Real-Time Scheduling of Aircraft Arrivals and Departures in a Terminal Maneuvering Area

Andrea D'Ariano, Dario Pacciarelli, and Marco Pistelli
Dipartimento di Ingegneria, Università degli Studi Roma Tre, via della Vasca Navale, 79 - 00146 Roma, Italy

Marco Pranzo
Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università degli Studi di Siena, via Roma, 56 - 53100 Siena, Italy

The aircraft scheduling problem (ASP) is the real-time problem of scheduling takeoff and landing operations at a congested airport in a given time horizon, taking into account the runways and the air segments in the terminal maneuvering area . The ASP can be viewed as a job shop scheduling problem with additional real-world constraints. Compared with the current literature based on job shop scheduling applied to solve the ASP, we enrich the existing models by including new formulations of relevant practical constraints. We introduce and analyze three alternative ASP formulations, in which the objective function is the minimization of delay propagation with respect to the off-line timetable. Scheduling rules, heuristic and exact methods are implemented and tested on instances from the Roma Fiumicino airport, in Italy. Computational experiments show that practical-size instances are solved to near-optimality by our branch and bound algorithm in a few seconds of computation. © 2015 Wiley Periodicals, Inc. NETWORKS, Vol. 65(3), 212–227 2015

Keywords: air traffic control; aircraft delay minimization; job shop scheduling; alternative graph; branch and bound

1. INTRODUCTION

The ever increasing air traffic demand of passengers and cargos all over the world is currently limited by the capacity of the airports, which are expected to become a serious bottleneck in air traffic control (ATC) operations in a near future. Aviation authorities are thus seeking methods to better use the existing airport infrastructure and to better manage aircraft movements in the proximity of airports, improving aircraft punctuality while maintaining the required level of safety. Decision support systems (DSSs) based on optimization algorithms may help to exploit at most the capacity available in the terminal maneuvering area (TMA) during operations. In this context, the optimization of takeoff/landing operations is a key factor to improve the performance of the entire ATC system. However, ATC operations are still mainly performed by human controllers with only a limited aid from automated systems. In most cases, computer support consists of a graphical representation of the current aircraft position and speed.

This article deals with the development of optimization models and algorithms for real-time aircraft scheduling at a busy airport. The algorithms have been tested on practical size instances from the Roma Fiumicino (FCO) airport, one of the world's busiest airports by passenger traffic and the main airport in Italy. Further computational results on the second busiest airport in Italy (Milano Malpensa, MXP) can be found in Pistelli et al. [25].

From a logical point of view, ATC decisions in a TMA can be broadly divided into: routing decisions, where a route for each aircraft has to be chosen from its current position to its destination. Scheduling decisions, where feasible aircraft sequencing and timing have to be determined in each airway, such that safety regulations are satisfied and a given performance index is optimized. Even if, in practice, routing and scheduling decisions in a TMA are taken simultaneously, the main objective of routing decisions is typically to balance the use of critical resources (e.g., alternative runways, air corridors) while that of scheduling is the delay minimization. This article focuses on real-time aircraft scheduling decisions with fixed routes, and we refer to this problem as the aircraft scheduling problem (ASP). We assume that the routing problem is solved off-line in a preliminary step. Aircraft routing and scheduling approaches are proposed in D'Ariano et al. [13].

Received December 2010; accepted December 2014
Correspondence to: A. D'Ariano; e-mail: a.dariano@dia.uniroma3.it
Contract grant sponsor: The Italian Ministry of Research, project FIRB "Advanced tracking system in intermodal freight transportation"; Contract grant number: RBIP06BZW8
DOI 10.1002/net.21599
Published online 6 February 2015 in Wiley Online Library (wileyonlinelibrary.com).

We now briefly recall the general procedure for takeoff/landing operations in the proximity of airports. For each TMA, landing aircraft move along predefined routes from an entry fix to a runway following a standard descent profile. During all the approach phases, a minimum separation distance between every pair of consecutive aircraft must be guaranteed for each air segment and runway of the TMA. This standard separation depends on the type and relative positions of the two aircrafts (at the same or different altitude). A minimum separation between aircrafts is mandatory and prescribed by international aviation regulations. By considering the different aircraft speeds, the safety distance can be translated in a separation time interval (STI). Similarly, departing aircraft leave the runway flying towards the assigned exit fix along an ascent profile, respecting separation standards. The runway can be occupied by only one aircraft at a time, and a safety separation time should be ensured between any pair of aircraft. Once a landing aircraft enters the TMA, it should proceed to the runway. However, it is possible to let aircraft wait in flight in holding circles until they can be guided into the landing sequence. Once in the holding circle, an aircraft can leave it only after the traversing of (or a multiple of) half length of the circle.

A landing aircraft is considered to be late when landing after its scheduled arrival time. A departing aircraft is supposed to takeoff within its assigned time window and is late whenever it is not able to accomplish the departing procedure within its assigned time window. A conflict occurs whenever two or more aircrafts do not respect the minimum required distance at the same air segment or runway. Any potential conflict must be detected and solved in real-time by human air traffic controllers by rescheduling aircraft movements [5].

In practice, the air traffic controllers of the TMA take their decisions in coordination with the en route controllers of neighboring areas. It is, therefore, worth investigation the extent at which DSSs should support the collaboration between air traffic controllers of different areas or should support only the decisions taken in the TMA, thus considering fixed the aircraft entrance time. In this article, we investigate two versions of the ASP. In the noncollaborative problem, the entrance time of each aircraft in the TMA is given in input and it is, therefore, a constraint to be satisfied in a feasible solution. In the collaborative problem, the entrance time of each aircraft in the TMA is only constrained to be greater than or equal to a minimum value, while the actual value has to be determined to optimize the objective function.

Summarizing the above discussion, the ASP can be stated as follows: given a set of aircraft willing to land/takeoff, and given for each aircraft its approach/leaving path, its current speed, its runway occupancy time, and a time window to accomplish the landing (departing) procedures, our approach assigns to each aircraft the start time from the fix (runway) and all relevant points in such a way that all aircraft conflicts are solved, all the constraints on safety separation distances are satisfied, and a given system performance index is optimized.

The original contribution of this article is threefold:

- With respect to the existing models for the management of aircraft traffic in a TMA, we increase the level of accuracy in modeling the ASP constraints, both in the collaborative and noncollaborative cases. Specifically, we combine the following modeling features: (i) holding circle constraints; (ii) STI constraints for air segments; (iii) blocking constraints for runways; and (iv) time windows constraints for the aircraft travel time in air segments.
- We adapt existing exact and heuristic algorithms to deal with the constraints of the specific ASP formulation.
- Extensive computational experiments carried out for the FCO airport demonstrate that practical size instances are solved at optimality or near-optimality by these algorithms within a short computation time, compatible with real-time application.

The article is organized as follows. Section 2 briefly reviews the literature on ASP, section 3 introduces our three formulations of the ASP, section 4 describes heuristic and exact solution methods, section 5 presents the computational results on the FCO airport and, finally, section 6 gives conclusions and future research directions.

2. LITERATURE REVIEW

The ASP has been the subject of many papers. In view of the survey papers of Kuchar and Yang [22], Ball et al. [5], and Bennell et al. [8], this section only reviews the papers most related to our work. The ASP can be broadly classified as static or dynamic, while existing models can be grouped as basic or detailed.

In the static ASP, landing/departing aircraft must be sequenced when all information is known in advance, whereas in the dynamic ASP aircraft enter the TMA one at the time, and a new sequence of takeoff/landing aircraft has to be recomputed every time a new incoming aircraft is known. We observe that the difference between static and dynamic problems is sometimes artificial, especially if previously scheduled aircraft can be resequenced when a new aircraft enters the TMA. In such cases, a static algorithm can be applied every time the next relevant event occurs, provided that it is fast enough to compute a solution during operations. Similarly, a dynamic approach may not be usable online if the CPU requirement is too high. In fact, for real-time purposes aircraft require a continuous monitoring of their position and speed, and therefore, the computation time of the algorithm is more important than static/dynamic classification.

Basic models include only the runways in the TMA, while detailed approaches also model the air segments. Most of the early papers on ASP falls in the former category and the choice is motivated by the fact that the runways are often the bottleneck of the TMA. With a basic model the ASP is typically formulated as a single or a parallel machine scheduling problem, while detailed formulations model the ASP as a job shop scheduling problem. In both cases, there can be additional constraints to take into account specific characteristics of the ASP. For example, no-wait constraints are used

to model the traversing of air segments [9, 21] and blocking constraints for the traversing of runways [1, 11]. A description of blocking and no-wait resources for general scheduling problems can be found in [18]. In the ASP, basic models are more tractable than detailed models and may lead to useful insights for the problem. At the same time, they are less realistic since bottleneck situations may happen also in air segments of the TMA and in any case a solution that is feasible for a basic model may not be feasible in practice. Basic models may have a limited impact on the practice of ATC and the recent trend of research is to incorporate more practical constraints in the model.

We next briefly review basic models for the ASP, more details are reported in [12]. Among the earliest papers on the dynamic ASP, Dear [15], Psaraftis [27], and Venkatakrishnan et al. [30] adopt the so-called constrained position shifting, that is, no aircraft can be sequenced forward or backward more than a specified number of positions with respect to the first in first out (FIFO) sequence.

Ernst et al. [16] tackle the static ASP using a specialized simplex algorithm. Beasley et al. [6] present a mixed-integer zero-one formulation for the static ASP and solve the problem using exact and heuristic algorithms. Hansen [19] proposes a genetic algorithm for the routing and sequencing of landing aircraft on a multiple runway case. Beasley et al. [7] formulate a dynamic ASP with multiple runways as a displacement problem. Zhan et al. [31] use an ant colony system combined with a rolling horizon approach to solve the dynamic ASP in a single runway setting.

Artiouchine et al. [3] present a basic model that addresses the problem of scheduling aircraft on a single runway. The objective is to assign landing times to aircraft with a monopattern, where all aircraft are identical and where a single holding pattern is only considered.

Soomer and Franx [28] developed an approach for the static ASP with a single runway based on airlines' priorities and collaborative decision making. Each airline owns a set of aircraft and provides for each aircraft the cost function associated to its delay.

Sölveling et al. [29] include several costs in the objective function, such as taxing, crossing, and gates idling operations, as well as CO_2 emissions costs. A mixed integer formulation of the problem is proposed and the problem is then solved by a general Mixed Integer Programming (MIP) solver.

Eun et al. [17] face a dynamic ASP in which the controllers are allowed to delay airborne aircraft using only a discrete set of delay values. The objective function is the minimization of aircraft delays. The solution approach is based on a branch and bound (BB) algorithm with linear programming and Lagrangian dual decomposition.

We next review detailed ASP models. This stream of research strives to develop models adherent to the ATC practice. The TMA is formulated as a job shop scheduling problem in which each machine models a portion of the TMA, either an air segment or a runway, and each aircraft corresponds to a job. The traversing of an air segment or a runway by a specific aircraft is known as an operation, whose processing time is equal to the traversing time of the associated air segment/runway. Additional constraints take into account the practical aspects of the ASP, such as speed constraints for each flying aircraft and sequence-dependent setup times to model STIs between consecutive aircraft.

Bianco et al. [9] consider a detailed model in which the TMA is formulated as a no-wait job shop scheduling problem with release times and sequence-dependent setup times to model aircraft safety separations. They implement a local search algorithm, which compares favorably versus a simple control rule sequencing the aircraft at the runways according to the FIFO rule. Results are shown for the MXP and FCO airports in Italy.

Adacher et al. [1] proposed a detailed model for the static ASP problem based on the alternative graph (AG) model of Mascis and Pacciarelli [23], that is a generalization of the disjunctive graph formulation. Differently from previous approaches based on job shop scheduling, the AG enables an accurate representation of the air traffic regulations to be taken into account when solving the ASP. The resulting model enriches the one of Bianco et al. [9] by including additional real-world constraints such as holding circles, time windows of [minimum, maximum] allowed aircraft speeds, multiple capacity of air segments, and blocking constraints at runways. Simple heuristic algorithms are proposed to solve the problem.

Artiouchine et al. [4] introduced another abstract model, where the routing and scheduling problem for landing operations is formulated through the K King problem. According to the authors, this approach is a strong abstraction of landing operations. For example, STI constraints are not accurately modeled, as well as the range of aircraft dynamics.

Bennell et al. [8] reviewed optimization approaches for scheduling aircraft in a TMA. According to the authors, there are only a few attempts to model the ASP including all the TMA resources. Many theoretical studies promise solutions which may not be possible to implement in practice, as some critical operational constraints are ignored or relaxed in the model or the required computational resources are unreasonable. Moreover, landing and takeoff problems are often solved separately, and there may be a very limited forecast of traffic both in time and space. These facts limit the knowledge of the impact of scheduling decisions on the delay propagation to ingoing and outgoing aircraft. In the conclusions, the authors point out that the scheduling algorithms can be of practical use only if they can provide near-optimal solutions within seconds of computation.

From the analysis of the ASP literature, we conclude that detailed models enable a better representation of the practical problem with respect to basic models. However, further research is needed to develop effective and efficient solution algorithms, as well as to find the right compromise between a number of factors, including the degree of detail in the formulation of relevant practical constraints, the effectiveness of solution algorithms, the time available to compute ASP solutions, the level of collaboration among en route and TMA controllers, and the time horizon of the prediction.

Moreover, even if the delay minimization is typically the main indicator of the solution quality, in practice traffic controllers pay attention to other indicators, such as the total time spent by the aircraft in the TMA. All these issues motivates this article.

3. PROBLEM FORMULATION

This section first introduces the AG model and then shows how objective function, variables and constraints of ASP are formulated. Three models are presented for managing traffic in a TMA and an illustrative example follows for the FCO airport.

AGs have been effectively used to model complex real-world job-shop scheduling problems arising in public transportation and manufacturing [14, 23]. In the ASP, an operation is an elementary movement by a specific aircraft, such as the traversing of an air segment, holding circle, or runway. As in the usual definition of the job shop scheduling problem, a job (aircraft) must undertake a prescribed sequence of operations on specific machines (TMA resources).

An AG is a triple $\mathcal{G} = (N, F, A)$, where $N = \{0, 1, \ldots, n-1, n\}$ is the set of nodes, F is a set of directed arcs (fixed), and A is a set of pairs of directed arcs (alternative). In set N, node $i = 1, \ldots, n-1$ is associated to the start time t_i of operation $o_i = o_1, \ldots, o_{n-1}$, while nodes 0/$n$ are associated to the start/end of the schedule (i.e., to the dummy operations o_0 and o_n). Arcs in the set F model the sequence of operations to be executed by a job (the route of a landing/departing aircraft in the TMA). Arcs in the set A model the sequencing decisions. If $((i,j),(h,k)) \in A$, arc (i,j) is the alternative of arc (h,k) and defines an order between two operations. Each arc (i,j), either fixed or alternative, has an associated weight $w_{i,j}$.

A selection S is a set of alternative arcs, at most one from each pair. Given a selection S, $l^S(i,j)$ denotes the length of a longest path from node i to node j in the graph $G(S) = (N, F \cup S)$. A selection, in which exactly one arc is chosen from each pair in A, is a feasible schedule (i.e., a solution to the ASP) if $G(S)$ has no positive weight cycles. In fact, a positive cycle represents a circular precedence between operations, that is, an operation which precedes itself. Given a feasible schedule S, the minimum start time of o_i is $t_i = l^S(0,i)$, for each $i \in N$. A feasible schedule S is optimal if $l^S(0,n)$ is minimum over all the solutions.

The problem of finding an optimal schedule in $\mathcal{G} = (N, F, A)$ can be formulated as a particular disjunctive program, that is, a linear program with logical conditions "and" ($\wedge$, conjunction) and "or" ($\vee$, disjunction):

$$
\begin{array}{ll}
\min t_n - t_0 & \\
\text{s.t.} & \\
t_q - t_p \geq w_{p,q} & (p,q) \in F \\
(t_j - t_i \geq w_{i,j}) \vee (t_k - t_h \geq w_{h,k}) & ((i,j),(h,k)) \in A
\end{array}
$$

Each variable t_i is the start time of o_i, operation o_0 precedes (o_n follows) all the other operations and t_0 (t_n) represents the start (the end) of traffic prediction. Given a fixed arc (p, q), timing is feasible if $t_q \geq t_p + w_{p,q}$. Given an alternative arc $((i,j),(h,k))$, if the arc (i, j) is selected than $t_j \geq t_i + w_{i,j}$ must be satisfied in a feasible schedule, alternatively if the arc (h, k) is selected than $t_k \geq t_h + w_{h,k}$ must be satisfied.

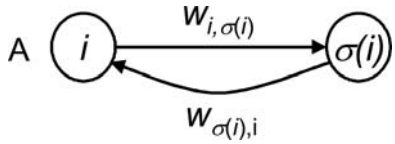

FIG. 1. Alternative graph (AG) formulation of a [min,max] travel time constraint.

3.1. *Modeling the ASP via AGs*

This subsection presents the AG formulation of ASP decision variables, constraints, and objective function. A TMA is composed of various types of resources: runways, air segments, and holding circles. We next illustrate the formulation of each type of resource. Given an operation o_i, we denote with $O_{\sigma(i)}$, the operation which follows o_i on the same job.

The traversing time of an air segment by an aircraft can range in an interval $[\phi, \gamma]$ of minimum and maximum travel time. In Figure 1, we represent aircraft A traveling in an air segment. The travel time constraint is modeled with a pair of fixed arcs $(i, \sigma(i))$ and $(\sigma(i), i)$. Arc $(i, \sigma(i))$ is weighted with $w_{i,\sigma(i)} = \phi$ while arc $(\sigma(i), i)$ is weighted with $w_{\sigma(i),i} = -\gamma$. These constraints impose that the entrance time in the next air segment $\sigma(i)$ must be $t_i + \phi \leq t_{\sigma(i)} \leq t_i + \gamma$. We observe that when $w_{i,\sigma(i)} = -w_{\sigma(i),i}$ the travel time constraints correspond to a strict no-wait constraint, as in the ASP model of [9].

Air traffic regulations impose that all aircraft flying on the same air segment must respect a minimum longitudinal separation depending on their category (heavy, medium, or light). This separation translates into an STI between the entrance/exit of consecutive aircraft in the same air segment. Overtaking is not allowed within an air segment. In this article, the minimum separation time between consecutive aircraft in the same air segment is modeled as a sequence-dependent setup time [2, 9]. We also consider the special case in which two parallel air segments are very close to each other (e.g., a common glide path), and the minimum longitudinal distance between aircraft is not sufficient to satisfy traffic regulations. An additional minimum diagonal distance is, therefore, imposed between aircraft flying on the parallel air segments. This traffic situation is modeled as a single air segment resource in which setup times do not depend on the aircraft sequence only but on the route chosen for each aircraft also.

Figure 2a and b shows how we model the sequencing decisions for two aircrafts on an air segment. In the example, two nodes represent the entrance (i) of aircraft A in an air segment and its exit, that is, the entrance ($\sigma(i)$) in the subsequent air

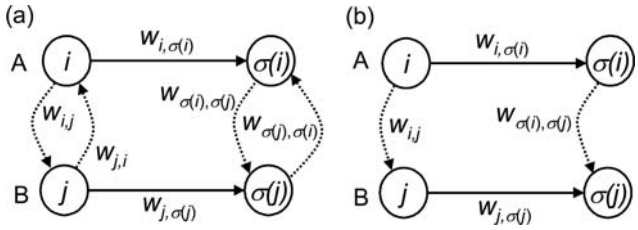

FIG. 2. AG formulation of an air segment.

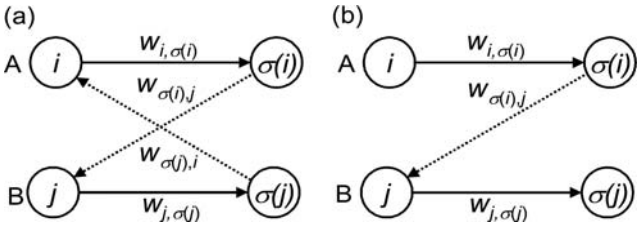

FIG. 3. AG formulation of a runway.

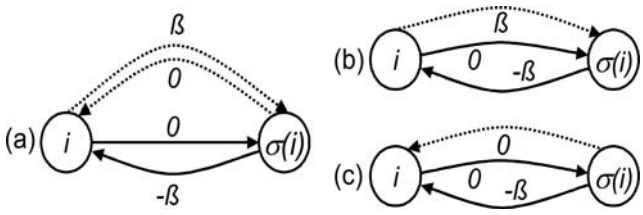

FIG. 4. AG formulation of a holding circle.

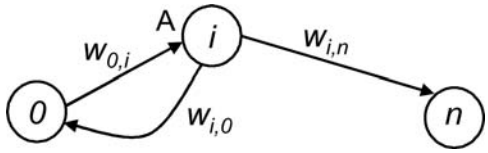

FIG. 5. AG formulation of a release, due date, and deadline constraints.

segment. Similarly, nodes j and $\sigma(j)$ are used for aircraft B. The no overtaking constraint and the minimum separation distance between aircraft is obtained by adding two alternative pairs $((i,j),(\sigma(j),\sigma(i)))$ and $((j,i),(\sigma(i),\sigma(j)))$. The weights on the alternative arcs correspond to the minimum STI between A and B in the air segment. The only two ways to select alternative arcs from the two pairs without generating positive weight cycles in the graph are either to select both arcs (i,j) and $(\sigma(i),\sigma(j))$ (i.e., aircraft A precedes B at both the entrance and exit of the air segment), or to select both arcs (j,i) and $(\sigma(j),\sigma(i))$ (i.e., B precedes A at both locations).

In Figure 2b, A is before B and the minimum STI at the entrance will be $w_{i,j}$ while the minimum STI at the exit will be $w_{\sigma(i),\sigma(j)}$. The aircraft sequencing on a runway is formulated by an alternative pair for each pair of aircraft landing or taking off from that runway. The runway is modeled as a blocking resource [18], since the presence of one aircraft on the runway imposes that no other aircraft can use it. If there are intersecting runways, no two aircrafts can use them simultaneously, that is, intersecting runways behave as a single blocking resource.

In Figure 3a, two aircrafts, A and B, require the same runway. Nodes i and j represent A and B entering the runway, while nodes $\sigma(i)$ and $\sigma(j)$ represent A and B leaving the runway (i.e., entering the subsequent resource). The weights of the fixed arcs $w_{i,\sigma(i)}$ and $w_{j,\sigma(j)}$ model the time to process the runway by aircrafts A and B. The alternative pair $((\sigma(i),j),(\sigma(j),i))$ is used to model the sequencing decision between A and B. If A precedes B then the alternative arc $(\sigma(i),j)$ is selected, otherwise $(\sigma(j),i)$ is selected (and B precedes A on the runway). The safety separation constrains a first aircraft leaving the runway and a second aircraft entering the runway are represented by the weights $w_{\sigma(i),j}$ (if A precedes B) and $w_{\sigma(j),i}$ (if B precedes A). In Figure 3b, A precedes B and B can enter the runway only after $t_j \geq t_i + w_{i,\sigma(i)} + w_{\sigma(i),j}$.

Holding circles are used by traffic controllers to let landing aircraft wait in flight when the TMA is congested. When an aircraft enters the airborne holding in response to a disturbed traffic situation, it must fly at a given speed and can leave the holding circle only after the traversing of (a multiple of) half length of the circle. We model the holding circle decisions via alternative pairs and the holding circle constraints via fixed arcs in the graph. In Figure 4a and b, the possible entrance of a landing aircraft in a holding resource is modeled. Two possible decisions are modeled: (i) the aircraft travel a half circle of weight β or (ii) the aircraft skip the holding circle.

In Figure 4a, the two alternative decisions are formulated with a pair of alternative arcs $((i,\sigma(i)),(\sigma(i),i))$, weighted β and 0. The holding circle constraints are modeled by the two fixed arcs $(i,\sigma(i))$ and $(\sigma(i),i)$ with weights 0 and $-\beta$. In a solution, one of the two alternative arcs must be chosen. Figure 4b shows the solution (i) in which the arc $(i,\sigma(i))$ is selected. This forces the aircraft to spend exactly β time units in the holding circle (since it causes a zero weight cycle with the fixed arc $(\sigma(i),i)$, thus imposing $t_{\sigma(i)} = t_i + \beta$). Figure 4c shows the solution (ii) in which the arc $(\sigma(i),i)$ is selected. This forces the aircraft to skip the holding cycle (since it causes a zero weight cycle with the fixed arc $(i,\sigma(i))$, thus imposing $t_{\sigma(i)} = t_i$). A description of how to extend our model to multiple circles can be found in Pistelli et al. [25], in which the AG formulation of a given number of half circles is shown, with suitable arc weights. In this work, we performed experiments by considering multiple (half) circles at the airborne holding resources.

Figure 5 shows release, due date, and deadline arcs. The release time is the minimum time at which the aircraft must enter the TMA. The fixed arc $(0,i)$ models the release time r_i of o_i with weight $w_{0,i} = r_i$. The deadline time is the maximum time at which the aircraft can enter the TMA. The fixed arc $(i,0)$ models the deadline time $\bar{d}_i$ of o_i with weight $w_{i,0} = -\bar{d}_i$. If o_i is late (i.e., $l^S_{0,i} + w_{i,0} > 0$), there is a positive cycle in the graph and the resulting schedule is infeasible.

Arc (i,n) in Figure 5 is used to model the objective function of the ASP. The due date time d_i is the scheduled time

of an aircraft on a specific resource, for example, the landing/takeoff time shown in the timetable. The fixed arc (i, n), with weight $w_{i,n} = -d_i$, is a due date arc that we use to measure the delay of the associated aircraft at the starting of operation i. If in a feasible schedule $t_i > d_i$, then the delay of the solution must be $t_n \geq t_i - d_i$. Therefore, given a feasible schedule S and since $t_i = l^S_{0,i}$, we have that $l^S_{0,n}$ corresponds to the maximum delay associated to S. The choice of the due date d_i allows to compute several components of the aircraft delay. Specifically, let α_i be the scheduled starting time of o_i in the timetable, and let τ_i be its earliest possible starting time compatible with the position of the associated aircraft at the beginning of traffic prediction, and disregarding the presence of other aircraft. If $\tau_i > \alpha_i$, the difference $\tau_i - \alpha_i$ is an unavoidable delay that cannot be recovered by rescheduling the aircraft.

Given a feasible schedule S such that $t_i > \alpha_i$, the quantity $t_i - \alpha_i$ denotes the total delay of o_i in S, and $t_i - \max\{\tau_i, \alpha_i\}$ denotes the maximum consecutive delay of o_i in S, since the unavoidable delay $\tau_i - \alpha_i$ is omitted [14]. Hence, if we choose a due date $d_i = \alpha_i$ for each operation with a scheduled starting time, then $l^S_{0,n}$ is the maximum total delay of S. Similarly, if we choose $d_i = \max\{\alpha_i, \tau_i\}$ for each operation with a scheduled starting time, then $l^S_{0,n}$ is the maximum consecutive delay of S. In this article, we minimize the latter quantity. The ASP objective is thus the minimization of the maximum consecutive delay over all aircraft.

For takeoff operations, we follow the procedure commonly adopted by air traffic controllers that consider a time window for takeoff between 5 min before and 10 min after the scheduled takeoff time (STT) [8]. Therefore, a departing aircraft is late if leaving the airport after 10 min from its STT, that is, the value α associated to the takeoff operation is STT plus 10 min. For each landing aircraft, we consider a value α equal to its scheduled landing time.

3.2. AG models of the ASP

We next describe three models for the collaborative and noncollaborative ASP.

Model 1 is a noncollaborative model in which the entrance time of each aircraft in the TMA is a fixed value r_i received in input from neighboring areas. The entrance time constraint is modeled by adding a release arc with weight r_i and a deadline arc with weight $-r_i$ to the ASP model for each aircraft.

Model 2 is a collaborative model in which the entrance time of each aircraft in the TMA can be defined in cooperation with the traffic controllers of neighboring areas. An optimal solution for this model will fix the "best" entrance times, that is, such that the maximum consecutive delay of the aircraft schedule is minimized. Clearly, delaying the entrance of a landing aircraft in the TMA has an impact on neighboring areas. To limit this impact, a late entrance is penalized in the objective function. This model is also a simplification of Model 1, since the holding circle decisions are not included. In other words, the entrance in the TMA is located at the exit from the holding circles, so that the time spent in a holding circle is penalized as a late entrance.

Model 3 is also a collaborative model. The difference with model 2 is that the entrance in the TMA is located at the entrance in the holding circles, that is, holding circle decisions are now included in the ASP model and only the delay at the entrance of the holding circles is penalized in the model. In other words, with this model a late entrance better corresponds to the impact on neighboring areas than in model 2.

We observe that, deadline arcs increase the complexity of model 1 significantly with respect to models 2 and 3, since the ASP formulation is severely constrained. Conversely, to implement in practice the solutions provided by the latter models, a larger degree of flexibility is necessary to set the entrance time of each aircraft in the TMA. Changing the aircraft entrance time requires not only a coordination action between the controllers of the TMA and the controllers of neighboring areas. The change is physically possible only if the schedules are found sufficiently in advance with respect to the start time t_0 of traffic prediction, to let enough time to the en route controllers to delay the approaching time of incoming aircraft according to the ASP solution.

3.3. Illustrative Example of the FCO Airport

Figure 6 shows a traffic situation with two landing aircrafts (A and B) and one departing aircraft (C). At the FCO airport, there are three runways (16L, 16R, and 25), two of which (16R and 25) are intersecting and cannot be used simultaneously. The TMA of this airport is divided into three holding circles [Tarquinia (TAQ), Campagnano (CMP), Ciampino (CIA); numbered from 1 to 3], seven air segments for landing procedures (4–10), three air segments for takeoff procedures (14–16) leading to the exit points of the TMA (ELVIN, RAVAL, and BOL), two runways (12 and 13), and a common glide path (11).

In the traffic situation of Figure 6, there are potential conflicts at two air segments (4 and 11) and at one runway (12), since these resources are used by more than one aircraft.

For this example situation, Figures 7–9 present the AG formulation of the three models of section 3.2. Each node of the graph represents an operation, for example, $A9$ is aircraft A entering air segment 9. Fixed (alternative) arcs are depicted with solid (dotted) arrows. For the sake of simplicity, the weight of fixed and alternative arcs is not shown in the figures. For a detailed description of all arc weights, we refer the reader to section 3.1.

The departing aircraft C is constrained to enter the runway after its minimum departure time, which is formulated with a release arc $(0, C12)$. The landing aircraft A and B are constrained to enter the TMA after their minimum departure time.

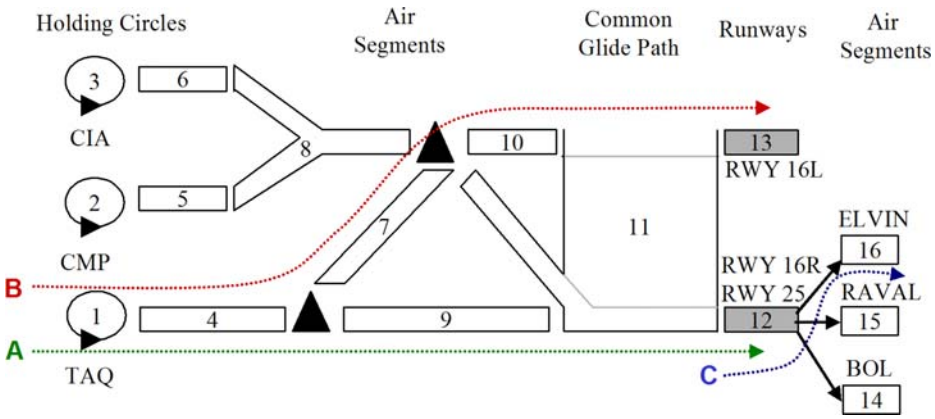

FIG. 6. A traffic situation of the FCO airport with landing (A,B) and departing (C) aircraft. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

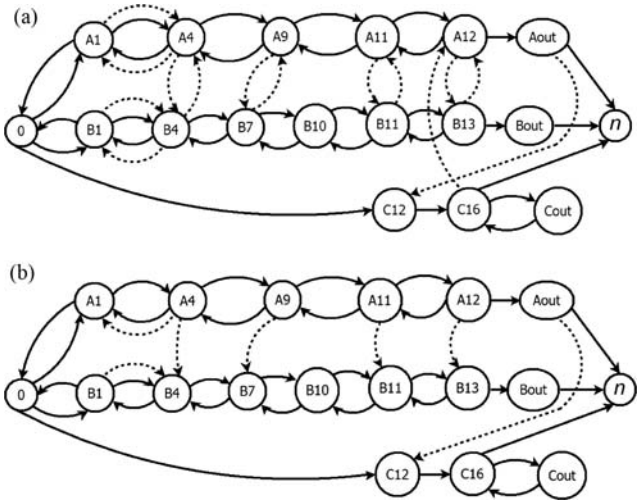

FIG. 7. The Model 1 formulation (a) and solution (b) for the example in Figure 6.

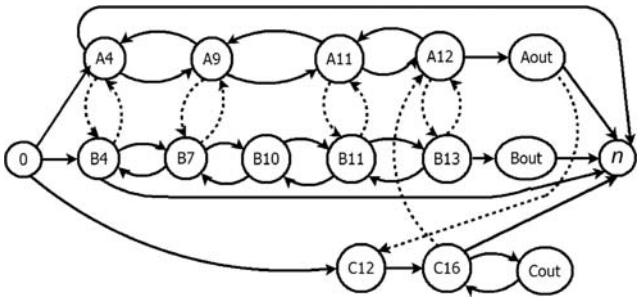

FIG. 8. The Model 2 formulation for the example in Figure 6.

Two holding circle decisions are modeled for each landing aircraft: travel or not half circle of weight β. Two alternative arcs model these decisions: $((A1, A4), (A4, A1))$ and $((B1, B4), (B4, B1))$.

Each aircraft must traverse each air segment along its route within a window of [minimum, maximum] traversing times, for example, arcs $(A9, A11)$ and $(A11, A9)$ for the air segment 9. The potential conflicts on air segments are modeled by four alternative pairs: $((A4, B4), (B7, A9))$ and $((B4, A4), (A9, B7))$ for air segment 4, $((A11, B11), (B13, A12))$ and $((B11, A11), (A12, B13))$ for the common glide path 11.

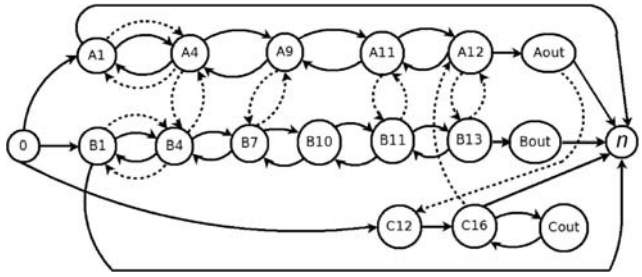

FIG. 9. The Model 3 formulation for the example in Figure 6.

Aircraft sequencing decisions on runway 12 are modeled by the pair $((C16, A12), (A\text{out}, C12))$. To minimize the maximum consecutive delay on the runways, the due date arcs $(A\text{out}, n)$, $(B\text{out}, n)$ and $(C16, n)$ are weighted with $w_{(A\text{out},n)} = -\max\{\tau_{A\text{out}}, \alpha_{A\text{out}}\}$, $w_{(B\text{out},n)} = -\max\{\tau_{B\text{out}}, \alpha_{B\text{out}}\}$ and $w_{(C16,n)} = -\max\{\tau_{C16}, \alpha_{C16}\}$, where α and τ are defined in section 3.1.

Figure 7a and b shows model 1, in which the entrance time of each landing aircraft is taken as a constraint for the problem. This formulation is shown in Figure 7a, with release and deadline arcs at the first operation of landing aircraft in the TMA: for aircraft A the arcs $(0, A1)$ and $(A1, 0)$ with weight $w_{0,A1}$ and $-w_{0,A1}$; for B $(0, B1)$ and $(B1, 0)$ with weight $w_{0,B1}$ and $-w_{0,B1}$. With these constraints, the entrance time of the two aircrafts must be equal to $w_{0,A1}$ and $w_{0,B1}$.

Figure 7b shows a feasible schedule for model 1. This solution is obtained by scheduling first aircraft A on the air segment 4 [selecting the arcs $(A4, B4)$ and $(A9, B7)$], on the common glide path 11 [selecting the arcs $(A11, B11)$ and $(A12, B13)$], and on the runway 12 [selecting the arc $(A\text{out}, C12)$]. Aircraft A skips the holding circle [selecting the arc $(A4, A1)$ with weight 0]. Aircraft B runs a half circle of weight β [selecting the arc $(B1, B4)$] to avoid the potential conflict with aircraft A on the air segment 4 that would cause a positive cycle in the graph due to the deadline constraint $(B1, 0)$.

In model 2, holding circle and deadline constraints are replaced with respect to model 1 by the due date arcs $(A4, n)$ and $(B4, n)$ with weights $w_{(A4,n)} = -\max\{\tau_{A4}, \alpha_{A4}\}$ and $w_{(B4,n)} = -\max\{\tau_{B4}, \alpha_{B4}\}$ (Fig. 8). As a result, each aircraft can spend an arbitrary amount of time in the holding circle. However, the time spent in the holding circle is penalized in the objective function. In other words, with model 2, the delay of landing aircraft is computed both at the entry fix and at the runway, while the delay of departing aircraft is only computed at the runway.

In model 3, the holding circle decisions are modeled as for model 1. Differently from model 1, deadline constraints are replaced by the due date arcs $(A1, n)$ and $(B1, n)$ with weights $w_{(A1,n)} = -\max\{\tau_{A1}, \alpha_{A1}\}$ and $w_{(B1,n)} = -\max\{\tau_{B1}, \alpha_{B1}\}$ (Fig. 9). With this formulation, the delay with respect to the minimum entrance time in the TMA is penalized in the objective function. The aircraft delay of landing and departing aircraft is computed as for model 2.

4. SCHEDULING ALGORITHMS

This section presents the scheduling algorithms used to solve the ASP. Note that even the restricted version of the ASP with three air segments and only landing aircraft is strongly Nondetermistic Polynomial-time (NP)-hard since it includes the no-wait flow shop problem with three machines as a special case [24]. Therefore, in the following, we focus on heuristic and BB algorithms. We first describe a constraint propagation technique to speedup the computation of feasible schedules and then describe five scheduling algorithms, ranging from simple dispatching rules to advanced exact approaches.

4.1. Static Implications

In the development of our algorithms, we extend to the ASP the concept of static implications introduced in [14] for blocking resources. A static implication occurs when, given two unselected pair of alternative arcs $((i,j),(h,k))$ and $((a,b),(c,d))$, it is possible to prove that the selection of (i,j) and (a,b) is infeasible. Therefore, in a feasible schedule the selection of (i,j) implies the selection of (c,d). In [14], static implications are shown to be very effective for train scheduling problems. The following proposition extends the concept of static implications to air segments (see the AG formulation of two aircrafts traversing an air segment in Fig. 2).

Proposition 4.1 *Consider two unselected alternative pairs for an air segment* $((j,i),(\sigma(i),\sigma(j)))$ *and* $((i,j),(\sigma(j),\sigma(i)))$ *. If* $w_{i,j} + w_{j,i} > 0$ *then arc* $(\sigma(i),\sigma(j))$ *is implied by the selection of (i, j) and arc* $(\sigma(j),\sigma(i))$ *is implied by the selection of (j, i). If* $w_{\sigma(i),\sigma(j)} + w_{\sigma(j),\sigma(i)} > 0$ *then arc (i, j) is implied by the selection of* $(\sigma(i),\sigma(j))$ *and arc (j, i) is implied by the selection of* $(\sigma(j),\sigma(i))$.

Proof. 1 The result follows from the observation that the selections $\{(i,j),(j,i)\}$ and $\{(\sigma(i),\sigma(j)),\sigma(j),\sigma(i)\}$ contain a positive weight cycle and are, therefore, infeasible. ∎

Algorithm AGH

begin
Set $S := \emptyset$
while $A \neq \emptyset$ **do**
  begin
  Choose a pair $((i,j),(h,k)) \in A$ according to a pre-defined evaluation criterion,
  Let $S' := S \cup \{(i,j)\}$, $A' := A \setminus \{((i,j),(h,k))\}$,
  $S' := S' \cup Stat(i,j)$, remove from A' the pairs associated to the arcs in $Stat(i,j)$,
  if (there is a cycle in $G(S')$) **or** (an arc from $Stat(i,j)$ is forbidden) **then**
    begin
    Let $S' := S \cup \{(h,k)\}$, $A' := A \setminus \{((i,j),(h,k))\}$,
    $S' := S' \cup Stat(h,k)$, remove from A' the pairs associated to the arcs in $Stat(h,k)$,
    if (there is a cycle in $G(S')$) **or** (an arc from $Stat(h,k)$ is forbidden) **then**
      begin
      The procedure fails in computing a feasible schedule,
      STOP
      end
    end
  Let $S := S', A := A'$.
  end
end

FIG. 10. Sketch of the AGH heuristics

The intuition behind Proposition 4.1 is that aircraft overtaking is not allowed in the proximity of the airport. Therefore, once a corridor is detected all the alternative pairs between two jobs passing in the corridor are implied and the sequencing decision of an aircraft preceding another aircraft can be propagated to all their air segments if the two aircrafts follow the same route. This constraint propagation technique is used during the execution of the algorithms to speedup the search process.

4.2. Practical Scheduling Rule

The FIFO rule solves the aircraft conflicts one at a time by assigning each conflicting resource to the first aircraft requiring it. This local rule requires no look-ahead control action since aircraft run in the airport area on the basis of their actual order of arrival at each air segment or runway. According to [8], the FIFO rule is close to the ATC control practice, even if human controllers may adjust the FIFO sequence to recover possible infeasibilities.

4.3. Greedy Heuristics Based on Arc Selection

The arc greedy heuristic (AGH) is a family of heuristic algorithms, which takes decisions based on global information available from the AG formulations [26]. A sketch of the general structure of these algorithms is given in Figure 10.

The family is based on the idea of repeatedly enlarging a selection by choosing an unselected pair at a time from set A

and by selecting one of the two arcs until a feasible schedule is found or an infeasibility (i.e., positive weight cycle in the graph) is detected. The heuristics in the AGH family differ from each other for the evaluation criteria used to choose the next unselected arc from A. Every time an alternative arc (i, j) is selected, all alternative arcs implied by Propositions 4.1 are also selected. This set is called Stat(i, j) in the algorithm sketched in Figure 10. If for a given selection S, there is an arc $(a, b) \in$ Stat(i, j) such that $((a, b), (c, d)) \in A$ and $(c, d) \in S$, we say that (a, b) is forbidden, since the addition of (i, j) to S leads to an infeasible solution.

Several evaluation criteria have been introduced in [26]. Here we present only two evaluation criteria, which from preliminary tests turn out to be the best performing for this problem. The avoid most critical completion time (AMCC) criterion chooses an unselected alternative pair $((i,j),(h,k))$ maximizing the quantity $\max\{l^S(0,i)+w_{i,j}+l^S(j,n), l^S(0,h)+w_{h,k}+l^S(k,n)\}$ and selects the arc causing the minimum delay [23]. In other words, this criterion tries to avoid the arc that would cause the largest increase of $l^{S'}(0,n)$ in $G(S')$. The avoid most similar pair (AMSP) criterion chooses an unselected alternative pair $((i,j),(h,k))$ maximizing the quantity $l^S(0,i)+w_{i,j}+l^S(j,n)+l^S(0,h)+w_{h,k}+l^S(k,n)$ and selects the arc causing the minimum delay [26]. In other words, this criterion chooses a pair with the largest sum of consecutive delays and selects the arc causing the smallest delay.

4.4. Algorithms Based on Job Selection

The job greedy heuristic (JGH) is a new family of constructive algorithms which combines different AGH algorithms. JGH iteratively extends a partial schedule until a complete graph selection is found or an infeasibility is detected. The decision taken at each step consists in the insertion of a complete job in the current partial schedule, keeping fixed the sequence of the previously scheduled jobs. A sketch of the general JGH structure is given in Figure 11, in which we use the following notation. We use J to denote the set of jobs, j_i to denote the set of nodes associated to the operations of the ith job in J, $\bar{N}$ to denote the set of nodes of scheduled jobs. Finally, y denotes the number of AGH criteria.

As shown in Figure 11, at each iteration, all the unscheduled jobs are evaluated for insertion as candidate using y AGH criteria and the best criterion for each job $i \in U$ is stored with the associated selection S^i . The best candidate job "bestjob" is inserted in the current partial schedule S according to the associated selection $S^{bestjob}$. Note that, at each iteration, the relative order among the previously selected jobs remains fixed. The procedure is repeated for at most $|U|$ times until a feasible schedule S is obtained or infeasibility is detected.

4.5. BB Algorithm

BB algorithms implicitly explore the whole set of solutions to find a globally optimal schedule. To speedup the search, lower bound and implication techniques are applied.

Algorithm JGH

begin
Set $\bar{N} := \emptyset$ (set of nodes of scheduled jobs)
Set $U := J$ (set of unscheduled jobs)
Set $S := \emptyset$ (current selection)
while $U \neq \emptyset$ **do**
  begin
  $bestjob := null, f(S^{bestjob}) = \infty$,
  forall $i \in U$ **do**
    begin
    Let $\bar{A} := \{((a,b),(c,d)) \in A : (a \in \bar{N} \wedge b \in j_i) \vee (b \in \bar{N} \wedge a \in j_i)\}$
    $f(S^i) = \infty$, (objective for the best insertion of job j_i)
    for $h = 1, \ldots, y$ **do**
      begin
      Starting from the alternative graph $\mathcal{G}(S) = (N, F \cup S, \bar{A})$,
      Compute a schedule by repeatedly applying the h-th AGH criterion,
      Denote with S^i_h the resulting selection,
      if $f(S^i) > l^{S \cup S^i_h}(0,n)$ **then** $f(S^i) = l^{S \cup S^i_h}(0,n)$ and $S^i = S^i_h$, (choose best AGH)
      end
    if $f(S^i) < f(S^{bestjob})$ **then** $bestjob := i$, $S^{bestjob} = S^i$,
    end
  if $bestjob = null$ **then** no feasible schedule found, STOP.
  else $\bar{N} = \bar{N} \cup j_{bestjob}$, $U := U \setminus \{bestjob\}$, $S := S \cup S^{bestjob}$
  (insert $bestjob$ in the schedule)
  end
end

FIG. 11. Sketch of the JGH heuristic.

Whenever the lower bound on the objective function for a subset of all solutions becomes larger than or equal to a known upper bound on the global optimum, the whole subset can be pruned. Implication techniques allow to enlarge a partial selection S whenever it can be proved that for some unselected pair of alternative arcs $((i,j),(h,k))$ the selection $S \cup (i,j)$ is infeasible. In this case, (h,k) must be selected and we say that S implies (h,k) and, from proposition 4.1, also the set Stat(h, k). When all the search space is explored, the BB algorithm stops returning the proven optimum. If the procedure is terminated before completing the search, the procedure returns a lower bound and an upper bound on the global optimum.

The exact algorithms we consider are adaptations of the BB algorithm described in [14], developed for a train scheduling problem and now modified to solve the ASP. In the following, we limit ourselves to briefly describe the main components of the algorithm and the modification with respect to the algorithm described in [14].

The initial upper bound is obtained by running the four heuristics proposed above (FIFO, AMCC, AMSP, and JGH). At each node of the enumeration tree (associated to a selection S), the BB algorithm computes a lower bound with an adaptation of the single machine Jackson preemptive schedule JPS(S) described in [20]. The JPS is computed for each air

segment and each runway. Implication techniques described in [14] are also used to speedup the computation, even if static implications are adapted to deal with air segment resources using proposition 4.1. The search strategy we use alternates four repetitions of the depth-first visit with the choice of a selection S with the smallest value of JPS(S) among the last five generated selections (i.e., best-first visit limited to the last selections enumerated). The choice of this hybrid strategy is motivated by the need for searching new feasible solutions (using the depth-first strategy) and promising selections (using the best-first strategy). We use a binary branching scheme to choose an unselected pair $((i,j),(h,k))$ for branching. Specifically, for each model, we use the best branching strategy resulting from the set of experiments reported in [12]: with models 2 and 3, we branch with priority on alternative pairs associated to runway resources, with model 1 all alternative pairs have the same priority. The first strategy is motivated by the fact that the runways are often the bottleneck resources of the TMA. For both strategies, we use the AMSP criterion to choose the next unselected pair for branching.

5. COMPUTATIONAL EXPERIMENTS

The tested instances are based on observations during a busy hour of national and international aircraft arrivals and departures at the FCO airport. We consider an entry fix for each aircraft in the TMA. We also assume that the routing problem is solved off-line, so that the workload of the runways is well balanced, and there is no conflict at runways when aircraft are on time. In our models, we fix a maximum of three round trips for each aircraft in a holding circle. The experiments are executed on a processor Intel i7 (2.84GHz), 8 GB Ram and Linux operating system. The AGLibrary, developed by the Operations Research Group of Roma Tre University, is used to implement and test the AG models and algorithms.

This section is organized as follows. We first describe our set of instances. We then compare the BB algorithm with the heuristic approaches in terms of feasibility rate, computation time, and solution quality. Finally, we study the different AG models of the ASP via the BB algorithm in terms of objective function value and other performance indicators, and explore its computational limit for large time horizons of traffic prediction. We also discuss the results and possible improvements.

5.1. *Description of the ASP Instances*

Table 1 presents the main characteristics of 240 instances we use to test the scheduling algorithms. For each model of section 3, we generate 80 disturbed traffic conditions based on random entrance delays for the landing and departing aircraft. Each row of Table 1 reports average data over 20 instances for a given model (column 1). Column 2 gives four time horizons of traffic prediction, ranging from 15 to 60 min. Column 3 shows the number of arriving/departing aircraft. The resulting graphs are described in columns 4–6 in terms of the number of nodes ($|N|$), the fixed arcs ($|F|$), and the pairs of alternative arcs ($|A|$).

In Table 1, we introduce four time horizons of traffic prediction. The two shortest time horizons (15 and 30 min) are interesting for real-time traffic control, since landing aircraft enter the landing planner's radar range around 30 min before touch down, as reported by Bennell et al. [8]. The two longest time horizons (45 and 60 min) are studied to evaluate the computation time and solution quality of our exact algorithm for challenging instances. For each time horizon of traffic prediction, we generate 10 Uniform and 10 Gaussian (randomly generated) entrance delays in the TMA. Columns 7–8 of Table 1 report on the perturbation characteristics as the maximum and average entrance delays (in seconds) and Column 9 shows the number of aircraft delayed at their entrance in the area under study, before solving the ASP problem (see, e.g., Fig. 6a). We only delay aircraft that enter the network in the first half of each time horizon in order to study the delay propagation in the second half of the time horizon.

Models 1 and 3 have the same number of fixed arcs, since each landing aircraft has one deadline arc and one due date arc in model 1 while each landing aircraft has two due date arcs in model 3. The number of nodes and alternative arcs is also the same, since models 1 and 3 have the same number of decision aircraft timing and ordering variables. Differently from models 1 and 3, model 2 has less nodes, fixed and alternative arcs

TABLE 1. Perturbed traffic situations and prediction horizons.

Graph model	Time horiz.	Arr/Dep aircraft	$\|N\|$	$\|F\|$	$\|A\|$	Max delay	Avg. delay	Delayed aircraft
1	15	6/1	44	290	103	450	170.4	4
1	30	16/4	118	1,605	688	900	234.7	10
1	45	22/7	166	2,906	1,292	1,350	345.6	14
1	60	32/16	259	6,425	2,963	1,800	465.1	24
2	15	6/1	38	218	67	450	170.4	4
2	30	16/4	102	1,413	592	900	234.7	10
2	45	22/7	144	2,642	1,160	1,350	345.6	14
2	60	32/16	227	6,041	2,771	1,800	465.1	24
3	15	6/1	44	290	103	450	170.4	4
3	30	16/4	118	1,605	688	900	234.7	10
3	45	22/7	166	2,906	1,292	1,350	345.6	14
3	60	32/16	259	6,425	2,963	1,800	465.1	24

TABLE 2. Model 2: FIFO versus BB for each time horizon.

Time horiz (min)	Sched. algo.	Comp. Time	Feas. Sched.	Opt. solut.	Max. delay	Avg. delay	Total DTTS
15	BB	0.01	20	20	70.9	10.6	169
15	FIFO	0.01	20	17	74.4	9.9	151
30	BB	0.17	20	20	303.9	91.9	1,130
30	FIFO	0.02	18	0	584.8	178.3	1,431
45	BB	0.96	20	20	265.6	78.5	1,531
45	FIFO	0.07	17	0	752.7	254.4	2,181
60	BB	3.23	20	0	1,068.2	362.4	2,673
60	FIFO	0.49	13	0	1,911.1	872.5	2,918

TABLE 3. Model 3: FIFO versus BB for each time horizon.

Time horiz. (min)	Sched. algo.	Comp. time	Feas. sched.	Opt. solut	Max. delay	Avg. delay	Total DTTS
15	BB	0.01	20	20	70.9	10.2	182
15	FIFO	0.01	20	16	73.0	7.6	181
30	BB	0.29	20	20	303.2	75.1	1,730
30	FIFO	0.02	18	0	583.7	96.5	4,378
45	BB	1.18	20	20	264.9	64.9	2,294
45	FIFO	0.08	17	0	752.7	137.2	8,156
60	BB	0.98	20	0	1,085.1	313.7	5,358
60	FIFO	0.47	12	0	1,918.3	447.7	37,306

since the fixed and alternative arcs of the holding circle constraints are not included in the corresponding graph. The three models are tested on the same entrance delay configurations.

5.2. *BB Versus FIFO*

Tables 2 and 3 report on the average results of BB and FIFO for models 2 and 3, respectively. We omit the results for model 1 due to the low feasibility rate of all heuristics, more detailed information can be found in [12].

For both tables, Column 1 shows the time horizon of traffic prediction, Column 2 the scheduling algorithm. Columns 3–5 present the average results obtained for all instances of each time horizon (20 instances per row): the computation time (in seconds) in Column 3, the number of feasible schedules in Column 4 and the number of optimal solutions (proven optimal by BB within 120s) in Column 5. For BB, the computation time refers to the time to compute its best solution (excluding the computation time of the initial heuristics). To obtain a fair comparison of the computational results, the values of the last three columns (columns 6–8) show the average on the instances for which a solution is found by FIFO. Columns 6 shows the maximum consecutive delays of the solutions (in seconds), which is the objective function of the ASP. Since practitioners are also interested in comparing alternative solutions on the basis of different indicators, Columns 7 and 8 show the average consecutive delays and the total increase of travel time spent by all aircraft in the TMA besides their minimum traversing time (delta travel time spent, or DTTS), both in seconds. DTTS is computed as follows. For an aircraft a landing at a runway r, let $t_{a\,r}$ and $t_{a\,e}$ denote the actual landing time at r and the actual entrance time in the TMA, respectively. Let also $\tau_{a\,r}$ and $\tau_{a\,e}$ denote the minimum landing time at r and the minimum entrance time in the TMA, computed disregarding the presence of the other aircraft. Then, the DTTS of aircraft a is $(t_{ar}-t_{ae})-(\tau_{ar}-\tau_{ae})$. The computation is similar for departing aircraft. The DTTS indicator is interesting for energy consumption reasons, since the smaller is the DTTS of aircraft in the TMA, the smaller is the overall energy consumption.

From the computational results of Tables 2 and 3, we conclude that both algorithms are very fast. However, FIFO often fails in finding a feasible schedule. The DTTS highlights a different behavior of FIFO with models 2 and 3, since with model 3, the aircraft spend much more time in the holding circles with respect to BB, especially for the largest time horizons.

For the 15-min time horizon, a small trade-off exists between minimizing the maximum consecutive delay and the other performance indicators. For time horizons of 30, 45, and 60 min, BB outperforms FIFO in terms of all performance indicators. These results thus demonstrate the potential benefit of introducing optimization-based DSSs to support the activity of air traffic controllers.

5.3. *BB Versus the AG Heuristics*

We next compare the BB algorithm with the arc greedy and JGH approaches in terms of feasibility rate, computation time, and solution quality. These results are presented in Tables 4 and 5 for models 2 and 3, respectively. In the first five columns, we show the same kind of information reported in Tables 2 and 3. To obtain a fair comparison of the computational results, the values of the last three columns report the average over those instances for which a solution is found by all the three heuristics (i.e., over the instances

TABLE 4. Model 2: AG heuristics versus BB for each time horizon.

Time horiz. (min)	Sched. algo.	Comp. time	Feas. sched.	Opt. solut.	Max. delay	Avg. delay	Total DTTS
15	BB	0.01	20	20	70.9	10.6	169
15	AMCC	0.01	20	20	70.9	10.6	169
15	AMSP	0.01	20	18	76.3	13.9	181
15	JGH	0.01	20	18	71.5	11.2	179
30	BB	0.17	20	20	302.9	92.1	1,128
30	AMCC	0.01	20	4	364.1	119.6	1,241
30	AMSP	0.01	20	1	426.1	144.9	1,223
30	JGH	0.01	20	0	368.7	118.6	1,204
45	BB	0.96	20	20	257.9	76.1	1,537
45	AMCC	0.01	20	3	314.8	96.8	1,709
45	AMSP	0.01	20	1	339.4	102.6	1,697
45	JGH	0.04	20	1	316.6	104.9	1,739
60	BB	3.23	20	0	1,086.4	396.5	2,755
60	AMCC	0.07	19	0	1,131.1	354.9	2,781
60	AMSP	0.08	20	0	1,130.9	406.0	2,871
60	JGH	0.19	20	0	1,093.6	431.5	2,803

TABLE 5. Model 3: AG heuristics versus BB for each time horizon.

Time horiz. (min)	Sched. algo.	Comp. time	Feas. sched.	Opt. solut.	Max. delay	Avg. delay	Total DTTS
15	BB	0.01	20	20	70.9	10.2	182
15	AMCC	0.01	20	20	70.9	10.2	182
15	AMSP	0.01	20	13	72.9	9.6	249
15	JGH	0.01	20	13	72.9	9.6	249
30	BB	0.29	20	20	302.2	74.9	1,745
30	AMCC	0.01	20	5	353.7	69.3	3,141
30	AMSP	0.01	20	3	396.5	74.5	3,274
30	JGH	0.01	20	2	361.5	65.3	3,092
45	BB	1.18	20	20	257.2	63.1	2,238
45	AMCC	0.01	20	2	316.8	62.5	3,591
45	AMSP	0.01	20	2	349.9	66.5	3,832
45	JGH	0.03	20	1	314.1	63.6	3,913
60	BB	0.98	20	0	1,105.2	357.4	6,252
60	AMCC	0.07	20	0	1,150.9	252.8	12,710
60	AMSP	0.07	20	0	1,147.4	275.2	14,078
60	JGH	0.19	18	0	1,105.2	275.0	16,456

successfully solved by AMCC for 60-min traffic predictions in Table 4 and over the instances successfully solved by JGH for 60-min traffic predictions in Table 5).

From the results of Tables 4 and 5, BB outperforms the other algorithms in terms of maximum consecutive delay minimization and total DTTS. As for the heuristics, AMSP is the most robust in terms of feasible schedules found while AMCC and JGH are the most performing in terms of maximum consecutive delay minimization. In particular, AMCC is the most performing for the smallest time horizons, while JGH is the most performing for the largest time horizon.

We observe that minimizing the maximum consecutive delay (the objective function of BB) has also a positive effect on the other two performance indicators, even if BB does not always produce the best values of the secondary performance indicators. For some instances, DTTS is better reduced by BB while the average consecutive delay is not. This is due to the fact that the average consecutive delay is collected for the departing aircraft only if they are late, that is, after 10 min from their scheduled departure time.

5.4. Study of BB Solutions for the Three ASP Models

Table 6 presents the BB results for all instances of each time horizon (column 1) and model (column 2). Columns 3–5 present the maximum consecutive delay at two locations (entry fix and runway) plus their maximum value. Columns 6–8 present the average consecutive delay at two locations (entry fix and runway) plus their average value. Column 9 shows the total DTTS (in seconds).

From the information on Table 6, we can analyze the solution quality of the different models in terms of various performance indicators. In general, model 1 presents the largest consecutive delays and total DTTS, due to the deadline constraints.

Comparing models 2 and 3, we recall that the time spent in the holding circle is penalized at the entrance of the TMA for model 2 while it is not penalized for model 3. As a consequence, the maximum and average consecutive delays at the entrance of the TMA are significantly larger for model 2. Moreover, in model 3, the maximum consecutive delay

TABLE 6. Comparison of the three models for each time horizon.

		Max. cons. delay			Avg. cons. delay			
Time Horiz. (min)	Graph model	Max.	In	Out	Avg.	In	Out	Total DTTS
15	1	117.5	—	117.5	51.4	—	51.4	558
15	2	70.9	27.5	69.5	10.6	6.2	14.4	169
15	3	70.9	21.4	69.5	10.2	4.8	14.9	182
30	1	350.8	—	350.8	125.9	—	125.9	3,102
30	2	302.9	293.1	302.2	92.1	88.3	95.0	1,128
30	3	302.2	172.4	302.2	74.9	49.6	95.1	1,745
45	1	562.8	—	562.8	168.4	—	168.4	5,290
45	2	257.9	231.0	301.1	76.1	63.2	85.7	1,537
45	3	257.2	137.3	257.2	63.1	32.5	86.3	2,238
60	1	1,180.2	—	1,180.2	402.4	—	402.4	10,046
60	2	1,087.3	931	1,087.3	398.3	365.6	420.1	2,778
60	3	1,111.7	649.1	1,111.7	343.7	227.3	421.4	6,085

TABLE 7. Comparison of the three models for each time horizon.

		Time best (BB 120s)		Avg. time tot.	BB solution		Opt solutions	
Time horiz. (min)	Graph model	Avg.	Max.	(BB 120s)	120s	3,600s	120s	3,600s
15	1	0.01	0.01	0.01	117.5∗	117.5∗	20	20
15	2	0.01	0.01	0.01	70.9∗	70.9∗	20	20
15	3	0.01	0.01	0.01	70.9∗	70.9∗	20	20
30	1	41.05	95.28	72.5	350.8	338.6∗	13	20
30	2	0.17	0.66	1.95	302.9∗	302.9∗	20	20
30	3	0.29	1.08	2.69	302.2∗	302.2∗	20	20
45	1	17.23	91.56	114.8	562.8	419.7	2	11
45	2	0.96	9.34	39.48	257.9	257.9∗	16	20
45	3	1.18	10.26	37.59	257.2	257.2∗	16	20
60	1	12.88	88.41	120	1,180.2	1,180.2	0	0
60	2	3.23	50.83	120	1,087.3	1,087.3	0	0
60	3	0.98	2.42	120	1,111.7	1,111.7	0	0

is almost always measured in output (except for the 15-min instances), while in model 2, it may happen that the maximum consecutive delay occurs at the entrance of the TMA (in particular for the 45-min instances). In general, model 2 exhibits a larger average consecutive delay but a smaller DTTS compared to model 3. This is for the same reason above, that is, the time spent by an aircraft in the holding circle increases the entrance delay for model 2 and the DTTS for model 3.

5.5. *Computational Aspects of the BB Algorithm*

This subsection analyzes the optimality gap of BB for different time limits of computation. Table 7 presents the average BB results on all instances of a specific time horizon (column 1) and model (column 2). For BB with a time limit of 120s: columns 3–4 report the average and maximum times to compute the best solution, column 5 the total computation time, column 6 the objective function value, and column 8 the number of proven-optimal solutions. For BB with a time limit of 3,600s, column 7 reports the objective function value and column 9 the number of proven-optimal solutions. An asterisk in the table indicates the all the 20 corresponding instances are solved to optimality within the time limit.

From Table 7, it can be observed that, due to the deadline constraints, model 1 is the most time consuming for the BB algorithm. In fact, for the instances up to 30 min (the time horizon of practical applications), BB proves optimality for all instances within a few seconds for models 2 and 3, while for model 1 only 33 instances out of 40 are solved to optimality within 120s and the time to find the best solution may increase up to 95s. The seven open instances can be solved to optimality by enlarging the computation time of BB up to 3,600s. Regarding the 45-min time horizon instances, an optimal solution is always quickly computed for models 2 and 3, even if for four instances proving optimality requires more than 120s. The best solution for model 1 is found on average in 17s of computation but optimality can be proved for only three instances out of 20 within 3,600s of computation. For the 60-min time horizon, we cannot prove optimality for any instance, even if a feasible schedule is always found within a few seconds and no solution is improved after 89s.

We next focus on the 33 instances that remained open at 120s for the time horizon of 30 and 45 min. Table 8 reports the performance of BB on these instances. Each instance is coded in Column 1 with the following *A–B–C–D–E* fields: Under *A*, we denote the number of delayed aircraft at the entrance

TABLE 8. Open instances.

Perturbation identifier	Lower bound	Solut					Time	Time
		Init	30s	60s	120s	Final	Best	Tot
10-U-231.5-30-1	323	—	491	430	398	398*	67.7	129.6
10-U-247.1-30-1	351	—	537	429	403	403*	91.1	395.2
10-U-263.1-30-1	414	—	528	528	480	480*	95.3	328.9
10-U-267.5-30-1	351	—	537	528	525	427*	233.9	373.4
10-U-229.6-30-1	334	—	590	528	528	383*	344.1	404.1
10-G-207.5-30-1	199	—	289	289	287	287*	89.2	201.7
10-G-224.3-30-1	211	—	464	376	307	307*	83.4	202.0
14-U-358.4-45-1	238	—	836	836	836	791	2,257.0	3,600.0
14-U-369.6-45-1	255	—	610	610	610	382	1,603.8	3,600.0
14-U-384.3-45-1	278	—	655	655	655	340*	1,079.7	2,406.0
14-U-384.5-45-1	262	—	725	725	725	601	3,002.0	3,600.0
14-U-370.4-45-1	251	—	847	847	847	740	3,384.0	3,600.0
14-U-352.0-45-1	255	—	—	742	742	480	3,490.0	3,600.0
14-U-347.7-45-1	342	—	709	709	709	709	0.9	3,600.0
14-U-382.6-45-1	275	—	705	705	705	660	2,838.0	3,600.0
14-U-349.5-45-1	207	—	438	284	284	284*	32.9	146.3
14-U-347.6-45-1	266	—	800	800	800	670	1,639.6	3,600.0
14-G-333.0-45-1	134	—	534	534	534	253*	180.1	1,594.5
14-G-326.4-45-1	105	—	537	537	537	537	0.8	3,600.0
14-G-330.5-45-1	117	—	392	392	392	275*	1,394.7	2,424.0
14-G-325.9-45-1	147	—	322	322	322	240*	581.4	637.2
14-G-339.4-45-1	225	—	597	597	597	257*	436.4	536.2
14-G-328.2-45-1	185	—	563	563	563	262*	641.1	3,182.0
14-G-311.9-45-1	190	—	451	451	451	243*	824.7	845.1
14-G-340.4-45-1	186	—	—	537	537	261*	3,348.0	3,375.0
14-U-384.3-45-2	278	341	325	325	325	325*	0.4	255.5
14-U-384.5-45-2	262	356	328	328	328	328*	0.8	444.6
14-U-347.7-45-2	342	459	389	389	389	389*	1.4	3,402.0
14-U-382.6-45-2	275	380	345	345	345	345*	0.6	647.5
14-U-384.3-45-3	278	341	325	325	325	325*	0.4	257.9
14-U-384.5-45-3	262	356	328	328	328	328*	0.6	459.9
14-U-347.7-45-3	342	459	389	389	389	389*	1.5	3,441.0
14-U-382.6-45-3	275	380	345	345	345	345*	0.6	606.8

of the TMA; *B* indicates the distribution type: Uniform (U) or Gaussian (G); *C* represents the average entrance delay in seconds; *D* indicates the time horizon of traffic prediction, from 15 to 60 min; *E* represents the model used for the ASP, where 1–3 are models 1–3. Column 2 shows the final lower bound computed by BB, while column 3 the best solution found by AMCC, AMSP, JGH, and FIFO (if any). Column 4–7 present the solution found by BB after 30, 60, 120, and 3,600 of computation. Each asterisk in column 7 indicates the proven optimality of the corresponding solution. Columns 8–9 report the computation time (in seconds) to reach the best solution and to terminate the algorithmic search.

From the results of Table 8, it turns out that for all tested instances BB improves the initial solution found by the heuristics. For the seven open instances with time horizon equal to 30 min, BB is able to close all instances within around 400s of computation, in five cases, the optimal solution is the same solution found at 120s. Regarding the 26 instances with time horizon of 45 min, BB is able to close 17 instances within the largest time limit and to improve the solutions obtained at 120s for seven of the other nine instances. For nine out of 17 instances solved to optimality, the optimal solution is the same found at 60s.

The analysis of Tables 7 and 8 confirms that models 2 and 3 are easier to solve than model 1. In fact, all the instances associated to the former models are solved to optimality. Regarding the 45-min time horizon instances, an optimal solution is always quickly computed for models 2 and 3, while a feasible schedule is always computed within 1 min for model 1.

As a concluding remark on our computational results, we observe that BB is quite effective for time horizons of practical interest. For time horizons larger than 45 min for models 2 and 3 and larger than 30 min for model 1, BB is still effective in finding feasible solutions but their quality cannot be proved.

6. CONCLUSIONS AND FUTURE WORK

This article presents heuristic and exact algorithms for computing feasible schedules for three AG formulations of the ASP. The proposed models generalize previous work

on job shop scheduling application to solve the ASP. Several solution algorithms are proposed, ranging from simple heuristics to exact algorithms. Test cases for the FCO airport demonstrate that the BB algorithm is able to solve at optimality instances of practical size within a time limit compatible with real-time application. Also when taking into account other performance indicators, BB often outperforms the heuristics.

Test experiments show that the support provided by our optimization models and algorithms has a significant potential for improving the performance of the TMA as well as to reduce the workload of air traffic controllers. Our approach could be used both in real-time and off-line, to assess the quality of alternative plans. However, since off-line analysis frequently focus on larger time horizons than real-time control, further research efforts are worth to develop good lower bounds or advanced metaheuristic algorithms.

Ongoing research is dedicated to the development of online DSSs for ATC at TMAs. The scheduling models and algorithms presented in this work should be part of the system core. Other research directions may be considered: the study of closed-loop control measures for handling disrupted traffic situations (see, e.g., [10]), the design of an integrated approach to deal with a dynamic setting of the DSS, the inclusion of rerouting decisions in the TMA to dynamically balance the load of each runway, the extension of our methodology to better deal with speed variations in the landing procedure, the development of combined aircraft scheduling and routing algorithms to solve the ASP.

Acknowledgments

Preliminary results on a subset of the ASP instances have been presented at the 13th International IEEE Conference on Intelligent Transportation Systems in September 2010 [11].

REFERENCES

[1] L. Adacher, D. Pacciarelli, D. Paluzzi, M. Pranzo, Scheduling arrivals and departures in a busy airport. Preprints of the 5th Triennial Symposium on Transportation Analysis, Le Gosier, Guadaloupe, French West Indies, 2004, pp. 1–4.

[2] A. Allahverdi, C.T. Ng, T.C.E. Cheng, M.Y. Kovalyov, A survey of scheduling problems with setup times or costs, Eur J Oper Res 187 (2008), 985–1032.

[3] K. Artiouchine, Ph. Baptiste, C. Durr, Runway sequencing with holding patterns. Eur J Oper Res 189 (2008), 1254–1266.

[4] K. Artiouchine, Ph. Baptiste, J. Mattioli, The K king problem, an abstract model for computing aircraft landing trajectories: On modeling a dynamic hybrid system with constraints. INFORMS J Comput 20 (2008), 222–233.

[5] M. Ball, C. Barnhart, G. Nemhauser, A. Odoni, "Air transportation: Irregular operations and control," in Handbooks in operations research and management science, G. Laporte and C. Barnhart (Editors), Elsevier, Amsterdam, The Netherlands, 14, 2007, pp. 1–67.

[6] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, D. Abramson, Scheduling aircraft landings—The static case, Transp Sci 34 (2000), 180–197.

[7] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, D. Abramson, Displacement problem and dynamically scheduling aircraft landings, J Oper Res Soc 55 (2004), 54–64.

[8] J.A. Bennell, M. Mesgarpour, C.N. Potts, Airport runway scheduling, Q J Oper Res 9 (2011), 115–138.

[9] L. Bianco, P. Dell'Olmo, S. Giordani, Scheduling models for air traffic control in terminal areas, J Scheduling 9 (2006), 180–197.

[10] J. Clausen, Disruption management in passenger transportation—From air to tracks, Proc 7th Workshop Algorithmic Approaches Transport Model Optim Syst (ATMOS 2007), in C. Liebchen, R.K. Ahuja (Eds), Internationales Begegnungs-und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Sevilla, Spain, pp. 30–47, ISBN 978-3-939897-04-0.

[11] A. D'Ariano, P. D'Urgolo, D. Pacciarelli, M. Pranzo, Optimal sequencing of aircrafts take-off and landing at a busy airport, Proc 13th Int IEEE Annu Conf Intell Transp Syst, Madeira Island, Portugal, September 19–22, IEEE, USA, 2010, pp. 1569–1574. doi:10.1109/ITSC.2010.5625114.

[12] A. D'Ariano, D. Pacciarelli, M. Pistelli, M. Pranzo, Real-time scheduling of aircraft arrivals and departures in a terminal maneuvering area, Tech. Rep. RT-DIA-198–2012 Dipartimento di Informatica e Automazione, Università degli Studi Roma Tre, 2012, pp. 1–32.

[13] A. D'Ariano, M. Pistelli, D. Pacciarelli, Aircraft retiming and rerouting in vicinity of airports, IET Intell Transp Syst J 6 (2012) 433–443.

[14] A. D'Ariano, D. Pacciarelli, M. Pranzo, A branch and bound algorithm for scheduling trains in a railway network, Eur J Oper Res 183 (2007), 643–657.

[15] R.G. Dear, The dynamic scheduling of aircraft in the near terminal area, Report R76-9, Flight Transportation Laboratory, MIT, Cambridge, MA, USA, 1976.

[16] A.T. Ernst, M. Krishnamoorthy, R.H. Storer, Heuristic and exact algorithms for scheduling aircraft landings, Networks 34 (1999), 229–241.

[17] Y. Eun, I. Hwang, H. Bang, Optimal arrival flight sequencing and scheduling using discrete airborne delays, IEEE Trans Intell Transp Syst 11 (2010), 359–373.

[18] N.G. Hall, C. Sriskandarajah, A survey of machine scheduling problems with blocking and no-wait in process, Oper Res 44 (1996), 510–525.

[19] V.J. Hansen, Genetic search methods in air traffic control, Comput Oper Res 31 (2004), 445–459.

[20] J.R. Jackson, Scheduling a production line to minimize maximum tardiness, Report 43, Management Science Research Project, University of California, Los Angeles, USA, 1955.

[21] J. Kim, A. Kröller, J.S.B. Mitchell, G.R. Sabhnani, Scheduling aircraft to reduce controller workload, Proc 9th Workshop Algorithmic Approaches Transp Model Optim Syst (ATMOS 2009), in J. Clausen, G. Di Stefano (Editors), Copenhagen, Denmark, 2009.

[22] J.K. Kuchar, L.C. Yang, A review of conflict detection and resolution modeling methods, IEEE Trans Intell Transp Syst 4 (2000), 179–189.

[23] A. Mascis, D. Pacciarelli, Job shop scheduling with blocking and no-wait constraints, Eur J Oper Res 143 (2002), 498–517.

[24] C. Papadimitriou, P. Kanellakis, Flowshop scheduling with limited temporary storage, J ACM 27 (1980), 533–549.

[25] M. Pistelli, A. D'Ariano, D. Pacciarelli, Optimization models and algorithms for air traffic control in the airspace of busy airports, Tech. Rep. RT-DIA-185–2011 Dipartimento di Informatica e Automazione, Università degli Studi Roma Tre, 2011, 1–26.

[26] M. Pranzo, C. Meloni, D. Pacciarelli, "A new class of greedy heuristics for job shop scheduling problems." in Experimental and efficient algorithms, Lecture notes in computer science, Springer, Germany, vol. 2647, 2003, pp. 223–236.

[27] H.N. Psaraftis, A dynamic programming approach for sequencing identical groups of jobs, Oper Res 28 (1980), 1347–1359.

[28] M.J. Soomer, G.J. Franx, Scheduling aircraft landings using airlines' preferences, Eur J Oper Res 190 (2008), 277–291.

[29] G. Sölveling, S. Solak, J.B. Clarke, E.L. Johnson, Scheduling of runway operations for reduced environmental impact, Transp Res D 16 (2010), 110–120.

[30] C.S. Venkatakrishnan, A. Barnett, A.M. Odoni, Landings at Logan airport: describing and increasing airport capacity, Transp Sci 27 (1993), 211–227.

[31] Z.-H. Zhan, J. Zhang, Y. Li, O. Liu, S.K. Kwok, W.H. Ip, O. Kaynak, An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem, IEEE Trans Intell Transp Syst 11 (2010), 399–412.