Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

Optimization models and algorithms for air traffic control in the airspace of busy airports

Marco Pistelli, Andrea D'Ariano, Dario Pacciarelli

RT-DIA-185-2011 **May 2011**

Università degli Studi di Roma Tre,
Via della Vasca Navale, 79
00146 Roma, Italy.

ABSTRACT

This paper addresses the real-time problem of Aircraft Conflict Detection and Resolution (ACDR) in a Terminal Maneuvering Area (TMA). ACDR is the problem of taking real-time airborne decisions on take-off and landing operations at a congested airport in given time horizons of traffic prediction. The possible aircraft control actions at each air segment and runway in the TMA are speed control, sequencing, holding and routing. We consider the inclusion of rerouting decisions in the TMA to dynamically search for alternative air segments and to balance the load of each runway. The objective function is the minimization of delay propagation and the decision variables are the aircraft timing and routing decisions. This problem can be viewed as a job shop scheduling problem with additional real-world constraints. We study different models for this problem, with increasing level of detail, by using alternative graphs. We investigate the effectiveness of several neighborhood structures for aircraft rerouting, incorporated in a state-of-the-art tabu search scheme based on a generalized critical path method. The effectiveness of solution algorithms are evaluated on practical size instances from the Rome FCO and Milan MXP airports, in Italy. Disturbances regarding the entrance time of aircraft in the TMA are simulated for assessing the optimization models and procedures under congested traffic conditions. The computational results also demonstrate the effectiveness of the tabu search algorithm to reduce delays and travel times when compared with the heuristic and exact aircraft scheduling solutions.

Keywords: Conflict Resolution, Microscopic Modeling, Optimization Algorithms.

1 Introduction

This paper deals with the development of optimization models and algorithms for improving real-time aircraft operations at busy airports. From a logical point of view, Air Traffic Control (ATC) decisions in a TMA can be broadly divided into: (i) Routing decisions, where an origin-destination route for each aircraft has to be chosen regarding air segments and runways; (ii) Timing decisions, where routes are fixed under traffic regulation constraints and aircraft passing timing have to be determined in each air segment, runway and (eventually) holding circle.

In practice, routing (i) and scheduling (ii) decisions in a TMA are taken simultaneously and a given performance index is optimized. The main objective of routing decisions is typically to balance the use of critical resources (e.g., alternative runways, air corridors) while the whole process is to limit aircraft delays [8].

Decision Support Systems (DSSs) based on optimization algorithms may help to exploit at most the capacity available in a TMA during operations. In this context, the optimization of take-off/landing operations is an important 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. As a result, the delay propagation is not effectively limited during landing and take-off operations.

The Aircraft Conflicts Detection and Resolution (ACDR) problem has been the subject of several studies (see the literature reviews in [3, 16, 20]).

The ACDR models can be broadly classified as basic or detailed. In the basic models only the runways are included in the TMA, while detailed approaches also model the air segments. In general, 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 also happen 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.

The basic model has been investigated in [2, 4, 5, 15, 17, 18, 22, 23, 24]. The detailed approach has been first studied in [1, 7]. However, these works are concentrated on the scheduling problem only. Our goal is to extend the ACDR problem to the possibility of rerouting aircraft in the detailed approach.

Recent studies have been dedicated to a complementary problem that is the ATFM (Air Traffic Flow Management) problem in large networks with multiple airports [6, 8, 9, 19]. Their approach presents a broader view on delay propagation compared to TMA models but the adopted models are macroscopic and potential conflicts between aircraft are not visible at the level of air segments and runways but in terms of aggregated flight paths only.

This paper focuses on the real-time control problem to provide optimal conflict-free airborne decisions at the TMA. Similar problems are also studied in railway transportation field for reordering and rerouting trains [11, 12]. However, the two types of problems have a quite different structure and require careful adaptation of existing models and algorithms.

In a recent work on the ACDR problem with fixed routes, that is an NP-complete problem, we developed a branch and bound algorithm in which aircraft routes are decided a preliminary step [13].

In this work the ACDR problem with flexible routing is partitioned into two subproblems. A rerouting problem, in which a route among a set of rerouting possibilities is associated to each aircraft, and a scheduling problem in which a start time is assigned to each operation.

The aircraft scheduling problem is modeled as a generalized job shop scheduling problem and is formulated via alternative graphs [21], that are able to enrich the model of Bianco et al. [7] by including additional real-world constraints, such as holding circles, time windows for aircraft speeds, multiple capacity of air segments and blocking constraints at runways. This formulation allows accurate modeling of future air traffic flows on the basis of the actual aircraft positions and speeds, and safety constraints. We describe and test retiming and rerouting algorithms for the ACDR problem on practical size instances of the Roma Fiumicino (FCO) and the Milano Malpensa (MXP) airports. The optimization procedures are evaluated in terms of delay minimization and travel time spent.

We applied our algorithms to instances derived from real data relating to Roma Fiumicino and Milano Malpensa, that are the two busiest Italian airports in terms of passenger flows. The structures of the two airports are introduced in the following paragraphs together with examples of detailed formulations. Networks models of the two airports differ not only in the structure of aircraft routes, but also in the types of rerouting allowed (air segments, runways, holdings).

For each configuration of aircraft delay, we implemented two job shop scheduling models for the resolution of ACDR instances: a more realistic one with stringent constraints at the entry fix, and another one that relaxes some hard timing constraints. The aim is to show how different policies of cooperation between airspace's controllers allow them to obtain alternative solutions for specific traffic disturbances.

The paper is organized as follows. Section 2 describes the ACDR problem and its formulation via alternative graphs. Section 3 presents the models for aircraft scheduling and routing. Section 4 discusses models with different level of detail in the formulation of problem constraints. Section 5 describes the system architecture and optimization algorithms. Section 6 reports on the instances tested for Fiumicino and Malpensa airports. Section 7 compares alternative configurations of the rerouting algorithms. Section 8 gives the computational results for the two airports. Section 9 concludes the paper and suggests further research directions.

2 Problem description and formulation

This section introduces the ACDR problem, describes alternative models and gives illustrative examples.

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* between every pair of consecutive aircraft must be guaranteed. This standard separation depends on the type and relative positions of the two aircraft (at the same or different altitude). By considering the different aircraft speeds, the safety distance can be translated in a separation time. 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 separation time should be ensured between any pair of aircraft. Once a landing/departing aircraft enters the TMA it should proceed to the runway. However, airborne (ground) holding circles can be used to make aircraft wait in flight (at ground level) until they can be guided into the landing (take-off) sequence.

Real-time traffic management copes with potential aircraft conflicts by adjusting the off-line plan (timetable) in terms of retiming, reordering, rerouting and holding actions. A *conflict* occurs whenever aircraft traversing the same resource (i.e, air segment or runway) do not respect the minimum separation time required for safety reasons. Separation times depend not only on the aircraft sequence but also on the route chosen for consecutive aircraft.

The main goal of this rescheduling process is to reduce aircraft delays (i.e., the difference between the arrival time at the runway/entry fix in the new schedule and that in the timetable) while satisfying traffic regulation constraints and the compatibility with the real-time position of each aircraft. The latter information enables the computation of its *release time*, which is the minimum time at which the aircraft can enter the network.

We compute aircraft delays as follows. A departing aircraft is supposed to take-off within its assigned time window and is late whenever it is not able to accomplish the departing procedure within its assigned time window. Since we prefer to absorb aircraft delays on the ground, departing aircraft are considered late only if leaving the airport after a given delay. We fix this delay to 10 minutes from their scheduled departure time that is defined in a time slot of 15 minutes. Arriving aircraft are late if landing after their scheduled arrival time. The aircraft delay is partly due to the entrance delay in the TMA and partly due to the additional delay caused by the resolution of potential aircraft conflicts. We minimize this second part, that is called *consecutive delay* [11, 12, 13].

The ACDR problem consists of choosing a route for each aircraft and conflict-free timings for all chosen routes such that separation times between aircraft are satisfied, no aircraft enters the network before its release time and consecutive delays are minimized.

We now introduce further notation before giving the alternative graph formulation of the ACDR problem. The traversing of an air segment or runway by an aircraft is known as *operation*. Each aircraft has associated a route that is denoted as the sequence of operations related to this aircraft, i.e., the sequence of operations to be executed on air segments, holdings or runways. The variables of the ACDR problem are the set of operations to be performed by each aircraft (routing decisions) and the start time t_i of each operation i (timing decisions). A timing specifies the start time of each operation in the route.

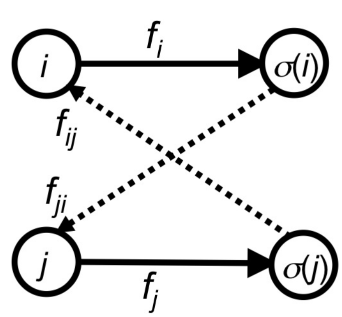

Figure 1: Two aircraft approaching a runway

Given an operation i, we denote with $\sigma(i)$ the operation which follows i on its route and with f_i a minimum *processing time* of operation i (see Figure 1). A timing is feasible if $t_{\sigma(i)} \geq t_i + f_i$, for every operation in the route. A set of feasible route timings is *conflict-free* if, for each pair of operations associated to the same resource, the minimum separation constraints are satisfied. For example, if i and j are two operations associated with the entrance of two aircraft in a runway (blocking resource) and f_{ij} is the separation time when i precedes j, the separation constraint requires that the aircraft associated to i must leave the resource at least f_{ij} time units before the aircraft associated to j can enter the runway, i.e., $t_j \geq t_{\sigma(i)} + f_{ij}$. Similarly, if j precedes i, $t_i \geq t_{\sigma(j)} + f_{ji}$ holds.

Associating a node with each operation and a job to the selected route for each aircraft in the TMA, the ACDR problem can be represented by an alternative graph that is a triple $\mathcal{G} = (N, F, A)$, where $N = \{0, 1, \ldots, n, *\}$ is the set of nodes, F is a set of directed arcs and A is a set of pairs of directed arcs.

Arcs in the set F, *conjunctive*, model aircraft routing decisions. Each fixed arc of the set F is associated to an operation i and has weight f_i. Arcs in the set A, *alternative*, model aircraft sequencing and holding decisions. If $((i,j),(h,k)) \in A$, arc (i,j) is the alternative of arc (h,k). Each alternative arc (i,j) has an associated weight f_{ij}.

The nodes $1, \ldots, n$ are associated to the operations of the aircraft routes defined by the set F. Additional nodes 0 and $*$ are used to model the start and completion of the schedule. A *selection* $S(F)$ is a set of alternative arcs, at most one from each pair. A selection, in which exactly one arc is chosen from each pair in A, is a *feasible schedule* (i.e., a solution to the ACDR problem) if the connected graph $(N, F \cup S(F))$ has no positive length cycles. Given a selection $S(F)$, a timing t_i for operation i is the length of a longest path from 0 to i (i.e., $l^{S(F)}(0,i)$). A selection $S(F)$ is *optimal* if $l^{S(F)}(0,*)$ is minimum over all the solutions.

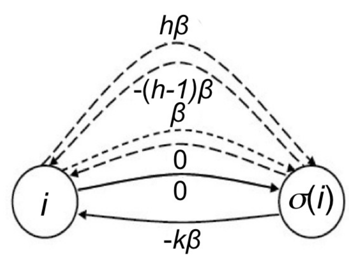

Figure 2: Alternative graph formulation of holding circles

Figure 2 shows the formulation of holding circles on the alternative graph. Let i be the entrance of the aircraft in the holding and $\sigma(i)$ the following operation. On the graph there is a pair of fixed arcs $(i, \sigma(i))$ and $(\sigma(i), i)$, and k alternative pairs. The length of $(i, \sigma(i))$ is 0. The length of $(\sigma(i), i)$, instead, is $-k\beta$, where β is the time required to do a complete half circle and k is the maximum number of half circles allowed. The k alternative pairs $((i, \sigma(i)), (\sigma(i), i))$ weight $h\beta$ and $-(h-1)\beta$ respectively, $h = 1, \ldots, k$. In a feasible solution we obtain $\bar{h}$ such that if $(i, \sigma(i))$ with weight $\bar{h}\beta$ is selected, also the arc $(\sigma(i), i)$ is selected too. In this case all the arcs $(\sigma(i), i)$ and $(i, \sigma(i))$ with $h > \bar{h}$ are selected: these arcs are redundant. The arcs corresponding to $\bar{h}$, instead, impose a $\bar{h}$ number of half circles for the aircraft.

3 Scheduling and routing models

This section introduces the scheduling and routing models for the ACDR problem. To describe the models, we use the Fiumicino airspace.

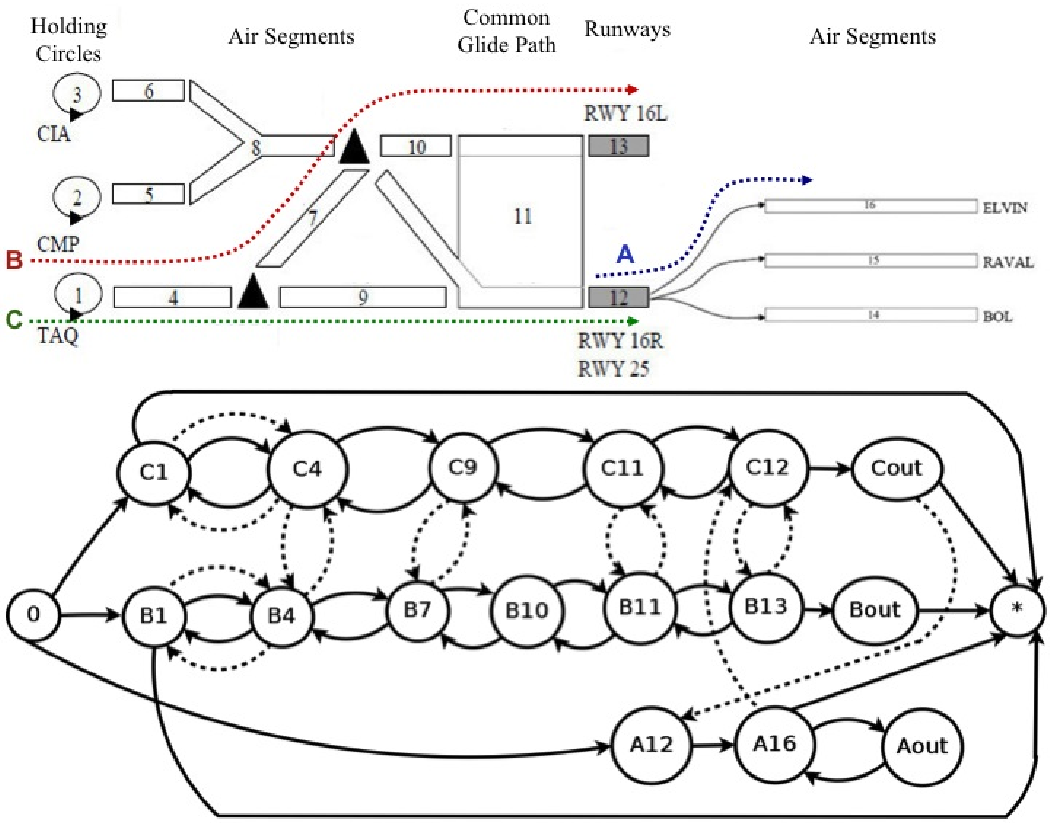

Figure 3: Example of a problem formulation

Figure 3 shows TMA of the FCO airport in which there are three runways (16L, 16R, 25), two of which (16R, 25) are intersecting and cannot be used simultaneously. The airport resources are 3 airborne holding circles (TAQ, CMP, CIA; numbered 1–3), 7 air segments for landing procedures (4–10), 3 air segments for take-off procedures (14–16) leading to the exit points of the TMA (ELVIN, RAVAL, BOL), two runways (12–13) and a common glide path (11). The latter resource includes two parallel air segments before the two runways for which, besides a minimum longitudinal distance between aircraft, traffic regulations also impose a minimum diagonal distance.

The alternative graph of Figure 3 shows a departing aircraft (A) and two landing aircraft (B and C).

Each aircraft flies through a given route (air segments, holding circles and runways) and overtaking is not allowed within an air segment in the TMA. Each node of the graph represents an operation, e.g., $C11$ is aircraft C entering air segment 11. Fixed (alternative) arcs are depicted with solid (dotted) arrows. On air segments, minimum and maximum traversing times are given for each aircraft, depending on its specific speed characteristics. This can be represented in $\mathcal{G}$ with a pair of fixed arcs (e.g., $(C4, C9)$ and $(C9, C4)$). The minimum separation time between consecutive aircraft in the same air segment is modeled as a sequence dependent setup time (see, e.g., the alternative pairs $((C4, B4), (C9, B7))$

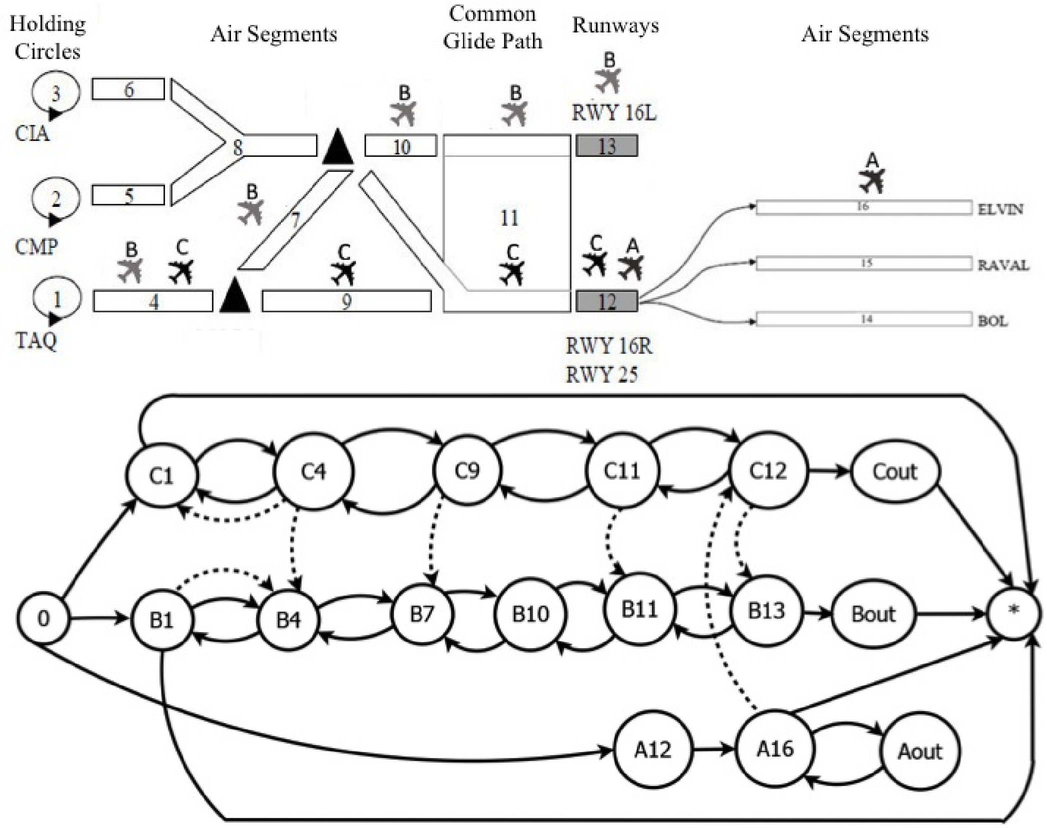

Figure 4: Aircraft scheduling solution

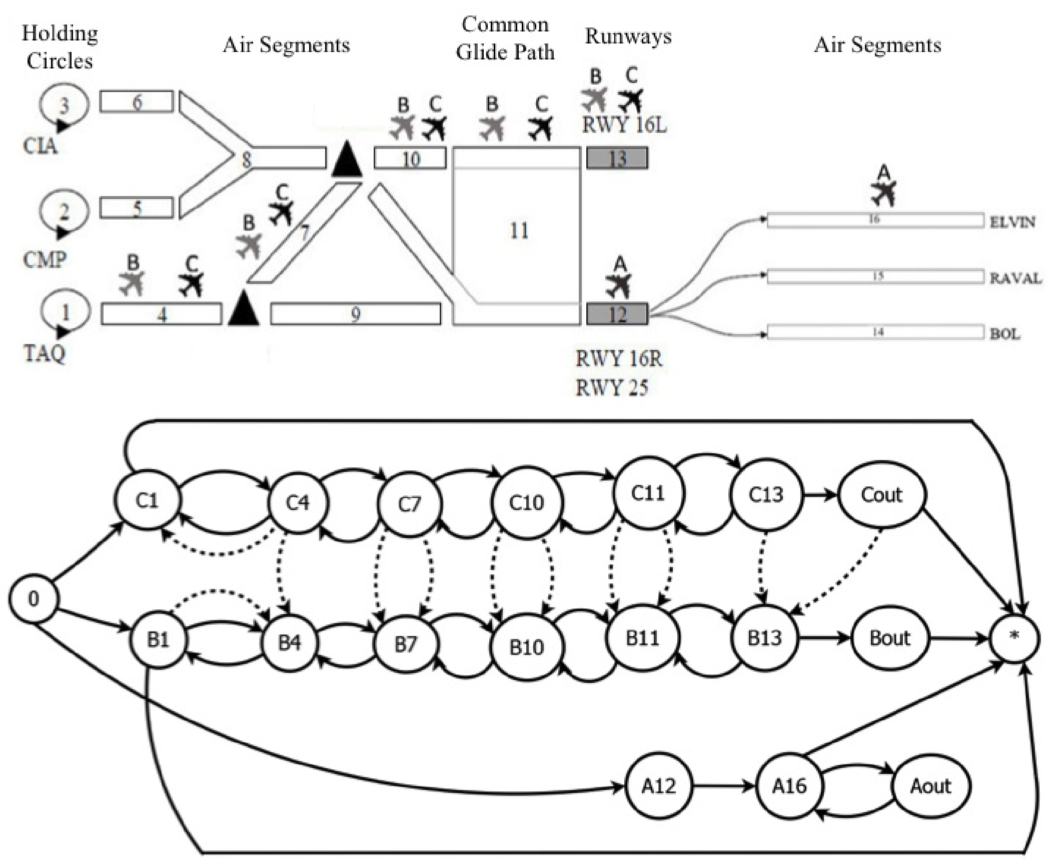

Figure 5: Scheduling solution with a route change

and $((B4, C4), (B7, C9))$.

The presence of aircraft on a runway imposes that no other aircraft can use it and thus the runway is a blocking resource (see, the alternative pair $((Cout, A12), (A16, C12))$. Once an aircraft enters an airborne holding circle, it must fly at a given speed and can leave the holding circle only after the traversing of at least half length of the circle. The two alternative decisions are formulated with two fixed arcs and a pair of alternative arcs. For example, the two fixed arcs for aircraft B are the solid arrows $(B1, B4)$ and $(B4, B1)$ with weights 0 and $-\beta$. The pair of alternative arcs for aircraft B is $((B1, B4), (B4, B1))$, weighted β and 0 and depicted with dotted arrows. In a solution, one of the two alternative arcs must be chosen. Choosing $(B1, B4)$ introduces a zero length cycle and forces the aircraft B to enter in the holding circle. Choosing $(B4, B1)$ introduces another zero length cycle but forces the aircraft B to skip the holding circle. To limit the number of round trips for each aircraft in a holding circle, we insert a due date arc for each aircraft at its entry fix. In the next section, we will show due date constraints can be replaced by deadline constraints in order to constraint the entrance time of aircraft.

We assume infinite capacity at the holding circles, and there are thus no conflicts between B and C (i.e., operations $C1$ and $B1$). For the aircraft routes of Figure 3, there are potential conflicts between B and C at two air segments (4 and 11) and between A and C at a runway (12).

We measure the objective function when an aircraft leaves a runway (e.g. $(A16, *)$).

Figure 4 shows a solution to the problem of Figure 3. This solution is obtained by scheduling aircraft C before than aircraft B on the air segment 4 (selecting the alternative arcs $(C4, B4)$ and $(C9, B7)$), and on the air segment 11 (selecting the alternative arcs $(C11, B11)$ and $(C12, B13)$), while aircraft A is scheduled first on the runway 12 (selecting the alternative arc $(A16, C12)$). We assume that aircraft C does not run any airborne holding circle (selecting the alternative arc $(C4, C1)$). Differently, aircraft B runs a half circle of weight β (selecting the alternative arc $(B1, B4)$) in order to avoid the potential conflict with aircraft C on the air segment 4. The half circle would cause a consecutive delay for the start time of node $C1$ due to the due date arc $(C1, *)$.

Figure 5 shows an alternative graph $\mathcal{G}' = (N', F', A')$ for which the route of C is changed. Specifically, aircraft C is now landing on runway 13. In the new graph, there are still potential conflicts between B and C at four air segments (4, 7, 10 and 11) and a runway (13), while there is no sequencing decision between A and C on the other runway (12).

4 Alternative detailed models

This section presents two types of ACDR models, with and without deadline constraints. To explain and compare the models we use alternative graph formulations of Malpensa airspace. Figure 6 shows an illustrative example with two landing aircraft and one departing aircraft in the TMA of Milano Malpensa. As for the Fiumicino case, landing aircraft enter the TMA at an entry fix (with or without) holding circle. In Malpensa case, aircraft enter the network at TOR, MBR or SRN (number 1-3).

The first model we developed is "Model 1", shown in Figure 7. In this model, the landing aircraft have a deadline at the beginning of their processing. In the alternative graph, a deadline constraint is represented by a fixed arc from the first operation of the job

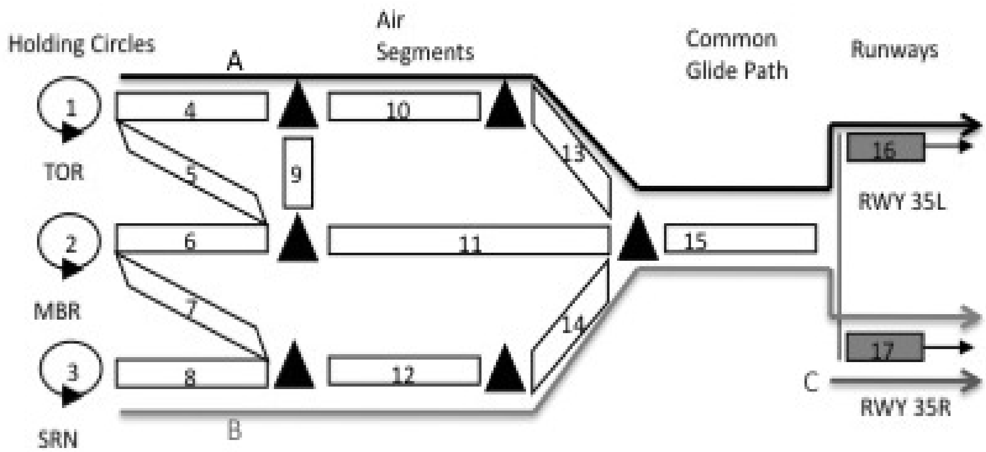

Figure 6: Example of ACDR with three aircraft, Malpensa

to node 0 (arcs $(A1, 0)$ and $(B3, 0)$). Deadline arcs have the same weight, with opposite sign, of the release time for each landing aircraft. It means that each landing aircraft must start its processing exactly when it appears at an entry fix of the TMA, and it can not be delayed at the entrance. This feature is close-fitting with the real operative procedures managed by the controllers of the TMA. In Model 1, departing aircraft have a due date constraint, instead of a deadline, at the beginning of their processing. This due date is represented by a fixed arc from the first operation of the job to node $*$ (arc $(0, C17)$).

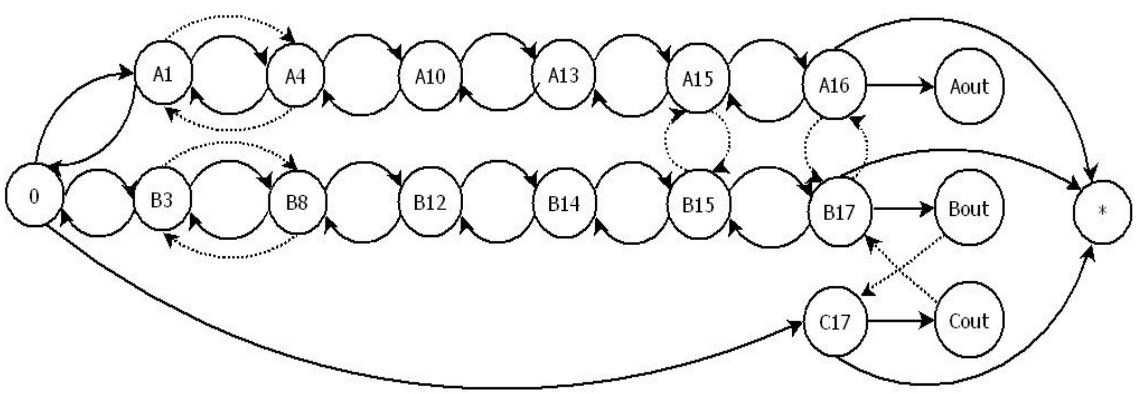

Figure 7: Alternative graph formulation of Model 1

The second model we developed is "Model 2", shown in Figure 8. In this model, landing aircraft have a due date at the beginning of their processing instead of a deadline. In the graph a due date at the entry fix is represented by a fixed arc from the first operation of the job to node $*$ (arcs $(A1, *)$ and $(B3, *)$). This type of due date has the same weight, with opposite sign, of the release time for each landing aircraft. The main difference between the two models is that Model 2 landing aircraft can begin their processing also after their entrance in the TMA, causing a delay penalty in the objective function. There are no differences between the models with regard to departing aircraft.

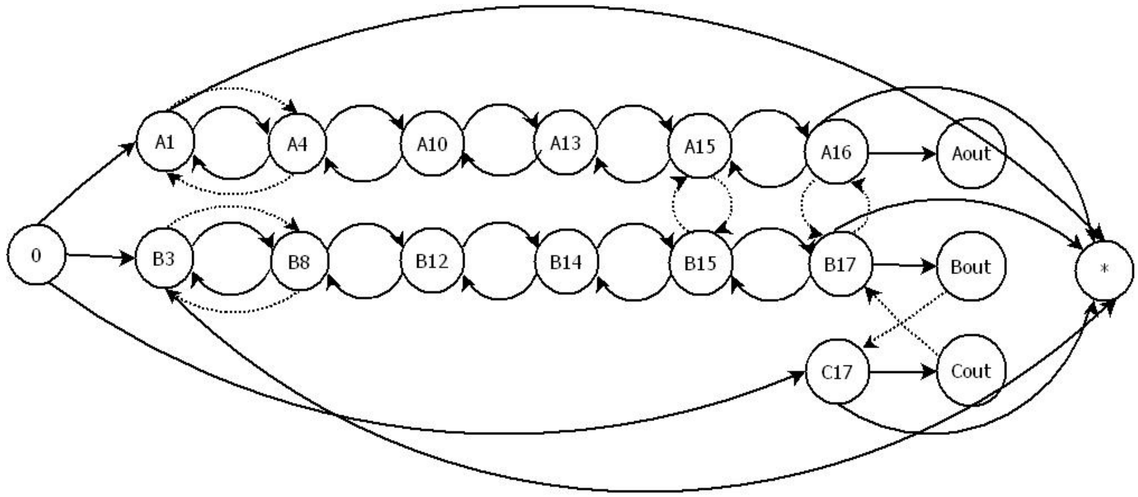

Figure 8: Alternative graph formulation of Model 2

The solutions provided by Model 2 consider a larger degree of flexibility regarding the entrance time of each aircraft in the TMA with respect to Model 1. Specifically, it is required that air traffic controllers control the entrance time of each incoming aircraft. This is clearly possible only if the solutions are found sufficiently in advance respect to the actual entrance time of aircraft and if there is a coordination action between the controllers of the TMA and the controllers of neighboring areas. These issues are relevant for a practical application of the proposed decision support system but we will not address them in the current study, that is more focused on a quantitative assessment of advanced optimization models and algorithms.

5 Scheduling and routing optimization algorithms

Figure 9 describes the architecture of our system for decision support, similar to the one proposed in D'Ariano et al. [12] for railway traffic control. The decision support system is in charge of computing a feasible aircraft schedule and then looking for better aircraft routing solutions.

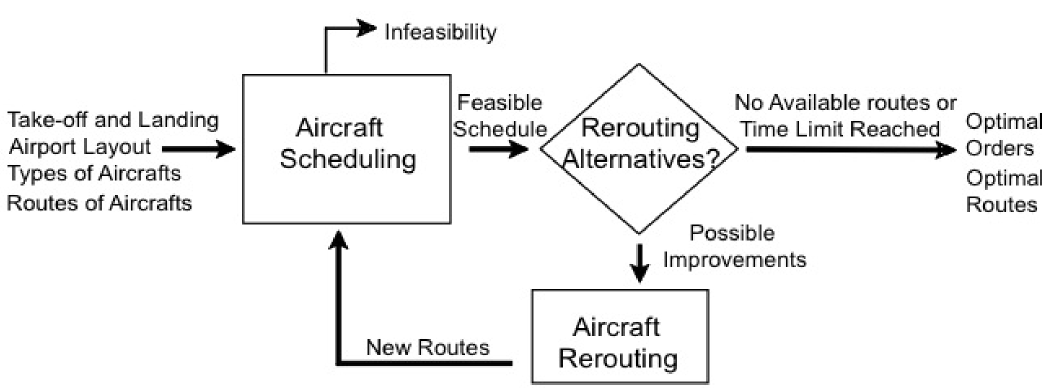

Figure 9: Schematic view of the decision support system

Given a timetable, the current status of the network, an off-line defined route and a set of rerouting options for each aircraft, the aircraft rescheduling module returns a feasible schedule for the given aircraft routes. At the first run the scheduling module considers the off-line routes. If no feasible schedule is found within a predefined time limit of computation, the human dispatcher is in charge of recovering infeasibilities by taking some stronger decisions (forbidden to the automated system), such as rerouting some aircraft to a different airport. When a feasible schedule is found, the aircraft rerouting module verifies whether a rerouting option, leading to a potentially better solution, exists. In our computational experiments, the rerouting module changes one aircraft route at each iteration. In general, our procedure may work as well with multiple rerouting options. For each changed route, running times and setup times are modified accordingly. Whenever rerouting is performed, the aircraft rescheduling module computes a new conflict-free timetable by thoroughly rescheduling aircraft movements. The iterative rescheduling and rerouting procedure returns the best solution found when a given computation time is reached or no rerouting improvement is possible.

The aircraft scheduling subproblem is solved by using the branch and bound (BB) procedure of [13]. The search scheme used in this paper branches with priority on sequencing aircraft on the runways.

The aircraft rerouting subproblem is solved by a tabu search (TS) based on the approach of [11]. Basic ingredients of a tabu search are the concepts of move and tabu list, which restrict the set of solutions to explore. From the incumbent solution, non-tabu moves define a set of solutions, called the neighborhood of the incumbent solution. At each step, the best solution in this set is chosen as the new incumbent solution. Then, some attributes of the former incumbent are stored in a tabu list of length λ, used by the algorithm to avoid being trapped in local optima and to avoid re-visiting the same solution. The moves in the tabu list are forbidden as long as these are in the list.

In this paper, we restrict the search to promising routes by using a strategy based on the neighborhood containing all the feasible solutions to the ACDR problem plus a set of five neighborhood strategies based on the critical path method. We briefly recall some concepts regarding the neighborhood strategies (more detail is given in [11]).

In the strategies we tested, the neighborhood of the current solution $(F, S(F))$ is made by all the fixed arcs differing from F exactly for one aircraft route and a move is the change of a route in F in order to obtain a new set of fixed arcs F'. To evaluate the move we have to value the solution $(F', S(F'))$. The implementation of a move corresponds to minimize the objective function on the graph with the new route. TS minimizes maximum and average consecutive delays in the TMA, in lexicographic order.

A move can be evaluated by these alternative methods:

- A lower bound (LB) for the aircraft scheduling problem that is based on the computation of a single machine preemptive schedule with implications for each air segment, as described in [13];

- An upper bound (UB1) for the aircraft scheduling problem that is based on recomputing the orders related to the rerouted aircraft while leaving unchanged the part of the schedule not related to this aircraft, as described in [11];

- Another upper bound (UB2) for the aircraft scheduling problem that uses the BB algorithm of [13], stopped after 10 seconds of computation.

Besides the use of the three methods for move evaluation, we always use UB2 for move implementation.

Given a solution $(F, S(F))$ to the ACDR problem and a node in neighborhood of the incumbent solution $i \in N(F) \setminus \{0, *\}$, let $\sigma^{-1}(i)$ be the node which precedes i on the new aircraft route. We say that i is a critical node in $S(F)$ if $l^{S(F)}(0,i) + l^{S(F)}(i,n) = l^{S(F)}(0,*)$. A critical node is a waiting node if $l^{S(F)}(0,i) > l^{S(F)}(0,\sigma^{-1}(i))$, such that $l^{S(F)}(0,i) = l^{S(F)}(0,h_i) + f_{hi}$. We call h_i the hindering node of i. By definition for each waiting node $i \in N(F) \setminus \{0, *\}$ there is exactly one hindering node (possibly node 0).

Given a node $i \in N(F) \setminus \{0, *\}$, we recursively define its backward ramification $R_B(i)$ as follows. If i is a waiting node then $R_B(i) = R_B(\sigma^{-1}(i)) \cup R_B(h_i) \cup \{i\}$, otherwise $R_B(i) = R_B(\sigma^{-1}(i)) \cup \{i\}$. We recursively define the forward ramification $R_F(i)$ as follows. If i is the hindering node of a waiting node k (i.e., $i = h_k$), then $R_F(i) = R_F(\sigma(i)) \cup R_F(k) \cup \{i\}$, otherwise $R_F(i) = R_F(\sigma(i)) \cup \{i\}$. Moreover, by definition $R_B(0) = R_F(0) = \{0\}$ and $R_B(n) = R_F(n) = \{n\}$.

Given $(F, S(F))$, we call critical path set $\mathscr{C}(F, S(F))$, the set of critical nodes, backward ramified critical path (BRCP) set $\mathscr{B}(F, S(F)) = \bigcup_{i \in \mathscr{C}(F,S(F))} R_B(i)$ and forward backward ramified critical path set (FBRCP) the set $\mathscr{R}(F, S(F)) = \bigcup_{i \in \mathscr{C}(F,S(F))} (R_B(i) \cup R_F(i))$.

We consider three different neighborhood structures:

- $\mathscr{N}_C$, the complete neighborhood, contains all the feasible solutions to the ACDR problem in which exactly one aircraft follows a different route compared to the incumbent solution;
- $\mathscr{N}_{BRCP}$, the backward ramified critical path neighborhood, contains only the aircraft with at least one operation in $\mathscr{B}(F, S(F))$;
- $\mathscr{N}_{FBRCP}$, the forward backward ramified critical path neighborhood, extends the previous neighborhood considering also the forward ramifications.

The tested search strategies are:

- **Complete**: Candidate solutions in $\mathscr{N}_C$ are evaluated (either with LB, UB1 or UB2) and the best is chosen as the new incumbent.
- **Restart**: Candidate solutions in $\mathscr{N}_{FBRCP}$ are evaluated (either with LB, UB1 or UB2) and the best is chosen as the new incumbent. When $\mathscr{N}_{FBRCP}$ is empty, $\gamma \geq 1$ consecutive moves are performed in $\mathscr{N}_C$ before searching again in the restricted neighborhood, where γ is a parameter of the tabu search algorithm. In each move, ψ randomly chosen routes are evaluated (either with LB, UB1 or UB2) and the best is chosen without implementing the move (i.e., without solving the ACDR problem with UB2). A new schedule is then computed with the UB2 algorithm only after the γ moves.
- **Hybrid1**: Candidate solutions in $\mathscr{N}_{FBRCP}$ are evaluated (either with LB, UB1 or UB2) and the best is chosen as the new incumbent. If $\mathscr{N}_{FBRCP}$ is empty, $\gamma > 1$ consecutive moves are performed $\mathscr{N}_C$ before searching again in $\mathscr{N}_{FBRCP}$, where γ is a parameter of the tabu search. In the latter case the move is chose by evaluating ψ candidate solutions in $\mathscr{N}_C$ with UB2 and implementing the best one by taking into

account the two objective functions in lexicographic order (i.e., the minimization of the [maximum; average] consecutive delays). After each move, a new schedule is computed with the UB2 algorithm.

- **Hybrid2**: As in Hybrid 1 but the candidate solutions in $\mathscr{N}_C$ are evaluated by taking into account only the average consecutive delay.

- **Hybrid3**: Candidate solutions in $\mathscr{N}_{FBRCP}$ are evaluated (either with LB, UB1 or UB2) and the best is chosen as the new incumbent if there is an improving move. If a local minimum is reached, ψ candidate solutions in $\mathscr{N}_C$ are evaluated (either with LB, UB1 or UB2) and the best solution found among those in $\mathscr{N}_{FBRCP} \cup \mathscr{N}_C$ is implemented. The two objective functions are considered in lexicographic order. After each move, a new schedule is computed with the UB2 algorithm.

For each neighborhood strategy, we performed a preliminary set of experiments on 27 instances of the ACDR problem. Each instance corresponds to a graph with a number of alternative pairs up to 1292. We analyzed (i) the number ψ of neighbors to be evaluated at each iteration, (ii) the tabu list length λ, (iii) the number γ of moves to be performed in case of restart. The best configuration for each neighborhood exploration strategy is as follows. For the Complete and Hybrid3 strategies the best values of (ψ, λ,γ) are (12; 27; 3). For the Restart, Hybrid1 and Hybrid2 strategies the best values of (ψ, λ,γ) are (10; 32; 5).

6 Description of the test cases

We tested the solution algorithms on real data from the Fiumicino (FCO) and Malpensa (MXP) airports. For each airport, we considered jointly aircraft landing and departing operations. The experiments are executed on a processor Intel i7 (2.84 Ghz), 8 GB Ram and Linux operating system. For each algorithm, we fix a maximum computation time of 120 seconds.

In Fiumicino case, we consider an entry fix for each aircraft in the TMA and two alternative routes for each landing aircraft, one route for each runway. The departing aircraft, instead, have a fixed path to leave the airport. Table 1 presents the main characteristics of 80 instances we use to test the ACDR algorithms. Each row reports average data over 20 instances. Column 1 shows four time horizons of traffic prediction, Column 2 the number of arriving/departing aircraft, Columns 3-5 the average number of nodes ($|N|$), of fixed arcs ($|F|$) and of alternative pairs ($|A|$). The average number of nodes and arcs, fixed and alternatives, is the same in both models, since we have sequencing decisions on the same resources and all the deadline arcs of the first model become due date arcs in the second one. Columns 6-7 give the maximum and average entrance delays (in seconds). For each time horizon, we generate 10 Uniform and 10 Gaussian (randomly generated) entrance delays in the TMA. We only delay aircraft that enter the network in the first half of each time horizon under study. Entrance delays are computed off-line before the resolution of the ACDR problem.

For Malpensa, Table 2 reports the main features for the 80 instances generated for both models, with the same entrance delay configuration as for the Fiumicino case.

Table 1: Disturbed operations and prediction horizon, Fiumicino

Time Horiz	Land/Dep Aircraft	$\|N\|$	$\|F\|$	$\|A\|$	Max Delay	Avg Delay
15	6/1	44	290	103	450	170.4
30	16/4	118	1605	688	900	234.7
45	22/7	166	2906	1292	1350	345.6
60	32/16	259	6425	2963	1800	465.1

Table 2: Disturbed operations and prediction horizon, Malpensa

Time Horiz	Land/Dep Aircraft	$\|N\|$	$\|F\|$	$\|A\|$	Max Delay	Avg Delay
15	6/5	51	336	121	450	118.9
30	13/6	107	1297	546	900	252.5
45	19/13	155	2382	1044	1350	333.1
60	23/16	187	3393	1519	1800	460.6

Table 3 shows the routes considered for each TMA. At the Fiumicino TMA the rerouting consists of changing both the air path and the runway for the landing aircraft. Departing aircraft are not rerouted. At the Malpensa TMA, we consider three groups of alternative routes: the combined one (aircraft can change route in air and runway), the air one (aircraft can change the route in air only, not the runway), and the runway one (aircraft can change the runway they use, not their route in air). Both landing and departing aircraft can be rerouted, but the departing aircraft are rerouted on the runways only.

Table 3: Total number of aircraft routes

Time Horiz	FCO	MXP		
		Combined	Air	Runway
15	13	34	17	22
30	36	70	35	38
45	51	116	58	64
60	80	138	69	78

7 Tuning of the tabu search algorithm

We next discuss the choice of which of the tabu search configurations presented in Section 5 perform better for the instances under study. Specifically, we analyze which of the five configurations mentioned in Section 5 is the most performing for each model. The results for Fiumicino airport are shown in Figures 10 – 13. In these figures, at time $t = 0$ we report the average results on the nearly optimal scheduling solutions with off-lines routes. The scheduling solutions are computed by BB. The average solutions of the aircraft rerouting algorithms are depicted each 10 seconds of computation, up to two minutes. As regard Fiumicino airspace, we choose Restart as the best configuration for both models. In fact, for Model 2 Restart gives the best performance both for the maximum and average consecutive delays. For Model 1, Restart presents results similar to Complete but the former has a better performance in terms of the number of feasible solutions found. In Figures 10 – 13, we only consider instances for which all configurations find a feasible solution.

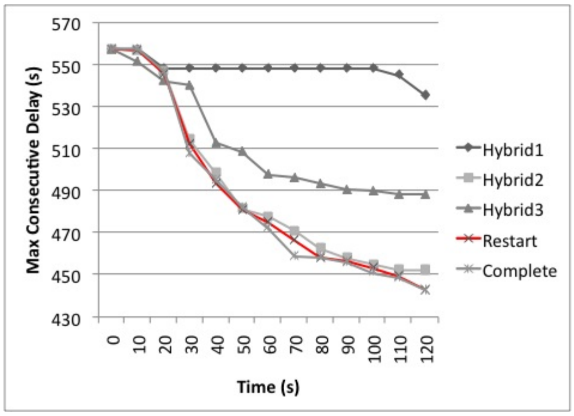

Figure 10: Maximum consecutive delay model 1, Fiumicino

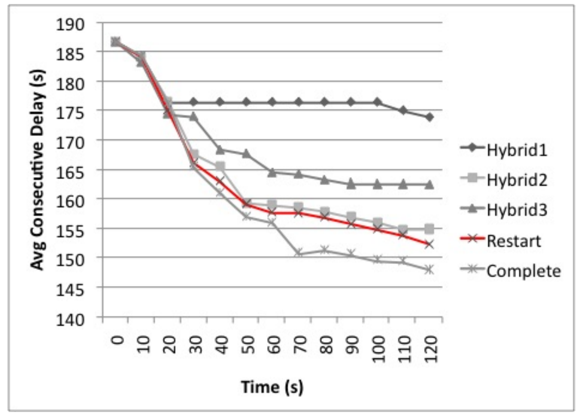

Figure 11: Average delay model 1, Fiumicino

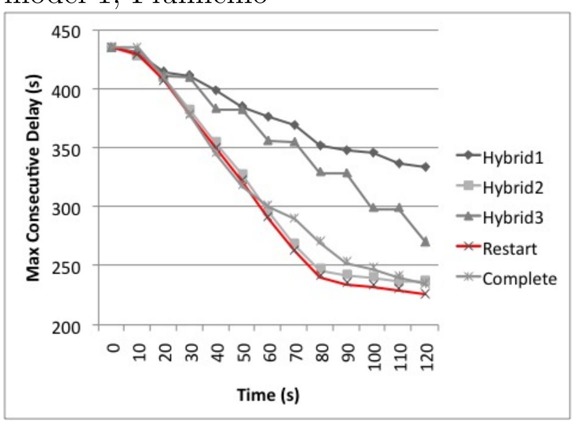

Figure 12: Maximum consecutive delay model 2, Fiumicino

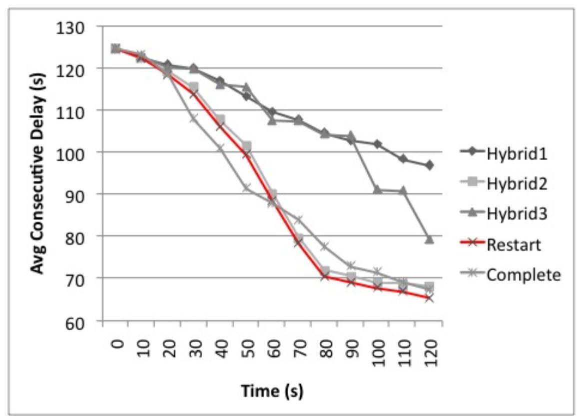

Figure 13: Average delay model 2, Fiumicino

In Malpensa case, we consider an entry fix for each aircraft in the TMA, but landing aircraft have more alternative routes compared to the other airport. In fact, aircraft have two or three different paths to reach the runways, depending on their entry point, and they are free to choose the runway to use. Also the departing aircraft have flexibility regarding the runway. In the Malpensa airspace, there are three groups of alternative routes (combined, air and runway). The result of the five tabu search configurations are shown in Figures 14 – 17. From these figures, we choose the Complete strategy as the best configuration for both models of the Malpensa airspace.

Comparing the TS solutions for the Train Conflict Detection and Resolution (TCDR) problem of [11] with those obtained for the ACDR problem we notice a different best configuration. For the former problem the best configuration is Hybrid 3, while for the latter problem the best configuration is, on average, Complete. The motivation of this difference is probably related to the different type of resources used in the alternative graph formulations. In the TCDR problem, all the resources are blocking. In the ACDR problem, most of the resources have multiple capacity (air segments), while a few of them are blocking (runways).

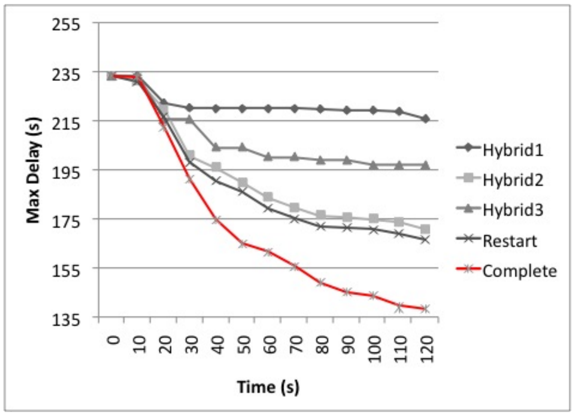

Figure 14: Maximum consecutive delay model 1, Malpensa

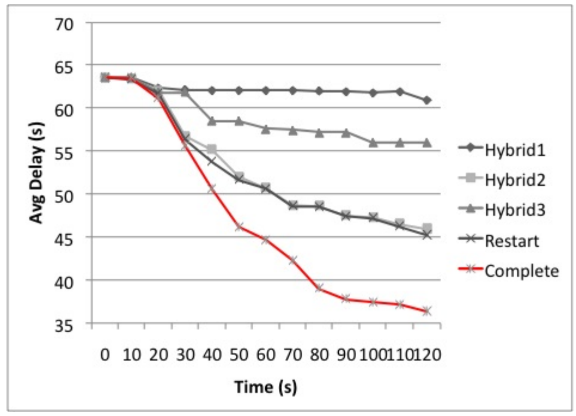

Figure 15: Average delay model 2, Malpensa

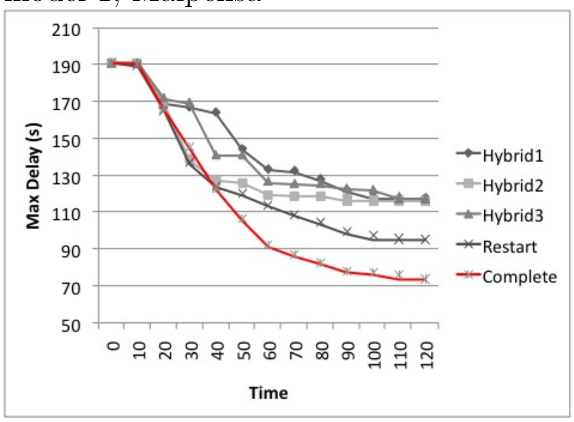

Figure 16: Maximum consecutive delay model 2, Malpensa

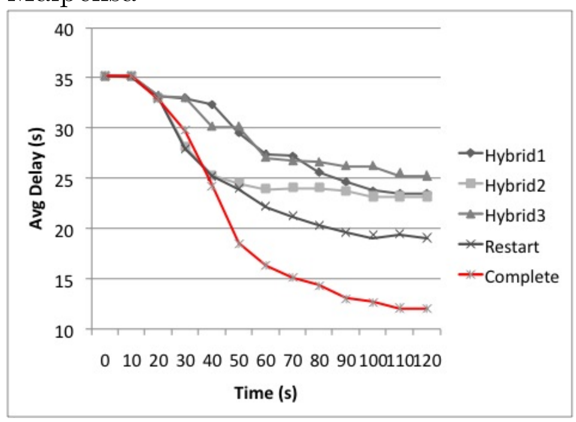

Figure 17: Average delay model 2, Malpensa

8 Experimental results

This section compares the results obtained by the best TS configurations of Section 7 with the performance of the scheduling solutions computed by BB with off-line routes. Both the algorithms have an execution time of 120 seconds. The comparison between TS and BB shows how rerouting alternatives, combined with aircraft retiming, can be used to get better solutions than the default routes.

Subsection 8.1 presents the results of Fiumicino airport, while Subsection 8.2 the result of Malpensa airport. Finally, Subsection 8.3 gives some examples of Gantt diagrams.

8.1 Fiumicino instances

Table 4 shows the results for the Fiumicino instances implemented with Model 1. Column 1 reports the algorithm tested, BB with off-line routes and TS with rerouting alternatives. Column 2 reports the time horizon for the instances, each row gives the average over 20 instances for time horizon. Column 3 reports the number of feasible solutions found.

Columns 4-6 report the maximum and the average consecutive delays (in seconds), and the number of delayed aircraft. Column 7 reports the value of Delta Travel Time Spent (DTTS), in seconds. For an aircraft a arriving/departing at/from a runway r, DTTS is equal to $t_{ar} - \tau_{ar}$, where τ_{ar} is the earliest possible time of a at r compatible with its current position and t_{ar} is its actual time at r in the schedule. This indicator is an interesting factor for energy consumption reasons.

Table 4: Rerouting Model 1, Fiumicino

ACDR Algo	Time Horiz	Feas Sched	Max Cons Delay (s)	Avg Cons Delay (s)	Delayed Aircraft	Total DTTS (s)
BB	15	20	117.5	51.4	3.5	558
TS	15	20	90.8	18.9	2.5	273.26
BB	30	20	350.8	125.9	12.8	3102.2
TS	30	20	335	121.1	14.2	2977.4
BB	45	20	562.8	168.4	22.9	5290.5
TS	45	18	469.5	140.8	21	4532.2
BB	60	20	1180.2	402.4	43.4	10045.8
TS	60	18	976.3	353.9	39.7	10658.9

In Table 4, TS outperforms the results of BB on the maximum consecutive delay minimization by 23% for 15-minute instances, by 4% for 30-minute instances, by 16% for 45-minute instances and by 17% for 60-minute instances. TS also improves the values of the other performance indicators. However, TS fails in finding a feasible solution in 2 cases for 45 minutes instances and in 2 cases for 60 minutes instances.

From additional experiments on the Fiumicino airport, the BB procedure achieved an average reduction of more than 40% of the maximum consecutive delay compared to FIFO (First In First Out). Such heuristic can be reasonably considered to be the scheduling rule used by air traffic controllers.

Further delay reduction can be achieved by the total enumeration algorithm for the ACDR problem, i.e., by executing the BB algorithm (with no time limit of computation) for all the combinations of aircraft routes. For the experiments of table 4 we found the optimal value of the maximum consecutive delay for the 15-minute traffic predictions, that is 78.9 seconds. For large instances, we were not able to compute the optimal solution in some hours of computation.

Table 5 shows the results for Fiumicino instances implemented with Model 2. We recall that in this case landing aircraft can be delayed twice, at the entry fix and at the runway (see Column 6).

In Table 5, TS outperforms the results of BB on the maximum consecutive delay minimization by 63% for 15-minute instances, by 21% for 30-minute instances, by 23% for 45-minute instances and by 60% for 60-minute instances. For this set of instances, TS also improves the value of the other performance indicators. A feasible solution is computed by TS for each delay configuration.

Regarding the total enumeration algorithm, we found the optimal value of the maximum consecutive delay for the 15-minute traffic predictions of the ACDR problem, that is 23.5 seconds.

Overall, for Fiumicino instances TS improves the solution found by BB for 156 instances out of 160. This set of computational results presents two important differences

Table 5: Rerouting Model 2, Fiumicino

ACDR Algo	Time Horiz	Feas Sched	Max Cons Delay (s)	Avg Cons Delay (s)	Delayed Aircraft	Total DTTS (s)
BB	15	20	70.95	10.24	2.75	582.6
TS	15	20	25.85	4.83	1.95	90.4
BB	30	20	302.25	74.89	21.65	1334.8
TS	30	20	239.75	64.65	16.25	1683.6
BB	45	20	257.2	63.11	28.7	2520.5
TS	45	20	196.85	46.98	18.95	1862.9
BB	60	20	1111.7	343.74	65.45	4632.6
TS	60	20	441.6	145.35	55.1	5543.6

between Model 1 and Model 2. The first one is that Model 2 always finds a feasible solution, while Model 1 experiences difficulty with the 45-minute and 60-minute instances. The second one is that we obtain significantly lower values of the objective function in Model 2 than in Model 1. This fact shows, together with the considerations on the number of feasible solutions, that Model 2 is easier to solve than Model 1.

In Model 2 aircraft can be delayed not only at the runway but also at the entrance of the TMA. For this reason it is important to understand where delay propagation is concentrated. Figures 18 and 19 show a detailed view on the performance of the rerouting algorithm for each instance. Each plot is done for the 80 timetable perturbation instances described in Section 6. The x-axis shows the instances ordered by increasing average entrance delay, while the y-axis reports the maximum or the average consecutive delays (in seconds), at the entrance of the TMA and at the runways. The average entrance delay ranges from 73 seconds (left-most point of Figures 18 and 19) to 506 seconds (right-most point of Figures 18 and 19).

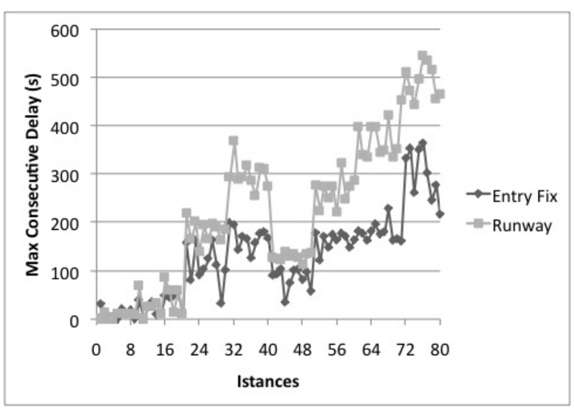

Figure 18: Maximum consecutive delays at entry fix and runway, Fiumicino

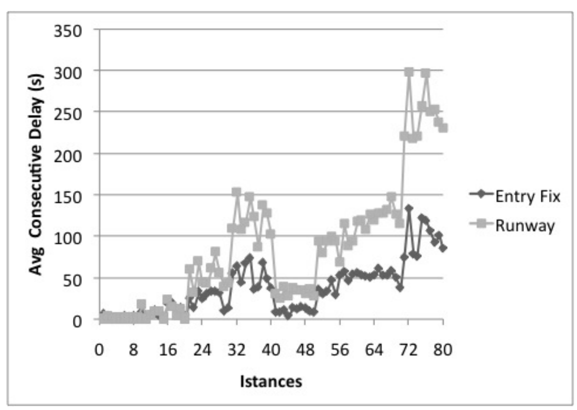

Figure 19: Average consecutive delays at entry fix and runway, Fiumicino

Despite entry fix and runway delays have a similar trend, it follows that the maximum consecutive delay concentrates more within the TMA rather than at the entry fix. This result is more evident for instances with a large entrance delay. Instances with Gaussian perturbation delay (approximately from 1 to 10, from 21 to 30, from 41 to 50 and from 61 to 70) show a smaller consecutive delay with respect to instances with uniformly distributed entrance delay.

8.2 Malpensa instances

Table 6 shows the performance of the BB and TS algorithms for the instances of Milano Malpensa implemented with Model 1. We report three rows for TS: TS (c) gives the results for rerouting actions allowed at air segments and runways (combined rerouting), TS (a) at air segments only (air rerouting), TS (r) at the runways only (runway rerouting).

Table 6: Rerouting Model 1, Malpensa

ACDR Algo	Time Horiz	Feas Sched	Max Cons Delay (s)	Avg Cons Delay (s)	Delayed Aircraft	Total DTTS (s)
BB	15	20	64.4	7.0	1.4	710.6
TS (c)	15	20	22.7	2.3	1.4	614.5
TS (a)	15	20	24.9	2.6	1.4	592.7
TS (r)	15	20	60.9	6.63	1.8	725.4
BB	30	20	139.1	24.1	5.8	1832
TS (c)	30	20	91.6	12.6	6.5	1586.7
TS (a)	30	20	127.8	22.1	7.4	186.8
TS (r)	30	20	94.9	14.1	6.8	1671.8
BB	45	20	305.9	61.6	13.2	3646.4
TS (c)	45	20	166	35.9	13.5	2987.8
TS (a)	45	20	234.9	57.9	14.9	3645.9
TS (r)	45	20	170.7	31.6	13.2	2900.6
BB	60	20	615.9	134.3	23.8	7271.5
TS (c)	60	20	303.6	106.5	24.7	6056.8
TS (a)	60	20	345.2	113.0	24.5	6445.6
TS (r)	60	20	314.5	153.8	23.5	5976.1

In Table 6, the three kinds of rerouting outperform the BB values for all the time horizons. TS (c), in particular, is the best configuration in all the cases. Regarding the other configurations, TS (r) obtains similar results to TS (c) for 30-minute, 45-minute and 60-minute instances. TS (a) obtains the worst results.

TS always obtains a feasible solution for all the instances. Compared to BB, TS (c) achieves an improvement on the maximum consecutive delay minimization by 65% for 15-minute instances, by 34% for 30-minute instances, by 46% for 45-minute instances and by 51% for 60-minute instances. The other indicators are improved too.

A comparison with the Fiumicino results, on the same model, reveals that the instances of Malpensa TMA are easier to solve than the Fiumicino instances. This is probably due to the existence of spare capacity for the Malpensa case that allows a more flexible management of traffic flow, even in presence of disturbances.

From additional experiments on the Malpensa airport, the BB procedure achieved an average reduction of more than 37% of the maximum consecutive delay compared to FIFO (First In First Out).

Table 7 reports the results for the Malpensa instances implemented by Model 2.

From the results of Table 7, the three kinds of rerouting settings always outperform the results of the BB procedure. TS (c) is the best configuration for 15-minute and 30-

Table 7: Rerouting Model 2, Malpensa

ACDR Algo	Time Horiz	Feas Sched	Max Cons Delay (s)	Avg Cons Delay (s)	Delayed Aircraft	Total DTTS (s)
BB	15	20	36.8	3.5	2.1	582.6
TS (c)	15	20	3.9	0.2	1.1	544.9
TS (a)	15	20	7.6	0.4	1.1	546.1
TS (r)	15	20	32.8	3.1	1.3	543.6
BB	30	20	92.4	8.1	5.9	1334.8
TS (c)	30	20	22.9	1.8	5.8	1235.1
TS (a)	30	20	88.8	7.7	6.8	1264.6
TS (r)	30	20	34.9	2.3	5.9	1211.9
BB	45	20	285.2	39.7	19	2520.5
TS (c)	45	20	75.1	9.0	11.15	1747
TS (a)	45	20	179.1	25.3	14.6	2174.8
TS (r)	45	20	72.9	8.1	10.9	1274.4
BB	60	20	343.3	88.4	40.25	4632.5
TS (c)	60	20	192.3	37.1	21.5	4186.1
TS (a)	60	20	273.4	57.5	26.4	4003.5
TS (r)	60	20	182.3	38.4	22.7	3316.5

minute instances, while TS (r) for the other ones. Compared to BB, TS (c) achieves an improvement on the maximum consecutive delay minimization by 89% for 15-minute instances, by 75% for 30-minute instances, by 74% for 45-minute instances and by 47% for 60-minute instances. TS (a) is again the worst of the three configurations.

Regarding the total enumeration algorithm, we also found the optimal value of the maximum consecutive delay of the overall problem for the 15-minute traffic predictions, that is 3.1 seconds.

Also for Malpensa instances, we obtain that the Model 2 gets the best performance in terms of the maximum consecutive delay minimization, compared to Model 1. The comparison between the two models shows that TS (c) obtains the best result for the instances with a lower number of alternative solutions (small time horizons). When the problem size increases and when we have a larger number of feasible rerouting possibilities, it is better to consider directly runways (large time horizons). However, TS (c) and TS (a) results are quite similar. This fact proves that runways are the most critical resources for the ACDR problem.

Figures 20 and 21 report the trend of the consecutive delay at the entrance of the TMA and at the runway. The average entrance delay ranges from 110 seconds (left-most point) to 495 seconds (right-most point). Also in this case, a growth of the entrance perturbation involves the concentration of the consecutive delays at the runways rather than at the entry fix. Since the traffic at Malpensa airspace is significant smaller than the airport capacity, there are a few conflicts and the consecutive delays are close to 0 for the first 28 instances. However, for average entrance delays larger than 225 seconds there is a significant increase on the number of conflicts to be solved, and therefore a kind of domino effect on the consecutive delays (i.e., stronger delay propagation).

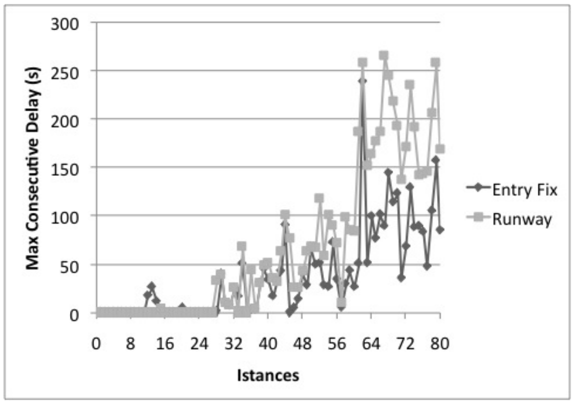

Figure 20: Maximum consecutive delays at entry fix and runway, Malpensa

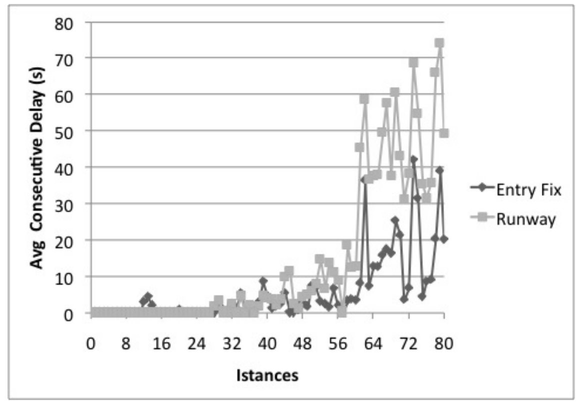

Figure 21: Average consecutive delays at entry fix and runway, Malpensa

8.3 Gantt charts

Gantt charts are a useful tool to show the solutions found by aircraft scheduling and routing algorithms. They may also be used by flight controllers to make TMA decisions. We next give an illustrative example for a 15-minute instance of Malpensa airport, with 6 landing aircraft and 5 departing aircraft.

Figure 22 shows three alternative routes for one of the landing aircraft of the instance. The route described by the solid line is the default route used by the BB algorithm. The other route is selected by the TS(c) algorithm, changing both the air path and the runway.

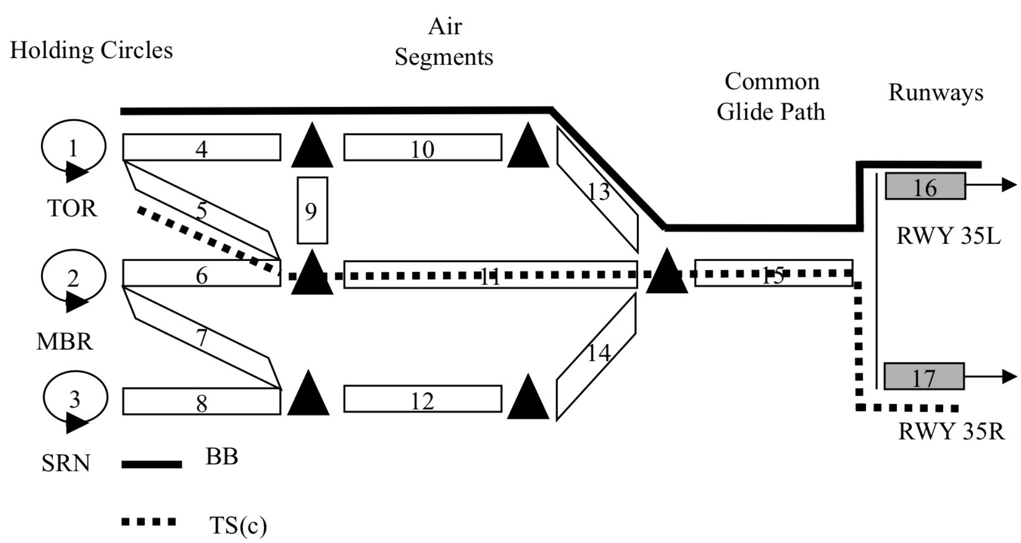

Figure 22: Different routes for an aircraft traversing the Malpensa airspace

Figure 23 describes the solution found by the BB algorithm. The critical resources are highlighted in yellow color. The maximum consecutive delay is 106 seconds. Specifically, the marked aircraft is the one of Figure 22.

Figure 24 describes the solution found by TS (c). The marked aircraft is routed on the air segments 5-11-15 instead of 4-10-13-15 and on the runway resource 17 instead of

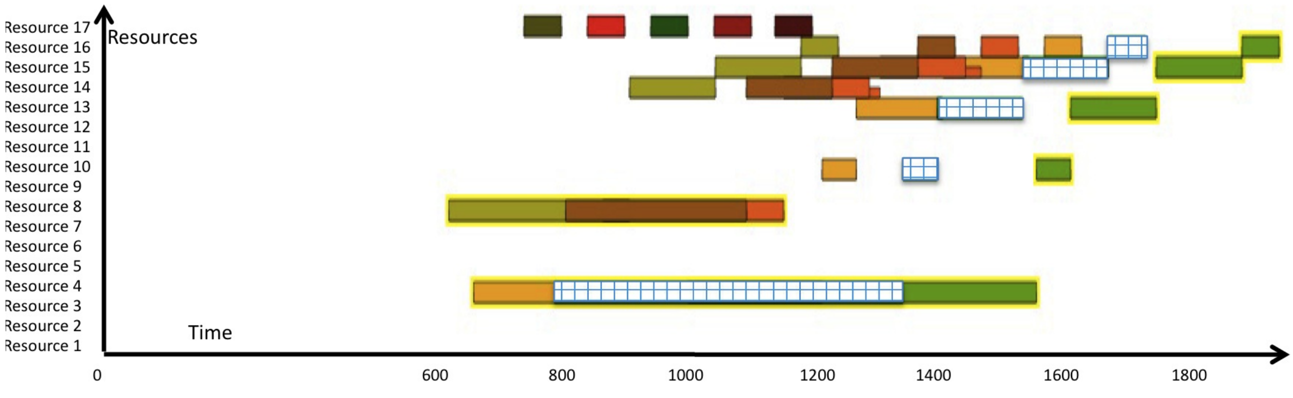

Figure 23: Gantt chart of the BB solution

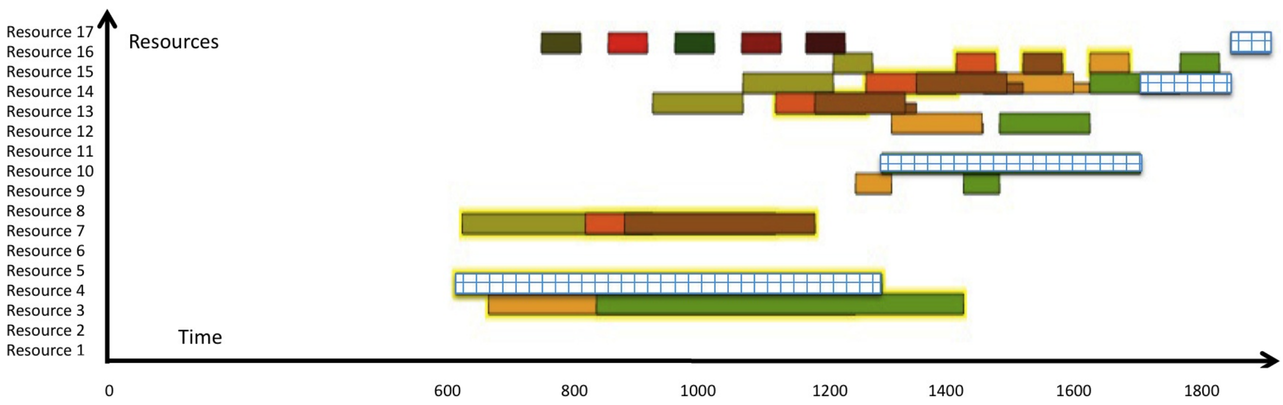

Figure 24: Gantt chart of the TS (c) solution

16. In the new solution the marked aircraft is postponed at the end of the schedule. The dark green aircraft is now scheduled before the marked one and is no more critical. The maximum consecutive delay decreases to 27 seconds.

9 Conclusions

This paper presents scheduling and rerouting algorithms for solving alternative graph formulations of the ACDR problem. Computational results for the FCO and the MXP airports demonstrate the effectiveness of our tabu search algorithm. TS is able to compute optimized aircraft timing and routing, including the possibility of rerouting aircraft in air and at runways. TS solutions are compared with the ones computed by BB with default routes. TS often outperforms BB in finding better quality solutions for the ACDR problem, also for large time horizons.

Results on MXP show that it is possible to obtain good solutions combining the rerouting on runways and air segments. However, runways are the most critical resources of the ACDR problem, specially for complex instances. For this reason, it is important to optimize the runway times and routes carefully. The results for both airports show that consecutive delays are larger for Model 1 than for Model 2. This is due to the major constraints given by the deadline arcs. However, the practical implementation of Model 2 (i.e., the ACDR instances without deadline constraints for the landing aircraft at the entry fix) requires active collaboration between the controllers of adjoining areas, coordinating the exit time from one airspace area with the entrance time in the next area.

Ongoing research is dedicated to the development of on-line decision support systems for air traffic control at TMAs and ATFMs. The scheduling and the rerouting algorithms presented in this work should be part of the system core. Specific research directions may consider the extension of our methodology to better deal with aircraft trajectory variations in the landing/departing procedure. Another interesting research topic regards the assessment of traffic control measures in presence of even more disturbed traffic situations [10].

Acknowledgments

This work is partially supported by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB "Advanced tracking system in intermodal freight transportation". Preliminary results will be presented at the 2nd International Conference on Models and Technology for Intelligent Transportation Systems [14].

References

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

[2] Andreatta, G. and Romanin-Jacur, G. (1987). Aircraft flow management under congestion. *Transportation Science* **21(4)** 249–253.

[3] Ball, M.O., Barnhart, C., Nemhauser, G., Odoni, A. (2007) Air Transportation: Irregular Operations and Control. In: G. Laporte and C. Barnhart (Eds.), *Handbooks in Operations Research and Management Science* **14(1)** 1–67.

[4] Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D. (2000) Scheduling aircraft landings – The static case. *Transportation Science* **34(2)** 180–197.

[5] Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D. (2004) Displacement problem and dynamically scheduling aircraft landings. *Journal of Operational Research Society* **55** 54–64.

[6] Bertsimas, D., Lulli, G., Odoni, A. (2011) An Integer Optimization Approach to Large-Scale Air Traffic Flow Management. *Operations Research* **59(1)** 211–227.

[7] Bianco, L., Dell'Olmo, P., Giordani, S. (2006) Scheduling models for air traffic control in terminal areas. *Journal of Scheduling* **9(3)** 180–197.

[8] Castelli, L., Pesenti, R., Ranieri, A. (2011) The design of a market mechanism to allocate Air Traffic Flow Management slots. *Transportation Research Part C* **19(5)** 931–943.

[9] Churchill, A.M., D.J. Lovell, Ball, M.O. (2010) Flight Delay Propagation Impact on Strategic Air Traffic Flow Management. *Transportation Research Record* **2177** 105–113.

[10] Clausen, J. (2007) Disruption Management in Passenger Transportation - from Air to Tracks. *Proceedings of the 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS 2007)*. In C. Liebchen, R.K. Ahuja (Eds), Sevilla, Spain.

[11] Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M. (2010) A tabu searc algorithm for rerouting trains during rail operations, *Transportation Research Part B* **44(1)** 175–192.

[12] D'Ariano, A., Corman, F., Pacciarelli, D., Pranzo, M. (2008) Reordering and local rerouting strategies to manage train traffic in real-time, *Transportation Science* **42(4)** 405–419.

[13] D'Ariano, A., D'Urgolo, P., Pacciarelli, D., Pranzo, M. (2010) Optimal sequencing of aircrafts take-off and landing at a busy airport, *Proceedings of the 13th IEEE Conference on Intelligent Transportation Systems*, Madeira Island, Portugal.

[14] D'Ariano, A., Pistelli, M., Pacciarelli, D. (2011) Aircraft retiming and rerouting in vicinity of airports, *Proceedings of the 2nd International Conference on Models and Technology for Intelligent Transportation Systems*, Leuven, Belgium.

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

[16] Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F. (1997) Daily Aircraft Routing and Scheduling, *Management Science* **43(6)** 841–855.

[17] Ernst, A.T., Krishnamoorthy, M., Storer, R.H. (1999) Heuristic and exact algorithms for scheduling aircraft landings. *Networks* **34(3)** 229–241.

[18] Eun, Y., Hwang, I., Bang, H. (2010) Optimal Arrival Flight Sequencing and Scheduling Using Discrete Airborne Delays. *IEEE Transactions on Intelligent Transportation Systems* **11(2)** 359–373.

[19] Ganji, M., Lovell, D.J., Ball, M.O., Nguyen, A. (2009) Resource allocation in flow-constrained areas with stochastic termination times. *Transportation Research Record* **2106** 90–99.

[20] Kuchar, J.K., Yang, L.C. (2000) A Review of Conflict Detection and Resolution Modeling Methods. *IEEE Transactions on Intelligent Transportation Systems* **4(1)** 179–189.

[21] Mascis, A. and Pacciarelli, D. (2002) Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research* **143(3)** 498–517.

[22] Psaraftis, H.N. (1980) A dynamic programming approach for sequencing identical groups of jobs. *Operations Research* **28(6)** 1347–1359.

[23] Venkatakrishnan, C.S., Barnett, A., Odoni, A.M. (1993) Landings at Logan airport: describing and increasing airport capacity. *Transportation Science* **27(3)** 211–227.

[24] Zhan, Z-H., Zhang, J., Li, Y., Liu O., Kwok, S.K., Ip, W.H., Kaynak, O. (2010) An efficient Ant Colony System based on receding horizon control for the Aircraft Arrival Sequencing and Scheduling Problem. *IEEE Transactions on Intelligent Transportation Systems* **11(2)** 399–412.