sub-model Y (YSM) objective function term is $(1 - \gamma_{xsm})$. The final constraints are those of variable splitting.

The constraints are grouped into two sub-models, XSM and YSM . The master model (MM) is the Lagrangian dual of ETP, and it takes decisions to update the Lagrange multipliers of the variable splitting constraints (λ_a^w) , where λ is the vector of components (λ_a^w) .

The multipliers are applied at each iteration to redefine the sub-models, and the sub-model's solutions are used by the MM to update the multipliers; thus, a cycle that terminates with an optimal or an ϵ -optimal solution is defined.

The MM is defined using the ETP Lagrangian dual problem:

$$\underset{\lambda \text{ free}}{\text{maximize}} \quad L(\lambda) \quad (14)$$

where $L(\lambda)$ is the sum of two terms: $L_x(\lambda)$ and $L_y(\lambda)$. $L_x(\lambda)$ is the objective function of the sub-model with the flow node conservation constraints $XSM(\lambda)$. $L_y(\lambda)$ is the objective function of the sub-model with the link capacity and TSC $YSM(\lambda)$.

$$XSM(\lambda) : L_x(\lambda) = \underset{x \in \{0,1\}}{\text{minimize}} \quad \sum_{w \in W} \sum_{a \in A^*} (\gamma_{xsm} c_a + \lambda_a^w) x_a^w \quad (15)$$

subject to

$$A^w x^w = b^w \quad \forall w \in W \quad (16)$$

$XSM(\lambda)$ is a multi-commodity network flow problem separable by commodity w , and the sub-problem of each commodity w may be easily solved by the shortest-path algorithm.

$$YSM(\lambda) : L_y(\lambda) = \underset{y \in \{0,1\}}{\text{minimize}} \quad \sum_{w \in W} \sum_{a \in A^*} ((1 - \gamma_{xsm}) c_a - \lambda_a^w) y_a^w \quad (17)$$

subject to

$$\sum_{w \in W} y_a^w \leq q_a, \quad \forall a \in A^* \quad (18)$$

$$\sum_{j \in N_p^{AR}} \left(\sum_{\substack{w \in W^D \\ j=d(w)}} \left\{ y_{j,an_p,t}^w + \sum_{t' \in [t, t+sto(w)]} y_{j,an_p,t'}^w \right\} \right) \leq 1 \quad \forall p \in DR, \forall t \in T \quad (19)$$

$YSM(\lambda)$ is difficult to solve; therefore, we must define a heuristic to solve it.

3.1 Lagrangian decomposition algorithm

The Lagrangian decomposition algorithm is defined by iteratively solving the MM and the sub-models. To solve the MM, the sub-gradient method has been employed. It establishes a sequence of dual variables λ that attempts to follow an ascent direction from $L(\lambda)$.

The sub-gradient of $L(\lambda)$ at iteration k is defined by $sg_{a,w}^k = x_a^w - y_a^w$, where x_a^w and y_a^w are the XSM (λ^k) and $YSM(\lambda^k)$ solutions at iteration k , respectively. The dual sequence of λ is defined by the

direction sg^k as follows: $\lambda^{k+1} = \lambda^k + \theta^k sg_{a,w}^k$, where θ^k is the step size in this iteration and $\lambda^k = (\lambda_{a,w}^k)$.

An extension of the sub-gradient method is described in terms of Kuhn–Tucker optimal conditions [24]. The step size is given by the following step expression: $\theta^k = \frac{z_{ETP} - L(\lambda^k)}{\|sg^k\|^2}$, where $\|\cdot\|$ is the Euclidean norm. Because the optimal value of the objective function z_{ETP} is unknown, an upper bound UB^k is obtained by using a heuristic algorithm to obtain an ETP feasible solution. In this case, a coefficient η^k is added to the step expression; η^k has a value between 0 and 2 and usually begins at 2 and decreases by a factor of 2 if the value of the Lagrange objective function does not increase in a given number of iterations, $\mu_k : \theta^k = \eta^k \frac{UB^k - L(\lambda^k)}{\|sg^k\|^2}$.

The LD algorithm to solve ETP is defined as follows:

ETP Lagrangian decomposition algorithm:

- **Initialization:** Set $k=0$, $\lambda^0=0$, $LB^0=0$, $UB^0=\infty$.
- **While** ($k < k_{\max}$)
 - Solve $XSM(\lambda^k)$ and $YSM(\lambda^k)$. Let x^k be the optimal solution of $XSM(\lambda^k)$ and y^k be the solution of $YSM(\lambda^k)$. Let $L_x(\lambda^k)$ and $L_y(\lambda^k)$ be the values of the respective objective functions and set $L(\lambda^k) = L_x(\lambda^k) + L_y(\lambda^k)$. Update the lower bound LB^k : $LB^k = \max\{LB^{k-1}, L(\lambda^k)\}$.
 - Find an ETP^k feasible solution xf^k solution. Update the upper bound UB^k : $UB^k = \min\{UB^{k-1}, z_{ETP}(xf^k)\}$.
 - If some termination criteria are fulfilled, stop.
 - Update $\lambda : \lambda^{k+1} = \lambda^k + \eta^k \frac{UB^k - LB^k}{\|sg^k\|^2} (sg^k)$
- **End while.**

The algorithms used to solve the sub-model and obtain the ETP feasible solution will be defined in the next sub-sections. k_{\max} is the maximum number of iterations.

Various termination criteria may be used; the one that has been used in computational tests is the relative error criterion, which is defined as $\frac{UB^k - LB^k}{LB^k} < \epsilon$.

3.2 YSM algorithm

The YSM algorithm is first defined by considering a simplified YSM that is defined by deleting the non-link-separable constraints (TSC) at YSM . In this case, each simplified YSM is separable by link a , and each one is exactly solved considering that it is a semi-assignment binary model. For each $a \in A^*$, the simplified YSM is defined by

$$L_y^a(\lambda^k) = \underset{y_a^w \in \{0,1\}}{\text{minimize}} \sum_{w \in W} \left((1 - \gamma_{xsm}) c_a - \lambda_{a,w}^k \right) y_a^w \quad (20)$$

subject to

$$\sum_{w \in W} y_a^w \leq q_a \quad (21)$$

The links are established in decreasing order on the link modified cost $\left((1 - \gamma_{xsm}) c_a - \lambda_{a,w}^k \right)$, and y_a^w is assigned to the links with negative modified costs, whereas the capacity of the links is not used completely. The solution for each model is given by the following:

$$\begin{cases} y_a^w = 1, & \text{if } \left((1 - \gamma_{xsm}) c_a - \lambda_{a,w}^k \right) < 0, \text{ while } \sum_w y_a^w \leq q_a \\ y_a^w = 0, & \text{otherwise} \end{cases}$$

For each iteration k , the simplified YSM^k algorithm is defined by the following:

For each arc $a \in A$

- **Sort** $y_a = \{y_a^w\}$ by increasing the order given by their associated modified cost $\left((1-\gamma_{xsm})c_a - \lambda_{a,w}^k\right)$. Let (W^*) be the set of flights with negative modified costs. $\left((1-\gamma_{xsm})c_a - \lambda_{a,w_1}^k\right) \leq \left((1-\gamma_{xsm})c_a - \lambda_{a,w_2}^k\right) \leq \dots \leq \left((1-\gamma_{xsm})c_a - \lambda_{a,w_{|W^*|}}^k\right) \leq 0$

Obtain the ordered list of flights (OW), where

$$OW = \{y_a^{w_1}, y_a^{w_2}, y_a^{w_3}, \dots, y_a^{w_{|W^*|}}\}, \text{ in which elements are referred as } y_a^{w_i}.$$

- **Set** $y_a^w = 0, \quad \forall w \in W$.

- **Set** $i = 1$

- **While** for each $w_i \in OW \left(\sum_{w_i \in OW} y_a^{w_i} \leq q_a, \forall i \leq |W^*| \right) : y_a^{w_i} = 1 \quad i = i + 1$

End While

End For

Following the same idea, the arcs of $YSM(\lambda^k)$ affected by TSC (not separable by arcs) are available to define a heuristic. For the remaining capacity arcs, the exact previous YSM algorithm is used. To simplify the heuristic description, we assume an independent takeoff runway and only one access runway node per takeoff.

The links affected by the TSC constraints are sorted by decreasing origin time, and the departing flights are assigned to the sorted set once the departing runway becomes available for use.

Now, we consider only the links affected by the TSC. For each DR considered, we define A_p^{*TS} as $A_p^{*TS} = \{a \in A^* : a = (j, an_j), \forall j \in N_p^{AR}\}$. From this set, we choose links that have negative modified costs: $\tilde{A}_p^{*TS} = \{a \in A_p^{*TS} : \left((1-\gamma_{xsm})c_a - \lambda_{a,w}^k\right) < 0\}$. The elements of this set are sorted by decreasing origin time. The sorted set is accounted by \bar{A}_p^{*TS} . The links are located over this set such that each link

$a \in \bar{A}_p^{*TS}$

$$\begin{cases} y_a^w = 1, & \text{if } \left((1-\gamma_{xsm})c_a - \lambda_{a,w}^k\right) < 0, \text{ while } \sum_{j \in N_p^{AR}} \left(\sum_{\substack{w \in W^D \\ j=d(w)}} \left\{ y_{j,an_p,t}^w + \sum_{t' \in |t, t+t_{sto}(w)|} y_{j,an_p,t'}^w \right\} \right) \leq 1 \\ y_a^w = 0, & \text{otherwise} \end{cases}$$

For each iteration k , the YSM^k algorithm is defined as follows:

For each link $a \in A_p^{*TS}$ and $p \in DR$:

- **Set** $y_a^w = 0, \quad \forall w \in W$

- **Define** the set of links affected by TSC with negative modified cost $(1-\gamma_{xsm})c_a - \lambda_{a,w}^k \leq 0 \rightarrow \tilde{A}_p^{*TS}$

- **Sort** \tilde{A}_p^{*TS} by decreasing minimum time at origin of the link, getting $\bar{A}_p^{*TS} = \{a_1, a_2, \dots, a_n\}$, where $a_l = (i_l, j_l, t_l) l = 1, 2, \dots, |\bar{A}_p^{*TS}|$, verifying $t_1 < t_2 < \dots < t_n$

- **For each** $l = 1, \dots, |\bar{A}_p^{*TS}|$

- **If** $\sum_{w \in W^D} \left\{ y_{i_l, j_l, t_l}^w + \sum_{t' \in |t_l, t_l + t_{sto}(w)|} y_{i_l, j_l, t'_l}^w \right\} \leq 1$

$$\text{Set } y_{a_l}^w = 1$$

- **End If**

End For

End For

3.3 XSM_ETP feasibility algorithm

To obtain an improved computational behavior of the sub-gradient method, we need an ETP feasible solution to update the ETP upper bound. The XSM_ETP feasibility algorithm calculates an ETP feasible solution xf_a^k from an XSM(λ^k) solution x^k .

The XSM_ETP feasibility algorithm calculates the shortest paths for all aircraft and then chooses the path that has the minimum destination time and the minimum path cost. This path is saved, and the residual link capacity is updated until some link collapses. The process is repeated for all aircraft.

For each iteration k, the XSM_ETP feasibility algorithm is defined as follows: Given x^k and UB^k for XSM^k

➤ **Set**

- List of aircraft with fixed path: $Wf = \emptyset$,
- List of aircraft without fixed path: $Wnf = W$
- Calculate the link modified cost, those used in XSM^k: $\gamma_{smx}c_a + \lambda_{a,w}^k$ and the residual link capacity, $res_a = q_a, \forall a \in A^*$

➤ **For each** $w = 1, \dots, |W|$

- Calculate the shortest path for all aircraft belonging to Wnf , using only links that have positive residual link capacity: $res_a > 0$
- Choose the aircraft w_c that arrives at its destination at minimum time, and for aircraft with equal time, choose the aircraft with the minimum path cost.
- Fix the path for this aircraft and update: $Wf \leftarrow Wf + w_c; Wnf \leftarrow Wnf - w_c$
- **Update** the residual link capacity: $res_a = q_a - \sum_{w \in Wf} x_a^w$

End for

➤ **Calculate** new upper bound UBf

Set $UB^k = \min(UBf, UB^k)$

4. COMPUTATIONAL EXPERIMENTS

All TP test results presented in this paper are based on actual flight planning data from the MBA during the year 2005, as supplied from the Spanish Air Navigation Agency [25]. The airport experienced annual traffic increases of 17.5%, 3.5%, -0.5%, 5.4%, 7.9%, and 9% during the years 1999 to 2005, respectively. The greatest congestion problems have arisen during the summer, when the monthly traffic exceeds four million passengers.

The MBA comprises six terminals, 166 stands, an arrival runway with five exits, a DR with three takeoff 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 3 depicts as example one of the several topology networks used to represent it. Nodes S1, ..., S5 are the exit nodes of the landing runway. Nodes NEP1, NEP2, and NEP3 are the access nodes of the takeoff runway. Terminal areas are denoted by 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 takeoff runway.

Steps are being taken to reveal the computational efficiency of TP as a tool for operational purposes. According to LEONARDO [26,27], "The newly developed TP tool and the gate/airline simulation tools have been verified using the LEONARDO CDM (Collaborative Decision Making) requirement and have performed sufficiently well before the start of the validation period."

Lagrangian decomposition was coded using the C++ Visual Studio compiler (Microsoft Corp., Redmond, WA, USA). B&B was used as a reference, and it was executed using GAMS 21.3 (Gams Development Corp., Washington DC 2007, USA) with CPLEX 9.0 (International Business Machines Corp., Armonk, N.Y., USA) solver. The computational tests were run using a MiTAC laptop (MiTAC International Corp., Taipei, Taiwan) with an AMD Athon 64Bit Processor (Advance Micro Devices, Inc., Sunnyvale, CA, USA), 3000MHz, 1 GB of RAM, and a beta version of Windows XP 64Bit (Microsoft Corp., Redmond, WA, USA).

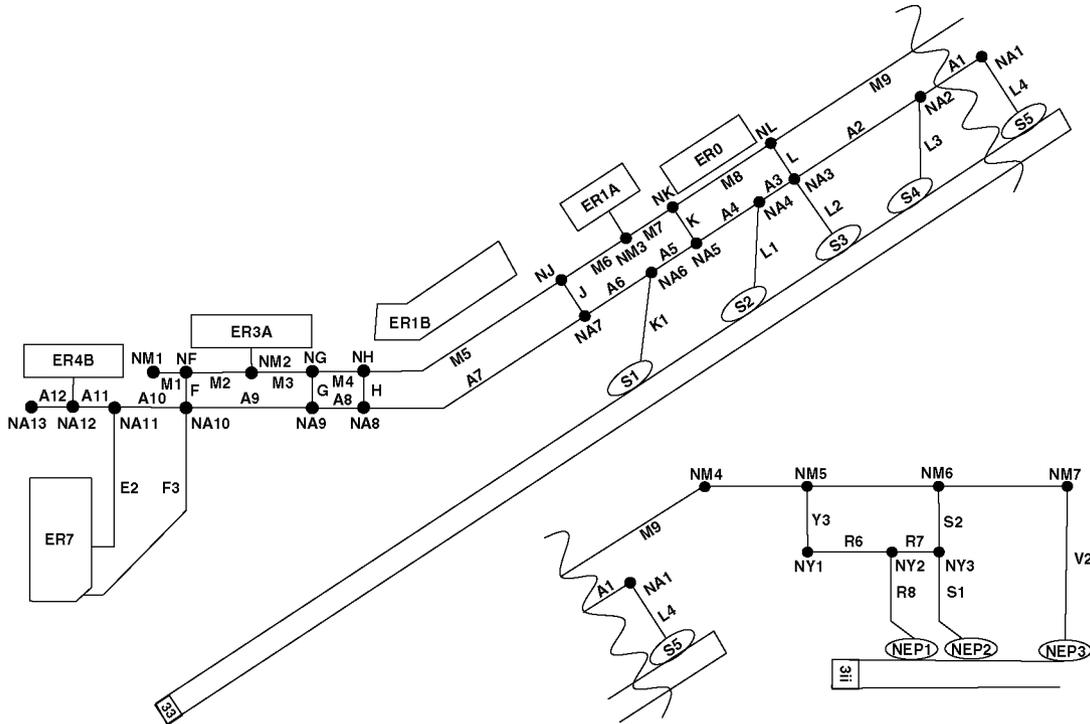


Figure 3. Madrid-Barajas Airport. Taxi planning network depiction, with landing runway (33) and takeoff runway (3ii). (Not to scale.)

The size of the space–time networks is very large. To substantially reduce the number of variables in the problem, a pre-processor has been defined to eliminate the node and links that will not be used by the considered flights. This pre-processor also prepares the files necessary for B&B and LD to solve the TP test network. The pre-processor also has other functions, such as verifying the coherence of the input data.

For a network like that of the MBA (Figure 3), we took as an example the flight data from 13:30 to 14:00, corresponding to an item of Table 1 with 14 arrivals and 16 departures; moreover, the data set has 230 nodes and 552 links, and the airport space–time network has 13 800 nodes and 33 120 links. The number of binary variables is obtained by multiplying the number of nodes and links by the number of flights (30). These considerations give a total of 1 407 600 binary variables. In the previous calculation, it was assumed that the time period (t_p) is 30 seconds, that the planning horizon time (t_{HP}) is 30 minutes, and that the number of periods ($|T|$) is 60.

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 taxiing times. Pina and De Pablo [28] stated that “since the possible combinations of stands and runways will greatly increase, the dispersion of the taxiing times will also increase. If we want to benefit from the new Madrid-Barajas airport, the realistic estimation of taxiing times will become a major issue.”

Lagrangian decomposition and B&B have many parameters that control their efficiency. A large number of prior tests (not included in this paper) have been performed to determine the optimal parameter configurations. The differences between these tests are not important. The following LD parameters were used for the computational tests: $\eta^0 = 185$, $\mu_k = 15$, $\lambda_{a,w}^0 = 13$, $\gamma_{xsm} = 0.9$. This parameter study has been detailed in the manuscripts by Jörnsten and Näsberg [20] and Barbas and Marín [23].

The most important parameter is the optimal per cent tolerance; for both B&B and LD, 10% ($\varepsilon=0.1$) has been considered. In any case, the solutions are feasible.

Table 1 shows the comparative tests between B&B and LD applied to ETP with a t_{sto} of 45 seconds for each flight using data from the MBA. The first column is the case that is analyzed; the

Table I. Branch and bound and Lagrangian decomposition comparative computational results.

Period	W	AW	DW	B&Bt (seconds)	B&Bv	LDt (seconds)	LDv
09:30–10:00	12	0	12	18.04	263	0.85	263
09:35–10:05	16	0	16	32.21	357	1.38	357
09:40–10:10	15	0	15	29.95	322	1.25	322
09:45–10:15	13	0	13	21.91	284	0.97	286
09:50–10:20	20	0	20	39.62	471	13.20	472
09:55–10:25	19	0	19	38.98	452	16.90	453
10:00–10:30	18	0	18	44.90	450	9.87	450
10:05–10:35	21	0	21	44.41	515	40.70	518
10:10–10:40	21	0	21	43.15	552	22.90	553
10:15–10:45	19	0	19	45.52	535	33.80	536
10:20–10:50	17	0	17	38.97	430	23.10	432
10:25–10:55	19	0	19	40.39	495	19.70	498
12:25–12:55	15	13	2	33.84	169	0.82	170
12:30–13:00	20	16	4	51.44	227	1.11	228
12:35–13:05	23	19	4	59.56	252	2.39	254
12:40–13:10	27	19	8	68.01	355	3.18	357
12:45–13:15	28	19	9	78.75	396	3.35	396
12:50–13:20	31	19	12	94.32	489	6.69	491
12:55–13:25	32	17	15	106.30	520	53.0	526
13:00–13:30	31	18	13	92.38	470	3.79	473
13:05–13:35	28	16	12	89.03	442	3.35	445
13:10–13:40	28	17	11	89.04	402	3.27	402
13:15–13:45	30	17	13	89.04	433	3.35	432
13:20–13:50	29	16	13	91.12	413	3.32	414
13:25–13:55	28	15	13	91.84	436	3.44	439
13:30–14:00	30	14	16	95.15	520	3.76	520
13:35–14:05	28	13	15	107.50	490	52.30	491
13:40–14:10	24	9	15	100.10	450	27.90	451
13:45–14:15	17	6	11	54.68	319	3.06	320
13:50–14:20	12	4	8	31.88	233	1.07	233
14:25–14:55	16	11	5	18.25	192	0.77	192
14:30–15:00	21	15	6	37.74	257	1.44	257
14:35–15:05	28	18	10	66.80	349	2.58	349
14:40–15:10	33	20	13	106.40	435	4.03	435
14:45–15:15	31	17	14	100.40	453	3.98	455
14:50–15:20	27	16	11	83.23	378	3.14	379
14:55–15:25	28	15	13	87.78	425	3.81	426
15:00–15:30	25	12	13	79.10	393	4.72	397
15:05–15:35	26	12	14	86.49	485	19.90	491
15:10–15:40	26	11	15	83.58	490	39.50	498
15:15–15:45	32	14	18	93.58	528	44.90	528
15:20–15:50	35	15	20	116.10	607	42.70	607
15:25–15:55	34	17	17	110.70	548	7.51	549
15:30–16:00	32	16	16	107.50	528	49.50	528
15:35–16:05	30	13	17	96.02	539	45.30	541
15:40–16:10	26	11	15	67.40	441	17.60	443
15:45–16:15	17	7	10	34.60	343	29.30	344
15:50–16:20	13	4	9	29.11	259	1.73	261
15:55–16:25	8	0	8	12.31	182	6.57	184
16:00–16:30	4	0	4	4.85	77	0.39	77

W, total number of flights; AW, number of arrivals; DW, number of departures; B&Bt, branch and bound CPU time; B&Bv, B&B ϵ -optimal objective function value; LDt, Lagrangian decomposition CPU time; LDv, LD ϵ -optimal objective function value.

next three columns are the number of flights. The second column is the total number of flights (IW) (W in Table 1), the next column is the number of arrivals (IW^A) (AW in the Table 1), and the third column is the number of departures (IW^D) (DW in Table 1). The following two columns are the B&B CPU (Central processing unit) time (B&Bt) and the B&B ϵ -optimal objective function value (B&Bv). For the B&Bt, all times (in seconds) used by GAMS/CPLEX are included. The last two columns are the LD CPU time (LDt, in seconds) and the LD ϵ -optimal objective function value (LDv).

It may be observed that the LD code is more efficient than that of the B&B; in some cases, it is more than 30 times faster, but in others, only approximately two times faster. Table 1 presents a portion of the computational results. The results that do not correspond to rush hours have been omitted. The same results have been obtained with other cases in the same network and using other topologies.

As shown in Table 1, results are presented from various planning periods. The data are given in increments of 5 minutes during rush hours. Thus, data uncertainty is minimized by this dynamic approach in which solutions are updated and communicated to the other modules every time those data change. This updating process must be executed at least once per minute. This condition justifies the necessity to use LD that needs less than 1 minute in each case.

Another advantage of LD compared with that of B&B (CPLEX/GAMS) is that the LD code is written in C++, which guarantees flexibility and independence with regard to any software constraints.

5. CONCLUSIONS AND FURTHER RESEARCH

The process of aircraft routing and scheduling through airport taxiways has been analyzed using simulation tools, but because only a few optimization tools exist, this study attempts to highlight their necessity. The TP model is a new optimization tool that represents the conflicts and airport congestion associated with taxiways in large airports. It is defined by a large space–time multi-commodity network with node and arc capacity constraints and other side constraints.

The B&R or F&R does not solve TP problems fast enough to be used in a dynamic context and to be integrated with other airport modules (AM, DM, and GM). Alternatively, the LD has been applied to solve it. Its efficiency in solving the sub-models and implementing the sub-gradient allows us to solve the TP in less than 1 minute.

The computational experiments were executed using data from the MBA. Computational results lead us to conclude that the LD is a better approach than the B&B. Its efficiency allowed TP to be integrated with other airport management modules, where it was run multiple times in a short period.

Although the idea of integrating all of the aforementioned management activities into a single optimization module is conceivable, the mathematical models involved would be too large to be handled efficiently by existing state-of-the-art solvers. Future research should focus on defining a model that combines TP with the other airport management tools. This combined airport management model would produce a more realistic perspective on airport problems, and it would avoid the problem of convergence among the results of the airport management modules.

A very important extension is to consider the robustness of TP when data may be uncertain or inexact, for instance when aircraft landing time is not exactly known, which may occur as a consequence of unexpected delays during the landing maneuver. The airport resources may also be affected by low-visibility conditions caused by rain, fog, or other contingencies.

ACKNOWLEDGEMENTS

This work was partially supported by the Ministerio de Fomento (Spain) under project PT2007-003-08CCPP and by the Ministerio de Educación y Ciencia (Spain) under projects TRA2005-09068-C03-01 and TRA2008-06782-C02-01. The author also wishes to thank to the anonymous reviewers for their helpful comments on this paper.

REFERENCES

1. ATAC Corporation. Airport and Airspace Simulation Model. 2006. URL <http://www.atac.com> [accessed on January 2011].
2. Bianco L, Dell’Olmo P, Giordani S. Scheduling models and algorithms for TMA traffic management. In *Modelling and Simulation in Air Traffic Management*, Bianco L, Dell’Olmo P, Odoni AR (eds). Springer Verlag: Berlin, 1997; 139–167.
3. Bolat A. Models and a genetic algorithm for static aircraft–gate assignment problem. *The Journal of the Operational Research Society* 2001; **52**:1107–1120.
4. Yan S, Tang CH. A heuristic approach for airport gate reassignments for stochastic flight delays. *European Journal of Operational Research* 2007; **180**(2):547–567.

5. Yan S, Chen CY, Tang CH. Airport gate reassignment following temporary airport closures. *Transportmetrica* 2009; **5**(1):25–41.
6. Durand N, Gotteland J. Genetic algorithms applied to air traffic management. In *Metaheuristics for Hard Optimization*. Springer Verlag: Berlin, 2006; 277–306.
7. Anagnostakis I, Clarke JP, Böhme D, Völckers U. Runway operations planning and control: sequences and scheduling. *Journal of Aircraft* 2001; **38**(6): 988–996.
8. Andersson K, Carr F, Feron E, Hall WD. Analysis and modelling of ground operations at hub airports. In *Proceedings of the Third USA/Europe Air Traffic Management R&D Seminar*, Napoli, 13–16 June, 2000. URL http://www.atmseminar.org/seminarContent/seminar3/papers/p_111_AGOADM.pdf [accessed on January 2011].
9. Idris H, Clarke JP, Bhuvra R, Kang L. Queuing model for taxi-out time estimation. *Air Traffic Control Quarterly* 2002; **10**(1):1–22.
10. Anagnostakis I. A Multi-objective, decomposition-based algorithm design methodology and its application to runway operations planning. *PhD Dissertation*, Massachusetts Institute of Technology, Cambridge, MA, 2003.
11. Stoica D. Analyse, représentation et optimisation de la circulation des avions sur une plate-forme aéroportuaire. *PhD Thesis*, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, Toulouse, 2004. Rapport LASS 04601.
12. Stamatopoulos MA, Zografos KG, Odoni AR. A decision support system for airport strategic planning. *Transportation Research. Part C* 2004; **12**:91–117.
13. Roling PC, Visser HG. Optimal airport surface traffic planning using mixed-integer linear programming. *International Journal of Aerospace Engineering* 2008; **2008**:732828.
14. Balakrishna P, Ganesan R, Sherry L, Levy B. Estimating taxi-out times with a reinforcement learning algorithm. In *Proceedings of the 27th Digital Avionics Systems Conference*, St Paul, MN, 26–30 October 2008. IEEE Press: New York, NY, 2008; 3.D.3-1–3.D.3-12.
15. Balakrishna P, Ganesan R, Sherry L. Airport taxi-out prediction using approximate dynamic programming: an intelligence-based paradigm. *Transportation Research Record* 2008; **2052**:54–61.
16. Balakrishna P, Ganesan R, Sherry L. Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: a case-study of Tampa Bay departures. *Transportation Research. Part C* 2010; DOI: 10.1016/j.trc.2010.03.003.
17. Marín A. Airport management: taxi planning. *Annals of Operations Research* 2006; **143**:189–200.
18. Marín A, Codina E. Network design: taxi planning. *Annals of Operations Research* 2008; **157**:135–151.
19. Marín A, Salmerón J. Taxi planner optimization: a management tool. *Proceedings of the Institution of Mechanical Engineers Part G—Journal of Aerospace Engineering* 2008; **222**:1055–1066.
20. Jörnsten KO, Näsberg M. A new Lagrangian relaxation approach to the generalized assignment problem. *European Journal of Operational Research* 1986; **17**:313–323.
21. Guignard M, Kim S. Lagrangian decomposition: a model yielding stronger Lagrangian bounds. *Mathematical Programming* 1987; **39**:215–228.
22. Marín A, Salmerón J. Electric capacity expansion under uncertain demand: decomposition approaches. *IEEE Transactions on Power Systems* 1998; **13**(2):333–339.
23. Barbas JA, Marín AG. Maximal covering code multiplexing access telecommunication networks. *European Journal of Operational Research* 2004; **159**:219–238.
24. Bazaraa MS, Sherali HD, Shetty CM. *Nonlinear Programming*. John Wiley & Sons: Hoboken, NJ, 1993.
25. AENA (Aeropuertos Españoles y Navegación Aérea). *Informe 2000 to 2005*. URL <http://www.aena.es/csee/Satellite?pagename=Estadisticas/Home> [accessed on January 2011].
26. LEONARDO. *Technical verification report for Barajas*. Document 41D01IND02R, Leonardo Project (European Commission), 2004. URL <http://leonardo.aena.es/> [accessed on January 2011].
27. LEONARDO. *Validation specification*. Document 51D01SIC10R, Leonardo Project (European Commission), 2004. URL <http://leonardo.aena.es/> [accessed on January 2011].
28. Pina P, De Pablo J. Benefits obtained from the estimation and distribution of realistic taxi times. In *Proceedings of the Sixth USA/Europe Air Traffic Management R&D Seminar*, 2005. URL http://atmseminar.eurocontrol.fr/past-seminars/6th-seminar-baltimore-md-usa-june-2005/papers/?b_start:int=60&-C= [accessed on January 2011].