Scheduling Arrivals and Departures in a Busy Airport

Ludovica Adacher* Dario Pacciarelli* Danilo Paluzzi* Marco Pranzo*

*Dipartimento di Informatica e Automazione, Università degli Studi Roma Tre
Via della Vasca Navale 79
00146, Roma, Italy
{adacher, pacciarelli, paluzzi, mpranzo}@dia.uniroma3.it

1 Introduction

With the increase in air traffic, airports are becoming a major bottleneck in Air Traffic Control (ATC) operations. 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. We refer to the problem of finding a 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., [6, 7, 1, 2]), and it 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, safety separation distances and a time slot to accomplish the landing (departing) procedures, find a schedule for each aircraft such that all the constraints are satisfied and a given system performance index is optimized. Typical objective functions are the minimization of the maximum delay or the minimization of the average delay.

2 The alternative graph formulation

The ASP can be formulated as a job shop scheduling problem with additional constraints [2], where each job corresponds to a landing/departing aircraft, and each air segment corresponds to a machine. In such representation, the processing time of an operation equals the traversing time of the associated air segment, and each operation has to be executed without interruption. There are additional no-wait constraints between the traversing of subsequent segments, and other constraints on the separation time intervals between consecutive aircrafts, which can be represented as sequence-dependent set-up times.

The *alternative graph* formulation [3], is an effective model for studying complex scheduling problems [4]. With this formulation the variables of the problem are the starting times t_i of the operations, $i = 1, \ldots, n$. 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}. 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 is described by the triple $\mathcal{G} = (N, F, A)$. $N = \{0, 1, \ldots, n+1\}$ is the set of nodes of the graph. While nodes $1, \ldots, n$ are associated to the operations, 0 and $n+1$ are dummy operations called *start* and *finish* respectively. F is a set of directed arcs (*fixed*) and A is a set of pairs of directed arcs. Arcs in the set A are called *alternative*. If $((i, j), (h, k)) \in A$, we say that arc (i, j) is the alternative of arc (h, k). A solution is obtained by choosing exactly one arc from each alternative pair, thus obtaining a set S of $|A|$ alternative arcs. A solution is feasible if the graph $\mathcal{G} = (N, F \cup S)$ has no positive length cycles. In our model, in order to represent real world constraints, the arc length can be positive, null or negative. In our formulation of the ASP, the minimization of the maximum delay for all aircrafts corresponds to the minimization of the starting time associated to the *finish* node.

3 Formulation of the aircraft scheduling problem

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

Runway constraint: The runway can host only one aircraft at the time. We model this constraint with a pair of alternative arcs for each pair of aircrafts willing to use the runway, as in Figure 1 (a). 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. If i is processed before j, j can start the landing/take-off only after the starting time of the operation $\sigma(i)$, when i leaves the runway. We model this situation with the alternative arc $(\sigma(i), j)$, the length e on the alternative arc being a safety time distance between i and j. Similarly, if j is processed before i, we have the alternative arc $(\sigma(j), i)$.

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 safety distance. Since near the airport overtaking is not allowed, we model this constraint with two alternative pairs (c, f) and (d, e), as in Figure 1 (b). 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 equals the safety time distance between the entrance/exit of the aircrafts into/from the air segment. For example, if aircraft A precedes aircraft B at the entrance of the air segment, then in a feasible solution 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.

Speed constraint: A flying aircraft cannot stop within a segment, or between consecutive air segments. We model the more general case, in which an aircraft may vary the traversing time of an air segment within given margins p_i and $p_i + \delta_i$, respectively, i.e. $t_i + p_i \leq t_j \leq t_i + p_i + \delta_i$. A tight no-wait operation is obtained when $\delta_i = 0$. Figure 1 (c) gives a graphical representa-

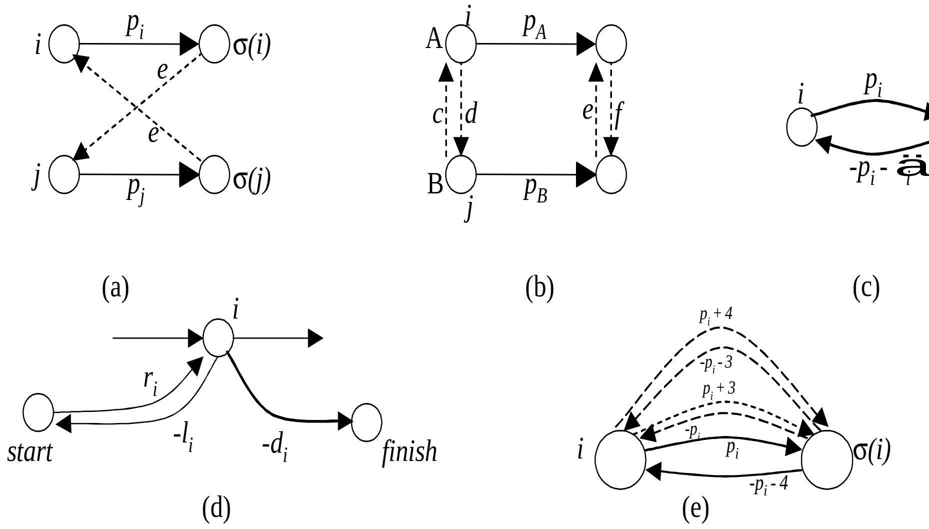

Figure 1: (a) Runway – (b) Air segment – (c) The speed constraint – (d) Time-slot constraints – (f) Holding circle

tion of this constraint.

Time-slot constraints: The time-slot constraints for an operation i require that the starting time of i belongs to a given time window, i.e. $r_i \leq t_i \leq l_i$. The earliest starting time (*release time*) is modeled with a fixed arc $(0, i)$ with length r_i; the latest possible starting time (*deadline*) is modeled with a fixed arc $(i, 0)$ with length $-l_i$. A different case arises when dealing with *due dates*. In fact, while in a feasible solution deadlines cannot be violated, the due dates can be exceeded thus causing a delay. The goal of the algorithm then in this case is to minimize the delays. A due date d_i is modeled with a fixed arc $(i, n+1)$, with length $-d_i$. An example of alternative graph representing time-slot constraints is shown in Figure 1 (d).

Holding circles: The holding circle permits to an aircraft to wait some time before starting the landing procedure. An aircraft can leave an holding circle only after the traversing of (or a multiple of) half length of the circle. Let t_{c1} be the time needed to perform one circle, and t_{c2} the time needed to perform half circle. The alternative graph model is represented in Figure 1 (e), where an alternative pair $((i, \sigma(i)), (\sigma(i), i))$ is defined for each circle. The lengths of the arcs are defined in such a way that, the starting time of operation $\sigma(i)$ is $t_{\sigma(i)} = t_i + c$, where c is the time spent inside the holding circle, given by the expression $c = nt_{c1} + mt_{c2}$, and where the integer n and $m \in \{0, 1\}$ represents the number of circles plus one possible half circle, traversed by the aircraft, respectively.

Modeling FCO airport: Figure 2(a) shows the model of the international Rome-Fiumicino (FCO) airport. At FCO there are three holding circles (TAQ, CMP, CIA). The landing paths are modeled using 8 air segments, and imposing a minimum separation between consecutive aircrafts. In particular, we use 7 independent segments to model the first part of the approach paths. At FCO there are two common glide paths that are interdependent because a diagonal

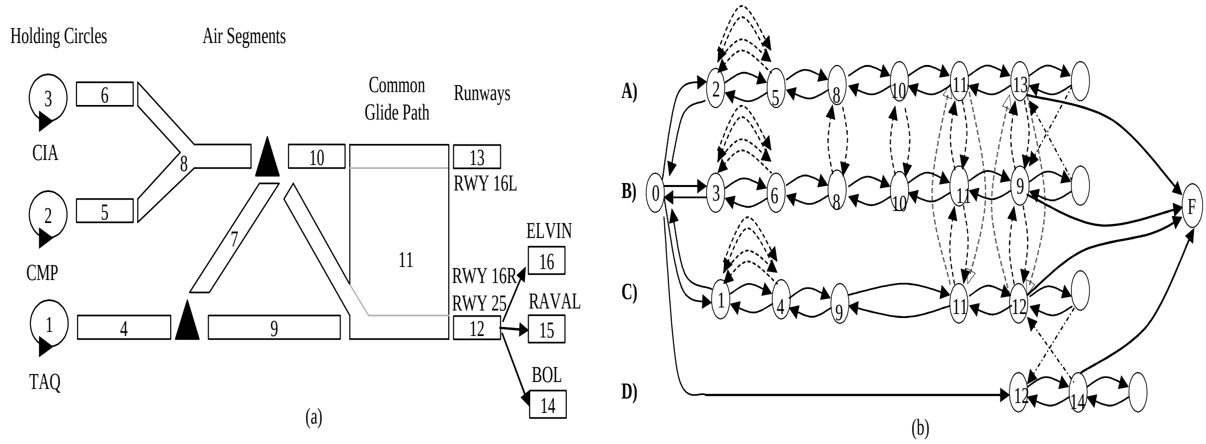

Figure 2: The FCO airport model and the associated alternative graph with three landing aircrafts, and a deperting aircraft.

separation between aircrafts landing on parallel runways has to be observed. For this reason, we model the common glide paths as a single segment. The runways constraints are used for the last operation of each aircraft. Note that, at FCO there are two intersecting runways, one of which is used for landing and the other for taking-off. These two runways cannot be used simultaneously, and therefore we model them using only one machine (numbered as machine 12 in Figure 2 (b). Three air segments are used to represent the three departure paths toward three exit fix (BOL, ELVIN, RAVAL).

We next show the alternative graph model for 4 aircrafts, where A, B and C are landing from CMP, TAQ and CIA, respectively. Aircraft D is taking-off toward BOL fix.

4 Computational experiences

Experiments are carried out considering as a reference test case one busy hour of arrivals and departures at the FCO international airport. 48 aircrafts of different types and 4 time slots are considerated, with the objective of minimizing the maximum delay. We compare the performance in terms of maximum delay minimization of three greedy rules: The FIFO is a commonly used dispatching rule, while the AMSP (Avoid Most Similar Pair) and the AMBP (Avoid Most Balanced Pair) are two different greedy described in [5]. The experiments run on a Pentium II 350MHz processor and a single run of the AMSP requires less than one minute.

Starting from the reference test case with 48 aircrafts, we generated other instances by perturbing the initial delay of same aircraft. Table 1 reports the results obtained for two of such instances, besides the reference case. In particular, the two perturbed instances are obtained by increasing the input delay of aircraft 9 and 30 of an additional delay of 60, 180 and 300 seconds, respectively, thus obtaining 6 perturbed instances. Aircraft 9 and 30 are two aircrafts obtaining the maximum delay in the reference test case when using the FIFO and the AMSP/AMBP rules respectively. We decided to delay the critical aircraft, i.e. the aircraft which causes the value of the objective function. As shown in Table 1, in the reference case,

	Reference Case	aircraft 9 delayed			aircraft 30 delayed		
		60 sec.	180 sec.	300 sec.	60 sec.	180 sec.	300 sec.
FIFO	209	77	462	513	8	8	108
AMSP	-113	–	-191	-48	-230	-173	7
AMBP	-113	-168	-168	-48	-168	-113	7

Table 1: Effects of input delay on critical aircrafts

the FIFO rule is not able to devise a solution which allows all the aircrafts to land and depart on time; on the other hand both AMSP and AMBP are able to achieve a solution with -113 seconds of lateness. In all perturbed instances the FIFO rule is clearly outperformed by the other two heuristics which are able to recover up to 5 minutes of input delay.

We observe that all these greedy heuristics may fail in finding a feasible solution since the problem is heavily constrained. Besides this the presence of the holding circles may cause an instable behavior. In fact, it may happen that increasing the delay of an aircraft the maximum delay of the solution decreases.

5 Conclusions and future research

In this paper we introduced the alternative graph model of ASP, and we presented some results of our experiments carried out at FCO international airport. We show that even a simple greedy heuristic is able to drastically reduce the maximum delay when compared to commonly adopted policies. We observe that an advanced real-time scheduling system may be useful in order to either increase capacity or to maintain the traffic conditions, improving the safety distances between aircrafts.

References

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

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

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

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

[5] M. Pranzo, C. Meloni and D. Pacciarelli, "A new class of greedy heuristics for job shop scheduling problems", *Lecture Notes in Computer Science* 2647, 223–236, (2003).

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

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