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

A Graph-Theoretical Model for the Aircraft Sequencing Problem

Ludovica Adacher, Dario Pacciarelli, Marco Pranzo

RT-DIA-79-2003 **May 2003**

adacher@dia.uniroma3.it, pacciarelli@dia.uniroma3.it, mpranzo@dia.uniroma3.it

ABSTRACT

The main purpose of optimized sequencing of landing/taking-off operations is to obtain a better exploitation of the existing airport infrastructure, thus improving the performances of the entire ATC system. In this paper we deal with innovative sequencing models for managing arrivals and departures times of the aircrafts at an airport. These models are based on the so-called alternative graph formulation, which resulted very effective for modeling and solving the traffic control problem of large railway networks. We show that the alternative graph model allows a detailed formulation of the air traffic control problem.

1 Introduction

With the increase in air traffic, airports are becoming a major bottleneck in Air Traffic Control (ATC) operations. Airports have a limited structural capacity, which cannot be increased without facing many serious problems (economical, environmental, safety rules, etc.). Aviation authorities are therefore seeking methods to better exploit the existing infrastructure, while maintaining the current level of safety. This paper deals with innovative sequencing procedures for managing arrivals and departures times for the aircrafts. It is well known, in fact, that a good sequencing of landing/taking-off operations could substantially improve the performances of the entire ATC system.

We refer to the problem of finding a good sequence of landing/taking-off for the aircrafts as the Aircraft Sequencing Problem (ASP). The ASP has been the subject of several papers (see, e.g., [15, 16, 3]). However, especially from the practical point of view, most of the early contributions suffered for a substantial lack of information due to the usage of very simplified models. Recent models tend to exhibit an increased level of realism and to incorporate a larger variety of constraints and possibilities, such as sequence dependent processing times, no-wait constraints, and earliness/tardiness penalties for each aircraft.

We distinguish two cases of ASP. In the *static* ASP one wants to sequence landing/departing aircrafts when all the information is know in advance. Beasley et al. [3] propose a mixed-integer zero-one formulation for the static ASP in the single and multiple runways cases. The problem is then solved using an exact and a heuristic algorithm. Ernst et al [8] tackle the static ASP of aircraft landings for the single runway case and extend it to the multiple runway case. They propose a specialized simplex algorithm and use it in a branch and bound and in a heuristic algorithm.

In the *dynamic* ASP aircrafts enter the system one at the time. Hence, a new sequence of landing/taking-off aircrafts has to be recomputed every time a future incoming aircraft is known. In such cases, a common solution approach consists of constraining the set of feasible positions in the sequence for the new aircraft to avoid aircrafts to be excessively delayed [4, 5, 6, 7]. With this constraint, usually called Constrained Position Shifting (CPS), no aircraft can be sequenced forward and rearward more than a specified number of positions from its position computed according to the First Come First Served (FCFS) discipline. CPS constraint is also used in the static ASP. For example, Psaraftis [15] develops an exact algorithm for this case, which makes use of the CPS concept.

We next define more specifically the ASP, the subsequent section is devoted to the definition of the alternative graph formulation. Then we formulate the ASP by means of the alternative graph formulation and show in particular an example of Air Traffic Control at the Rome-Fiumicino (FCO) airport. Some conclusions follow.

2 The aircraft scheduling problem

Air Traffic Control decisions can be broadly divided into two classes of subproblems:

- Routing decisions, where a path for each aircraft has to be determined from its current position to its destination (see, e.g. [1]).
- Sequencing decisions, where paths are considered fixed, and a feasible sequence of aircrafts has to be found for each fixed point, which satisfies all the constraints of the problem and optimizes a given system performance index.

To a large extent, these two problems are independent. In other words, it is possible to determine the optimal flight paths first, and then to solve the sequencing problem.

We next briefly recall the general procedure for landing/taking-off operations. For each Terminal Maneuvering Area (TMA) landing/departing aircrafts move along predefined paths from a fixed point to a runway and vice versa. Moreover, every aircraft flies along the common approach path following a standard descent profile and during all the approach phases a minimum separation distance between every pair of consecutive aircrafts must be guaranteed. The runway can be occupied by only one aircraft at a time. Similarly, departing aircraft must leave the runway flying towards the assigned fix along an ascent profile, respecting separation standards.

A further typical safety rule is that every pair of aircrafts must maintain a minimum separation distance also when moving on adjacent paths (at the same or different altitude), this separation depending on the type and relative positions of the two aircrafts. By considering the different aircraft speeds, this safety distance can be translated in a Separation Time Interval (STI). These separations are mandatory and recognized by international aviation regulations.

Summarizing the above discussion, the ASP can be stated as follows: given a set of aircrafts willing to land/take-off, and given for each aircraft the approach/leaving path, the current speed, the runway occupancy time, and a time slot to accomplish the landing (departing) procedures, assign to each aircraft the starting time from the fix (runway) in such a way that all the constraints are satisfied and a given system performance index is optimized. Typical objective functions are the minimization of the maximum arriving/departing time or the minimization of the average delay.

The ASP can be formulated as a job shop scheduling problem with additional constraints [5, 6], where a job represents a landing/departing aircraft. By dividing paths into smaller air segments, the landing or departing phases for an aircraft can be represented by a sequence of operations, each one corresponding to the traversing of a segment by an aircraft. Each segment can be therefore viewed as a machine, while the traversing of a segment can be represented as an operation of the job (aircraft) on that machine (segment). The processing time of the operation is equal to the traversing time of the air segment, which depends on the aircraft characteristics.

Clearly, in such representation, each operation has to be executed without interruption, and there are additional no-wait constraints between the traversing of subsequent segments. The additional constraints on the separation time intervals can be represented as sequence-dependent set-up times.

3 The alternative graph formulation

While the job shop problem is a clean problem, ASP is rich of additional constraints, such as no-wait constraints, departure and arrival times, sequence dependent setup times, etc.

The *alternative graph* formulation, first introduced in Mascis and Pacciarelli [9, 10], is an effective model for studying complex scheduling problems, arising in manufacturing [14] as well as in railway traffic control [12]. With this formulation the variables of the problem are the starting times of the operations, i.e. the time at which a given aircraft enter a given segment. We denote by t_i the starting time of operation i, $i = 1, \dots, n$.

There is a set of precedence relations among operations. A *precedence relation* (i, j)

means that the starting times of the j-th operation must be greater or equal to the starting time of the i-th operation plus a given *delay* f_{ij}, which in our model can be either positive, null or negative. Precedence relations are divided into two sets: *fixed* and *alternative*. Alternative precedence relations are partitioned into pairs, and are used to represent the sequencing decisions.

The alternative graph formulation can be viewed as a linear program with logical conditions "or" ($\vee$, disjunction), as in Balas [2].

$$\begin{array}{lll} \min & t_n - t_0 & \\ s.t. & t_j - t_i \geq f_{ij} & (i,j) \in F \\ & (t_j - t_i \geq a_{ij}) \vee (t_k - t_h \geq a_{hk}) & ((i,j),(h,k)) \in A \end{array}$$

The alternative graph is described by the triple $\mathcal{G} = (\mathcal{N}, \mathcal{F}, \mathcal{A})$. $N = \{0, 1, \ldots, n+1\}$ is the set of nodes of the graph. While nodes $1, \ldots, n$ are associated to the operations of the problem, 0 and $n+1$ are dummy operations called *start* and *finish* respectively. F is a set of directed arcs and A is a set of pairs of directed arcs. Arcs in the set F are *fixed*. Arcs in the set A are *alternative*. If $((i,j),(h,k)) \in A$, we say that (i,j) and (h,k) are paired and that (i,j) is the alternative of (h,k). In our model the arc length can be positive, null or negative, in order to represent real world constraints.

In our formulation of the ASP, the minimization of the starting time of the *finish* node corresponds to the minimization of the maximum delay for all aircrafts.

4 Formulation of the aircraft scheduling problem

In this section we illustrate some examples of alternative graphs associated with typical constraints arising in ASP.

4.1 Runway constraint

The runway constraint takes into account the safety rule on runway, which can be occupied by at most one aircraft at a time. We model this constraint with a pair of alternative arcs. More in detail, let us now consider two aircrafts using the same runway. Let i (resp. j) be the operation associated to the traversing of the runway for an aircraft, and $\sigma(i)$ (resp. $\sigma(j)$) be the subsequent operation for that aircraft. We associate with i and j a pair of alternative arcs. Each arc represents the fact that one operation must be processed before the other. If i is processed before j, the runway can host j only after the starting time of the operation $\sigma(i)$, when i leaves the runway. We model this situation with the alternative arc $(\sigma(i), j)$, as in Figure 3. The weight e on the alternative arc takes into account the safety constraint STI, i.e., the constraint that the runway becomes available for j only e time after the starting time of $\sigma(i)$. Similarly, if j is processed before i, we have the alternative arc $(\sigma(j), i)$.

4.2 Separation time interval on an air segment

An air segment can be represented as a resource with multiple capacity in which two consecutive aircrafts must maintain a fixed STI. Since the overtaking is not allowed within

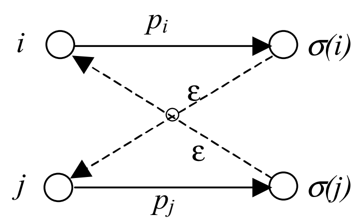

Figure 1: The runway constraint.

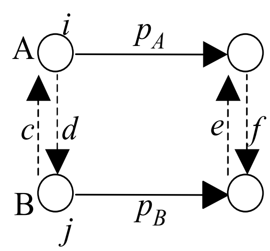

Figure 2: The alternative graph model for STI constraint on an air segment.

the same path, we model this constraint with two alternative pairs (c, f) and (d, e), as in Figure 2.

In this case for each pair of aircrafts (A and B in the figure), and for each air segment, two pairs of alternative arcs must be inserted, namely the pair (c, f) and (e, d). The length of such arcs is precisely the STI between the aircrafts at the entrance/exit of the air segment. For example, if aircraft A precedes aircraft B at the entrance of the air segment, then B must enter at least d time units after A, and therefore must exit from the air segment at least f time units after A. Note that in a feasible solution the arcs c and d or e and f cannot be selected, otherwise there would be a positive length cycle in the resulting graph. Therefore, in a feasible solution, either d and f or c and e must be selected, thus satisfying the no overtaking constraint.

4.3 Speed constraint

A flying aircraft cannot stop within a segment, or between consecutive air segments. We model the more general case, in which a minimum and the maximum processing times p_i and $p_i + d_i$, respectively, are specified for completing operation i, i.e. we give the chance to an aircraft of varying its speed within given margins. A tight no-wait operation is obtained when $d_i = 0$. Figure 3 gives a graphical representation of this constraint.

4.4 Release time

Some operations have associated a release time, such as for example the departure time of an aircraft. A release time r_i means that operation i cannot be executed until time r_i. This can be easily modeled by adding an arc $(0, i)$ from the start operation to operation

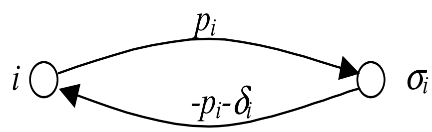

Figure 3: The speed constraint.

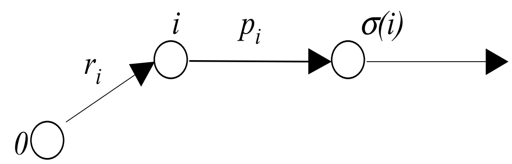

Figure 4: Release time of an operation.

i, weighted with the release time r_i, as shown in Figure 4.

4.5 Time-slot constraint

The time-slot constraint $[e_i, l_i]$ for an operation i, means that the starting time of i must belong to an earliest/latest possible start time window. This is modeled with two fixed arcs $(0, i)$ and $(i, 0)$ with weights e_i and $-l_i$, respectively, with $l_i \geq e_i$. An example of alternative graph representing a time-slot constraint is shown in Figure 5.

4.6 An example: Modeling an airport

Figure 6(a) shows the model of the Rome-Fiumicino (FCO) airport. The airport is composed by three runways. Two of them are intersecting and used for departing and landing, respectively. The minimal longitudinal separation between two consecutive aircraft flying along the same trajectory is dependent on the aircraft types. The minimal diagonal separation between two aircraft traveling in parallel dependent path is independent of the

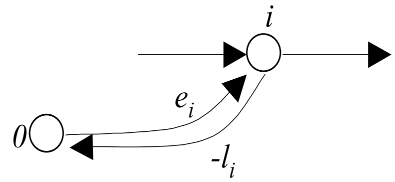

Figure 5: The time-slot constraint.

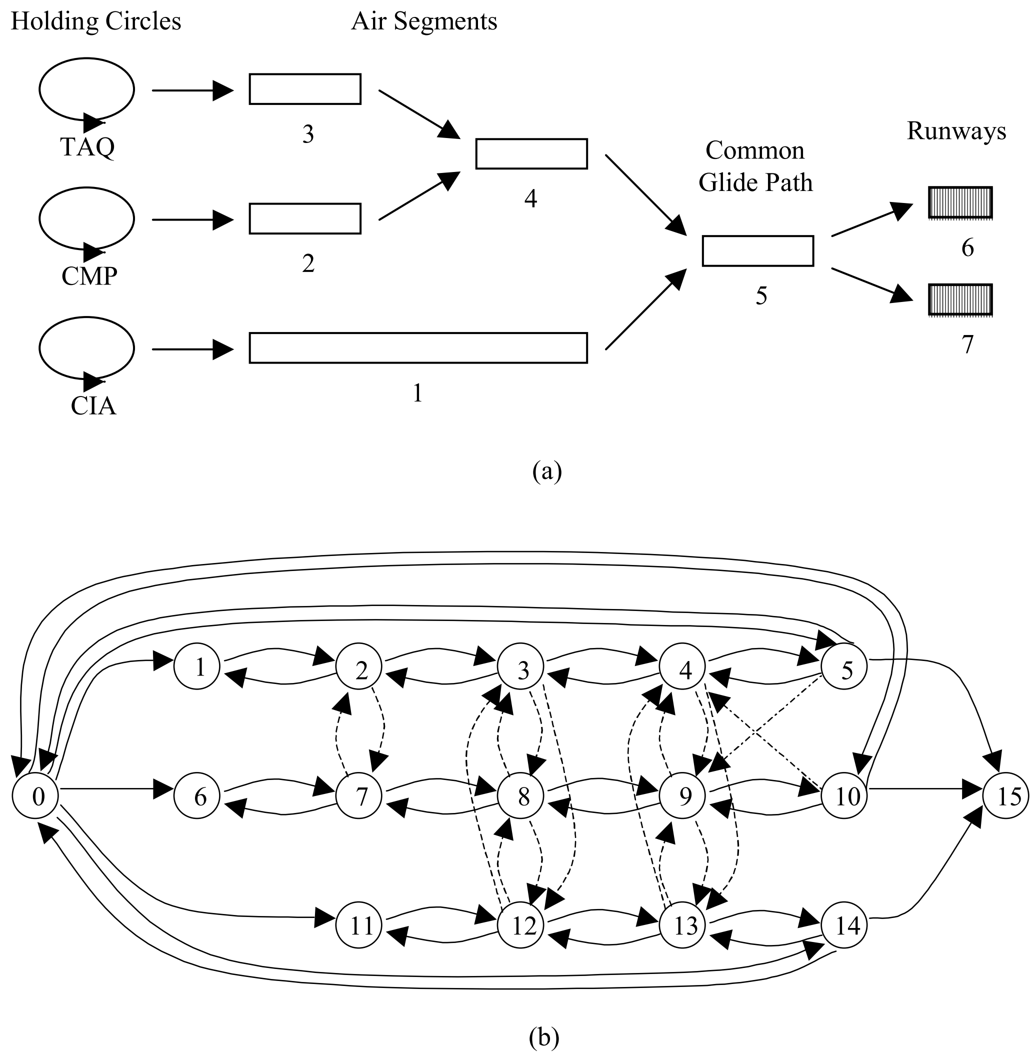

Figure 6: The FCO airport model and the associated alternative graph with three landing aircrafts.

aircraft category.

We model the FCO airport as in [6], with the following machines.

- The three Holding Circles are not modeled explicitly, since they are regarded as an infinite capacity input buffer.
- The landing paths are modeled using five air segments, and imposing a minimum separation between consecutive aircrafts. In particular, we use four independent segments to model the first part of the approach paths. The final parts of the approach paths (common glide paths) are dependent because a diagonal separation between aircrafts landing on parallel runways has to be observed. For this reason the common glide paths can be modeled as a single segment, with different separation constraints between consecutive landing aircrafts. The time lag between a pair of aircrafts equals the diagonal separation if the aircrafts move on different glide paths,

while it equals the longitudinal separation for aircrafts moving on the same path.

- The runways constraints are used for the last operation of each aircraft. Note that, the two intersecting runways cannot be used simultaneously, and therefore we model them using only one machine. Hence, the FCO airport can be modeled as a job shop problem with seven machines. Machines from 1 to 5 are air segments, while machines 6 and 7 are runways.

We next show the alternative graph model for three aircrafts landing at Fiumicino airport. Aircraft A is a heavy type and it is entering in the CMP holding circle. The aircraft B is a heavy type and it is approaching the FCO airport from TAQ holding circle, whereas aircraft C is a medium type and it is approaching from the CIA holding circle. Each aircraft has its release time, which is the starting time from which the aircraft is available for beginning the landing procedure. Moreover, for each aircraft a non-relaxable time window is given for accomplishing the landing procedure in the assigned time slot.

Figure 6(b) illustrates the alternative graph for the three landing aircraft example. For sake of clarity we indicate each node of the alternative graph by the pair (aircraft, machine). Each alternative pair of arcs is associated to the usage of some common resource. In particular, aircrafts A and B share resource 5, and resource 6. Aircrafts A and C share resources 4 and 5. Aircrafts B and C share resource 5.

The pairs of fixed arcs with weights e_i and $-l_i$ represent the time-slot constraint for aircrafts A, B, and C. Note that no feasible solution exists in which one of the aircrafts accomplishes the landing procedure later than l_i or earlier than e_i.

We observe that the same model can be easily extended to represent in details, not just a single airport, but also an entire air sector, composed of several fixed points, thus allowing to formulate and solve a real air traffic control problem.

5 Conclusions and future research

In this paper we introduced the alternative graph model of ASP. The proposed model could be used as a scheduling layer, and integrated in a Decision Support System (DSS). Then, solution algorithm similar to the ones used by Pacciarelli et al. [11, 13] for the train scheduling problem could be used to develop advanced DSS for aiding air traffic controllers.

References

[1] Adacher L., M. Lucertini, C. Meloni, 2002, A decentralized approach to air traffic flow management, *Proceedings of the advanced workshop on ATM system architectures and CNS technologies needed to cope with the air traffic capacity problem and related evaluation tools (ATM'02)*, Capri, Italy, 133–141.

[2] Balas E., 1979, Disjunctive programming, *Annals of Discrete Mathematics*, **5**, 3–51.

[3] Beasley J.E., M. Krishnamoorthy, Y.M. Sharaiha, D. Abramson, 2000, Scheduling aircraft landings - the static case, *Transportation Science*, **34**, 180–197.

[4] Bianco L., P. Dell'Olmo, S. Giordani, 1997, Scheduling models and algorithms for TMA traffic management, *Modelling and simulation in air traffic management*, eds. Bianco L. et al., Springer Verlag, 139–168.

[5] Bianco L., P. Dell'Olmo, S. Giordani, 1999, *Coordination of the traffic flows in the TMA, Proceedings of the workshop advanced technologies and their impact on air traffic management in the 21th century (ATM'99)*, Capri, Italy, 110–119.

[6] Bianco L., P. Dell'Olmo, S. Giordani, 2001, Aircraft Flow Scheduling in the Terminal Maneuvering Area, *Proceedings of the Triennial Symposium on Transportation Analysis (TRISTAN IV)*, Sao Miguel, Portugal, 281–286.

[7] Dear R.G., Y.S. Sherif, 1991, An algorithm for computer assisted sequencing and scheduling of terminal area operations, *Transportation Research A*, **25**, 129–139.

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

[9] Mascis A., D. Pacciarelli, 2000, *Machine Scheduling via Alternative Graphs*, Report DIA-46-2000, Dipartimento di Informatica e Automazione, Università Roma Tre, Roma, Italy.

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

[11] Mascis A., D. Pacciarelli, M. Pranzo, 2001, Train scheduling in regional railway networks, *Proceedings of the Triennial Symposium on Transportation Analysis (TRISTAN IV)*, Sao Miguel, Portugal, 487–492.

[12] Mascis A., D. Pacciarelli, M. Pranzo, 2002, *Models and Algorithms for traffic management of rail networks*, Report DIA-74-2002, Dipartimento di Informatica e Automazione, Università Roma Tre, Roma, Italy.

[13] Mascis A., D. Pacciarelli, S.J. van de Stouwe, 2001, DSS for real-time traffic optimization, *Proceedings of the 5th World Congress on Railway Research (WRCC 2001)*, Cologne, Germany.

[14] Pacciarelli D., M. Pranzo, 2000, *Production Scheduling in a Steelmaking-continuous Casting Plant*, Report DIA-50-2000, Dipartimento di Informatica e Automazione, Università Roma Tre, Roma, Italy.

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

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