OPTIMIZING INBOUND AIRCRAFT STREAMS USING AIRCRAFT DERIVED DATA AND RTA FUNCTIONALITY

D. Nieuwenhuisen, R. de Muynck
National Aerospace Laboratory (NLR)
A. Fokkerweg 2
1059 CM Amsterdam
The Netherlands
+31 88 511 3113
dennis.nieuwenhuisen@nlr.nl; robert.de.muynck@nlr.nl

Keywords: *arrival management, rta, mixed integer linear programming, aircraft landing problem*

Abstract

For the Schiphol Night Time Continuous Descent Approaches project a novel arrival management system has been developed to manage the nightly arrival stream at Schiphol airport. This planning system has been used during flight trials that took place in November 2011. The challenge was to create a data-link with the aircraft to be able to use Flight Management Computer (FMC) derived data in the planning process. This data was then processed into an optimized schedule using an algorithm based on a mixed integer linear program. Finally, the planning was shared with the aircraft which used their RTA function to execute the planning. The planing was also presented to the controller providing a unique view on inbound aircraft far beyond radar coverage. This paper focuses on the arrival management system including its algorithm and interfaces.

1 Introduction

Scheduling aircraft is the process in which the departure or landing order and times of a list of aircraft is established. The goal of such a process is often to optimize the landing sequence given certain criteria. Such criteria can include runway capacity, fuel burn, flight time, controller workload etc.

For arriving aircraft, the scheduling task is to determine a landing order, landing time and (possibly) runway assignment. This task is part of the process called arrival management and the tool used for it is called an Arrival Manager (AMAN). Usually some planning horizon is used for inbound traffic. Such a horizon marks the distinction between planned and not yet planned aircraft. After an aircraft passes the horizon, it is planned by the algorithm. The choice of a suitable planning horizon is often influenced by a number of limiting factors. First, it is often not known when aircraft will arrive. As aircraft usually do not have a direct data-link connection with Air Navigation Service Providers (ANSP), they first need to be picked-up by radar of the affected ANSP. Only then their position (and estimated time of arrival) is known with sufficient precision and an accurate planning becomes possible. Secondly, aircraft are often controlled by multiple centers on their way from top of descent to landing. This is especially true for Europe where sectors and Flight Information Regions (FIR) are often relatively small. Placing a planning horizon outside a FIR leads to the introduction of complex information exchange and coordination tasks between centers. Finally, the larger the planning horizon, the more uncertainty surrounds the actual time of arrival of the aircraft (e.g. because of unexpected wind).

Most larger airports have some sort of AMAN implemented in their system. Depending on the airport and the AMAN, the task to execute the planning is usually left to the controllers. They try to control the aircraft such that the sequence generated by the AMAN is realized. A drawback of this method is that ground-based Trajectory Predictors (TP) are used to estimate the time of arrival of the aircraft. Such estimations are sensitive to errors because of unexpected weather information and lack of knowledge of aircraft intent or aircraft parameters (such as weight). Modern aircraft have FMCs that are able to produce good predictions of Estimated Times of Arrival (ETA). In addition they have Required Time of Arrival (RTA) capabilities that allow them to use a closed-loop process to arrive at a certain point at some predefined time. Unfortunately, most ANSP do not have a capability to interface with these FMC functions.

In this paper an arrival management system is described that has been developed for and used in the recent Schiphol Night Time Continuous Descent Approaches project which was executed under the umbrella of the SJU's AIRE-II project.

1.1 AIRE-II

The SESAR Joint Undertaking (SJU) launched the Atlantic Interoperability initiative to Reduce Emissions II program (AIRE-II). Under this initiative, companies and research organizations were invited to capitalize on present technology and to work collaboratively in order to perform integrated pre-operational validation projects, including flight trials, to demonstrate the benefits of implementing solutions for the reduction of CO_2 emissions[1].

As a response, the Dutch ANSP LVNL, KLM and NLR teamed up in an effort to effectuate a greening of the nightly operation at Amsterdam Schiphol airport. They were supported by Maastricht Upper Area Control (MUAC), NATS and Delta Airlines. Schiphol airport is one of the busiest in Europe. As the airport lies in a densely populated area, strict regulations are in effect concerning noise abatement. This has resulted in the establishment (by Dutch law) of so called *night transitions*. These transitions are mandatory routes that are flow below FL70 in the Schiphol TMA during nightly hours. As are result, controllers cannot intervene in the route of these flights below FL70 (except of course for safety reasons). Aircraft need to be sequenced before they pass FL70. Besides the transitions to abate noise, aircraft also fly Continuous Descent Approaches (CDAs) during night operations at Schiphol as far as practicable. Unfortunately, often the flight paths of aircraft tend to conflict and the controller needs to abort their CDAs in order to ensure safety. This not only happens for aircraft from the same direction, but also aircraft that enter the TMA from opposite directions can have conflicting arrival times. Because of the transitions, this cannot be corrected below FL70 and needs to be done earlier by intervening in the aircraft's CDA. Aircraft for which the controller needs to intervene in their CDA because of conflicting traffic are said to arrive in a *bunch*.

The project goal was to optimize the nightly arrival stream by influencing aircraft en-route and as a result prevent bunching and increase the number of undisturbed CDAs. This should lead to a significant reduction of CO_2 emissions. The project ran from September 2010 till February 2012.

1.2 AIRE-II Schiphol project activities

The project comprised of a number of different activities. After creating the concept of operations, work began to design a planning system capable of optimizing the arrival stream. The planning system consists of a ground based tool, which has a data-link to the Operations Control Center (OCC) of KLM. KLM OCC on its turn maintains a direct data-link with the KLM aircraft using the Aircraft Communications Addressing and Reporting System (ACARS). As a result, the planning system is able to communicate directly with the aircraft through KLM OCC. As more than 75% of the inbound Schiphol

[1] From the specifications attached to the invitation to Tender SJU/LC/0039-CFP.

flights at night are from KLM (and its subsidiaries), this covers a large amount of the inbound traffic. Other airlines (which have far less night-time flights to Schiphol) participate either by coordination using R/T or by entering flight crew responses in a web-based interface connected to the planning system (in case of Delta Airlines).

A communication protocol has been designed that allows for efficient information exchange between the planning system and the aircraft. The planning algorithm generates an optimal planning in terms of fuel and prevention of bunches. The planning is shared with the aircraft that are requested to use their RTA functionality to realize the planning.

Prior to the flight trials, several tests have been conducted to ensure the system worked as planned. These tests have verified the system was technically sound, but have also checked that the ATC procedures were correct and practical.

The actual trial has been conducted during 4 consecutive nights in November 2011. The results of the trial have been published in [10] and in [8]. In this paper the planning system including the planning algorithm are described.

Section 2 will describe the planning system in detail. This section includes a description of the data-link, the planning algorithm and the Human Machine Interface (HMI). In Sections 3 and 4 the results of the trial will be described in terms of the planning system and conclusions will be drawn.

2 The Planning System

The planning system is built around the observation that aircraft (i.e. FMC) derived landing time estimates (ETA) can in theory be much more accurate than their ground-based counterparts (i.e. trajectory predictors). The FMC has knowledge about many aircraft parameters (e.g. weight) and is able to take airline preference into account. An important prerequisite is that the FMC has access to accurate wind information and knows in advance which route will be flown. When developing the planning system, it was assumed that the preferred arrival time for airlines is the ETA as predicted by the FMC. The optimization algorithm (see Section 2.2) therefore attempts to plan aircraft as close to this ETA as possible.

The planning system executes a number of different tasks:

1. communicate with aircraft using data-link;
2. generate an optimized planning;
3. present the planning on an HMI for the controllers.

The planning system works by collecting FMC derived ETA for all inbound aircraft. From this information, an optimal schedule is generated. From this schedule a Planned Time of Arrival (PTA) is extracted for each inbound aircraft. This is the time the aircraft should arrive according to the schedule. The list of PTA times together with other planning information is presented on the controller HMI. This HMI is displayed on a laptop that is placed in the operations room. As aircraft are connected to the planning system long before they appear on the radar, the system provides a unique view on aircraft that are still far from the FIR. To keep the controller in the loop, he/she needs to acknowledge the PTA before it is sent to the aircraft. An overview of the planning system is depicted in Fig. 1.

2.1 Data-link

Within the project, data-link is the prime source of information for the planning system. For the KLM aircraft, a data-link that runs through KLM OCC is available. KLM is able to receive FMC derived data using this ACARS data-link. In addition, the flight-crew is able to respond to requests using accept/reject buttons. This existing ACARS communication infrastructure is used as the basis for the data communication used by the planning system. To ensure an efficient data exchange during the planning process, a set of messages has been defined. The set consists of the following messages:

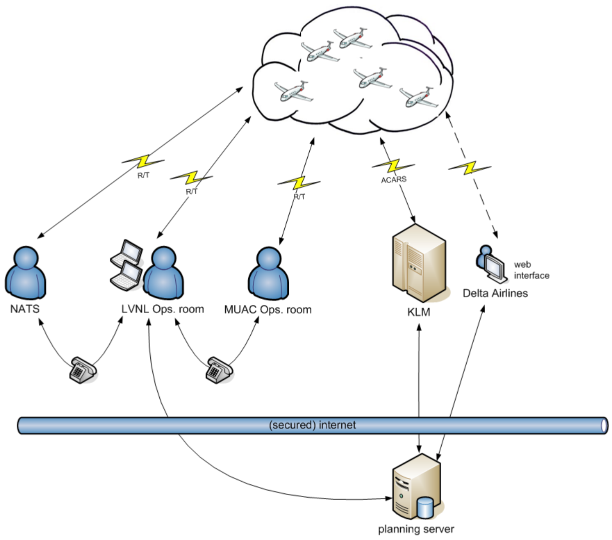

Fig. 1 Schematic view of the planning system

1. Identification message. This is an existing flight update message that is sent by each aircraft to KLM OCC around 90 minutes before landing. It is used to raise awareness by the planning system about an inbound flight.

2. Runway/transition message. This is a new message that is displayed in the cockpit informing the flight crew about the runway/transition they can expect.

3. ETA message. This is an ETA update message that can be sent by the aircraft on request (when it receives a request message from the planning system).

4. PTA message. This message communicates the PTA to the flight crew. This message is displayed in the cockpit.

5. Acknowledge/reject message. This is the response from the flight crew to the proposed PTA.

6. Not active message. This message is sent when a trial has been announced in a NOTAM but has been canceled afterward. This message is displayed in the cockpit.

In Fig. 2 the message exchange process is depicted. As can be seen from the figure, the planning process consists of three phases.

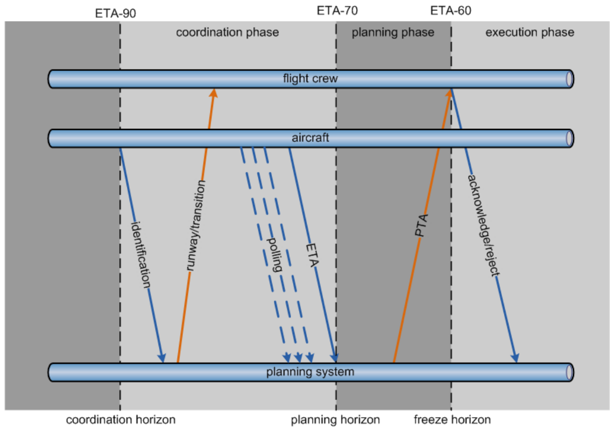

Fig. 2 Message exchange process

1. Coordination phase. This phase is used to ensure that the planning system has a reliable FMS derived ETA (i.e. based on the right runway and transition).

2. Planning phase. When a reliable ETA is available and a flight passed the *planning horizon*, it becomes part of the optimization process.

3. Execution phase. After passing the *freeze horizon*, the flight is frozen in the planning process and executes its PTA.

Aircraft receive information about the active transition and runway before their ETA threshold is requested. In addition all KLM aircraft receive a weather update through data link well before the freeze horizon. This ensures that the ETA times communicated by the aircraft are as accurate as possible. The planning process starts around ETA-90min when the aircraft send an update to OCC which is used as identification message. The planning system responds by sending details about the active runway and transition. This message is displayed in the cockpit and the flight crew is requested to make this runway and transition part of the active route in the FMS. The planning system will then poll the aircraft until the flight crew has made this input. When polling the aircraft the flight crew is not in the loop. As soon as the planning system detects that the proper runway and transition are active, the planning system requests the aircraft to calculate an ETA at the threshold (which is now guaranteed to be based on the right runway and transition). Again the flight crew is not in the loop. This ETA is used to determine the moment the aircraft enters the planning phase which starts at ETA-70min and ends at ETA-60min. During the planning phase, the ETA, together with the ETA of all other aircraft is used by the planning system to generate an optimal schedule. This schedule assigns a PTA to each aircraft. When the aircraft passes the freeze horizon at ETA-60min, its PTA is frozen and communicated to the aircraft. The PTA is displayed in the cockpit. The flight crew is now able to either accept or reject the PTA. When accepted, the pilot agrees to try to realize the PTA. The flight now enters the execution phase. In case of a reject, the flight will be treated as a regular flight.

2.1.1 Interfacing with non-connected aircraft

The procedure from the previous section is followed for about 75% of the aircraft (which belong to the KLM group). For the planning process to be efficient, the remaining flights also need to participate. For flights from Delta Airline, a similar procedure as for the KLM flights is followed, except that there is no direct datalink with the aircraft. A Delta employee forwards the requests from the planning system to the flight crew and types in their responses in the web-interface. For the controllers this is a transparent process in which they see no difference between aircraft with a direct data-link and aircraft in which such a man-in-the-middle is used.

For the remaining aircraft (about 15%), coordination (by telephone) with upstream sectors is used. The upstream sector uses R/T to communicate with the aircraft. The information exchange is the same as with connected aircraft. The flight crew responses are coordinated between the sectors and manually typed in in the planning system by the controller.

2.2 The planning algorithm

At the hearth of the planning system lies the planning algorithm. The task of this algorithm is to generate an optimal schedule based on the downlinked ETA times.

2.2.1 Previous work

The scheduling problem at hand can be formulated as an on-line scheduling problem (see [12] for an overview). In this respect, on-line means that the algorithm does not have access to the whole input set at once, but rather learns the input piece-by-piece, or in this case when the system learns about an arriving aircraft. There are some other properties from the scheduling domain that apply to this problem:

- The problem is not *preemptive*. This means a single job (a flight) cannot be broken into pieces.
- The problem uses *release times*. Release times are the moment a job becomes known to the system. Here, this is the moment an aircraft becomes known to the system when it passes the planning horizon.
- There are no *precedence constraints*, i.e. flights that *need* to land before others. However, the system could be parametrized such that it has a preference to plan one flight before another.
- The *running time* of a job equals the minimum separation between two flights. This may for example be dependent on the wake-categories of the two aircraft.

This type of problem has been an active topic of research for many years. Most algorithms presented attempt to lower the necessary computation time as this may get excessive when more aircraft are involved in the optimization process. Lowering the computation time can be done by limiting the search space such that unnecessary computations are avoided. It can also be done by using heuristics at the expense of completeness. Already in 1976, in [7] an algorithm was presented based on a sliding window. The algorithm tries to lower the necessary computation time by using some assumptions (e.g. limit the amount of aircraft that are involved in the optimization process) that limit the solution space. In 1992 an algorithm called the Implicit Enumeration scheduling algorithm was presented in [3] which was part of the then used NASA CTAS system. The algorithm assigns a landing sequence and landing times to arriving aircraft. It implicitly enumerates all solutions but defines criteria to reduce the necessary computation time by static limiting (when a constraint is violated), dynamic limiting (when the current solution has no potential to be better than the best-so-far solution) and depth limiting (which adds a sliding window approach).

In [2] the off-line version of the problem is solved by using a linear program. The algorithm is capable of handling multiple runways. An optimal solution is given using a tree search. In addition a heuristic is described based on an upper bound for the solution to curtail the tree search. [11] builds on these results and gives two heuristic solutions. Genetic algorithms have also been used to solve the problem [4, 5, 6, 9]. The advantage of genetic algorithms is that a valid solution is always available and the algorithm can be easily run on multiple processors.

The dynamic (or on-line) case of the problem is treated in [1]. A generic decision problem is first defined and then applied to the aircraft landing problem. Next, one optimal (based on linear programming) and two heuristic solutions are presented, based on previous work.

2.2.2 *Formal formulation of the scheduling problem*

We will start by modeling the problem as an off-line scheduling problem. This means that we assume that we have information about all flights participating in the optimization process beforehand. In Section 2.2.5 the translation to the on-line version is described. Throughout the remainder of this paper, the following notation is used to describe the problem at hand.

- There are n different flights i that need to be planned: $i \in \{1 \ldots n\}$.
- Each flight i communicates a target threshold time ETA_i^{opt}, which is the optimal time the flight would like to pass the threshold (the ETA).
- The optimization process generates for each flight i a planned time of arrival (PTA) at the threshold: PTA_i. This is the time the flight crew is requested to use as an RTA.
- Each flight i has an earliest time, ETA_i^{min} which is the earliest time the aircraft can pass the threshold. As a result $PTA_i \geq ETA_i^{min}$. The earliest time is dependent on the time the aircraft is able to gain.
- Analogously, with each flight is associated a latest time ETA_i^{max}. The latest time is dependent on the time the aircraft is realistically able to lose. In theory it is possible that no solution exists that fulfills all constraints. Therefore we allow aircraft to be planned after ETA_i^{max}, but at a high penalty. This ensures that a solution always exists.
- Each flight occupies a certain minimum width in the schedule. This width is not fixed but is determined by the wake-category of the aircraft and possibly by other constraints for example determined by the flight dynamics of specific types of aircraft. A matrix M is used to store minimum separation times where element M_{ij} determines the minimum separation time between flights i and j when i lands before j.
- The decision variable δ_{ij} determines whether aircraft i lands before aircraft j.

$$\delta_{ij} = \begin{cases} 1, \text{if aircraft } i \text{ lands before aircraft } j, \\ \qquad\qquad (i,j) \in \{1 \ldots n\}^2 \\ 0, \text{otherwise.} \end{cases}$$

2.2.3 *Formulation of constraints*

We will now formulate the basic problem as a linear program. This has been done before in several papers. Here we roughly follow the formulation of [11] with some extensions. First, PTA_i cannot be earlier than ETA_i^{min}:

$$PTA_i \geq ETA_i^{min},\ \forall i \in \{1\ldots n\} \tag{1}$$

Either flight i or j will land first, therefore:

$$\delta_{ij} + \delta_{ji} = 1,\ \forall (i,j) \in \{1\ldots n\}, i \neq j \tag{2}$$

The minimum separation M_{ij} between two flights needs to be respected when aircraft i lands before aircraft j. For this, matrix M is used:

$$PTA_j \geq PTA_i + M_{ij},\ \forall (i,j) \in \{1\ldots n\}, i \neq j$$

The above equation fails when aircraft j lands before i; to ensure the equation still holds we use a large number K:

$$PTA_j \geq PTA_i + M_{ij} - \delta_{ji}K \quad \forall (i,j) \in \{1\ldots n\}, i \neq j, K \gg 0 \tag{3}$$

2.2.4 Objective function

To generate an optimal planning, we will use the constraints of the previous section to optimize an objective function. Our objective function will try to minimize the sum of the deviations of the flights from their target time (ETA^{opt}) by using a linear objective function. To create such function, we introduce three variables. The first variable is t_i^a, which is the time aircraft i arrives before optimal time ETA_i^{opt}. The second variable is t_i^b, which is the time i arrives after optimal time ETA_i^{opt}. We define the following:

$$t_i^a = \max(0, ETA_i^{opt} - PTA_i)$$

$$t_i^b = \max(0, PTA_i - ETA_i^{opt})$$

Either t_i^a or t_i^b will be 0. In case the aircraft lands at its preferred time, both are 0. Note that we assume here that, for example, arriving 30 seconds earlier than ETA^{opt} has the same cost as arriving 30 seconds late. This may not always be the case. Introducing a weighting factor for t_i^a and for t_i^b gives the possibility to make a distinction between arriving earlier than ETA^{opt} or arriving later. Here, we do not use such factor.

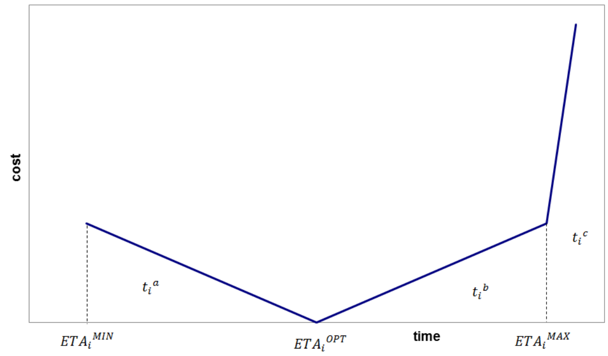

Fig. 3 Objective function

In some extreme circumstances (i.e. heavy bunching) it may not be possible to plan a flight i before ETA_i^{max}. We prefer to have a flight planned after ETA_i^{max} than to have an infeasible problem that does not lead to a solution. Simply removing the upper bound on t_i^b may however lead to a solution in which one aircraft is moved beyond ETA^{max} while a solution exists with the aircraft *before* ETA^{max}. To prevent this, we introduce t_i^c that allows an aircraft to be moved beyond ETA^{max} but at the cost of a large penalty (by multiplying by 100 in the objective function). t_i^c is the time aircraft i is planned beyond its latest time ETA_i^{max}.

$$t_i^c = \max(0, PTA_i - ETA_i^{max})$$

The individual cost function for flight i is defined by:

$$t_i^a + t_i^b + 100 t_i^c$$

The shape of the resulting cost function for a single aircraft is shown in Fig. 3.

We need some extra constraints to link the variables to the ETA min/max times for each aircraft:

Constrain t_i^a, t_i^b and t_i^c:

$$t_i^a \geq 0,\ \forall i \in \{1\ldots n\} \tag{4}$$

$$t_i^a \leq ETA_i^{opt} - ETA_i^{min},\ \forall i \in \{1\ldots n\} \tag{5}$$

$$t_i^a \geq ETA_i^{opt} - PTA_i,\ \forall i \in \{1\ldots n\} \tag{6}$$

$$t_i^b \geq 0,\ \forall i \in \{1 \ldots n\} \tag{7}$$

$$t_i^b \leq ETA_i^{max} - ETA_i^{opt},\ \forall i \in \{1 \ldots n\} \tag{8}$$

$$t_i^c \geq 0,\ \forall i \in \{1 \ldots n\} \tag{9}$$

$$t_i^b + t_i^c \geq PTA_i - ETA_i^{opt},\ \forall i \in \{1 \ldots n\} \tag{10}$$

As the objective is to minimize the deviation from the target times, the objective function becomes:

$$minimize \sum_{i=1}^{n} (t_i^a + t_i^b + 100 t_i^c) \tag{11}$$

Now the goal of the optimization is to minimize equation 11 subject to constraints (1)...(10).

2.2.5 Translation to an on-line problem

In a realistic situation (such as in the case of the AIRE-II project), the problem is not off line but rather on line. This means that we do not know the whole input set (the arriving aircraft) in advance, but rather "learn" them one by one when they are identified by the system. As this is a continuous process, we cannot wait until we have collected information about all aircraft. We need to communicate a PTA when an aircraft passes the planning horizon. As a result, three groups of aircraft can be distinguished (see Fig. 4):

1. Flights that cannot yet be planned because there is insufficient information available. These are flights that are known to the system but did not yet communicate a reliable ETA. Flights in this group are said to be part of the *coordination group*.

2. Flights for which there is sufficient information available to include them in the planning process. For these flights a reliable ETA is available. This is the *planning group*.

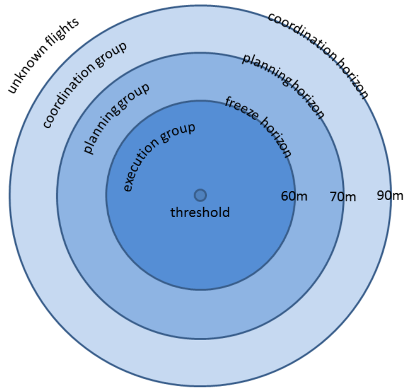

Fig. 4 Groups of flights during the planning process

3. Flights that have already been planned, we call these aircraft the *execution group*.

Flights from the coordination group cannot be part of a planning process because the information about them is incomplete; we do not know their ETA with sufficient precision (e.g. because they do not yet have the proper transition programmed into the FMS). Aircraft from the execution group are already planned and their PTA is frozen. Therefore only aircraft from the planning group can be part of an optimization process.

After a PTA has become definitive for a flight, the flight moves to the execution group. In the Schiphol AIRE-II trial, there was no negotiation of the PTA; only one PTA was generated for each flight. Flights from the execution group will form a constraint to flights from the planning group as they occupy a slot. Additional constraints are used to ensure that when a flight is planned, they will respect the slots that flights from the execution group occupy. Say we have n flights $i \in \{1 \ldots n\}$ in the planning group and m flights $j \in \{n+1 \ldots n+m\}$ in the execution group. First we introduce additional δ variables to ensure that flights from the planning group are planned before or after flights from the execution group. For this, we re-state the definition of δ_{ij}:

$$\delta_{ij} = \begin{cases} 1, \text{if aircraft } i \text{ lands before aircraft } j \\ \qquad (i,j) \in \{1 \ldots n\}^2 \\ 1, \text{if aircraft } i \text{ lands before aircraft } j \\ \qquad i \in \{1 \ldots n\}, j \in \{n+1 \ldots n+m\} \\ 0, \text{otherwise.} \end{cases}$$

Again, only one of the flights can land first:

$$\delta_{ij} + \delta_{ji} = 1,\ \forall i \in \{1 \ldots n\}, \forall j \in \{n+1 \ldots n+m\} \tag{12}$$

Table M is extended to also contain the fixed flights. To respect the minimum separation we add the following constraints:

$$PTA_j \geq PTA_i + M_{ij} - \delta_{ji}K \quad \forall i \in \{1 \ldots n\}, \forall j \in \{n+1 \ldots n+m\}, K \gg 0 \tag{13}$$

$$PTA_j \geq PTA_i + M_{ij} - \delta_{ji}K \quad \forall i \in \{n+1 \ldots n+m\}, \forall j \in \{1 \ldots n\}, K \gg 0 \tag{14}$$

Now the goal of the optimization process can be re-stated as to minimize equation 11 subject to constraints (1)...(14).

We now describe the planning procedure for the on-line problem. As soon as a flight from the planning group passes the freeze horizon, all other flights in the planning group are collected. These aircraft form the input of the linear program. The flights from the execution group form a list of constraints for this program.

Only for the flight that passed the freeze horizon, a PTA is communicated to the aircraft. The flight now moves from the planning group to the execution group (as its PTA is frozen now). The other flights participating in the optimization process remain untouched until a next flight passes the horizon. This approach resembles a sliding window approach. When flights have landed, they are removed from the execution group so that they do not add unnecessary constraints to the linear program.

The linear program is solved using the solver `lp_solve`[13]. This is a mixed integer linear programming (MILP) solver based on the revised simplex method and the Branch-and-bound method for the integers. We need to be able to use integer variables as the δ variables can only have integer values 0 or 1.

2.3 The planner HMI

The planning system interfaces with the controllers using an HMI (Fig. 5). In addition, the man-in-the-middle at Delta airlines also uses this interface to communicate with the planning system. The interface is web-based to ensure compatibility with a variety of systems and uses AJAX to cater for the dynamic nature of the interface. Safety is ensured by using a login system and the use of secure http.

Fig. 5 Screenshot of the planner HMI

The interface is divided in 3 areas. The topmost area is used to manually add flights, read information about the runway in use, show the current TMA wind and to activate/deactivate the planning system. When deactivated while a trial was previously announced, each inbound aircraft automatically receives the trial canceled message. The center area shows the flights in the execution group (that passed the freeze horizon), the bottom area shows the flights in the coordination and planning groups.

The center and bottom areas present a number of different columns with information about the aircraft. The first column shows whether a data-link is active with the aircraft. The next columns show callsign, aircraft type, runway, transition and Scheduled Time of Arrival (STA), this is the arrival time according to the flight plan. The next columns show the planning information. The ETA_{THR} shows the last down-linked FMS derived ETA for the threshold. The PTA_{THR} shows the planned time of arrival for the threshold. In between, the Δ column shows the difference between the two. The PTA_{IAF} shows a ground based estimate for TMA entry and is not used in the planning process. Finally, the status column shows the status of the flight. The following statuses can be distinguished:

- Expected. The flight has been identified as inbound but did not yet pass the planning horizon.
- Planning. The flight has passed the planning horizon, but not the freeze horizon.
- Waiting ATCo acknowledge. The flight has been assigned a PTA and awaits ATCo acknowledge.
- Waiting crew response - the PTA has been sent to the aircraft and awaits flight crew approval.
- Accepted. The PTA has been accepted by the flight crew.
- Rejected. The PTA has been rejected by the flight crew.

For the bottom part, some information is not yet known (e.g. the PTA) and is omitted.

During the AIRE-II flight trials, the HMI was displayed on a laptop in the operations room. As the controllers liked to know which flights to expect, all flights were manually added to the system before the trial started to show their scheduled time of arrival. When the actual flight sent its identification message, these were correlated. For the non-connected flights, coordination with the upstream sectors was used, using regular telephone connections. The upstream sectors used R/T to coordinate with the flight deck. The information that was coordinated consisted of:

1. Runway/transition from the upstream sector to the flight deck.
2. FMS derived ETA (based on right runway/transition) from the flight deck to the upstream sector.
3. PTA from the upstream sector to the flight deck.
4. Accept/reject from the flight deck to the upstream sector.

3 Algorithmic results

The planning system has been used during the Schiphol Night time CDA AIRE-II trials. In these trials, the planning system was used during 4 consecutive nights in November 2011. The planning system ran on a 3.0GHz Xeon processor. Towards this trial, tests with the data-link, both with and without a pilot in the loop have been performed. This has resulted in some improvements in the data-exchange. Most issues could be attributed to small differences in communication protocol between different aircraft types. Except for one issue on the first day of the trial, no technical issues have come to light during the trial. The planning system therefore provided a solid platform for the flight trials.

Some algorithmic results observed during the trial are depicted in Table 3. The table shows some planning examples that occurred during the November 2011 trials. As can be seen, the amount of aircraft involved in the planning process is relatively small.

The results from the table clearly show that the algorithm performed nearly real-time and no delay in the performance of the planning algorithm has been experienced in practical scenarios.

To further test algorithmic performance, some synthetic tests have been performed on the same type of computer. Testing showed that the calculation time increases exponentially with

#Flights in planning group	4	2	2	3	5
#Flights in execution group	2	12	17	11	9
#Equations for linear program	60	82	112	120	190
Calculation time (s)	0.003	0.006	0.012	0.021	0.036

Table 1 Some calculation results observed during the flight trials.

the number of constraints in the linear program. Adding flights to the planning group increases calculation time more than adding flights to the execution group. This is as expected as there are more constraints involved for flights in the planning group. If many flights have ETAs very close to each other, the exponent of the calculation time increases. In worst-case (with ETA very close to each other), experiments show that around 6-7 flights in the planning group is the maximum for sufficient performance with the current implementation (see Table 3). Because the planning group consists of flights with ETA within 10 minutes, even in this unlikely scenario this seems sufficient in practice. Code optimization and parallelization could increase algorithm performance further if necessary.

4 Conclusions and lessons learned

The planning system presented in this paper is able to automatically de-bunch inbound aircraft by using FMC derived data and the RTA functionality of aircraft. To do so, it uses an ACARS based data-link with the aircraft and is able to directly communicate with the flight crew. The optimization algorithm, based on a mixed-integer linear program is able to provide near real-time performance on realistic input sets. During the trials, the amount of bunching was significantly reduced, enabling more top-of-descent CDAs. More elaborate results of the Schiphol AIRE-II trials can be found in [8] and [10].

During the trial some valuable lessons were learned:

- A planning horizon of more than 60 minutes resulted in inaccuracies mainly caused by directs offered by upstream sectors. When accepted, a direct obviously makes the previously sent ETA incorrect. The possibility for re-negotiating the PTA with the planning system may offer a solution to this problem. Algorithmically this comes down to moving a flight from the execution group back to the planning group.
- Some more controller expertise should be incorporated in the planning process. Either by providing the opportunity to manually adapt the sequence or by incorporating this expertise in the planning process. An example of such an improvement is the grouping of flights arriving from the same direction. Algorithmically this requires the addition of some constraints.
- RTA performance is currently strongly dependent on aircraft type. Time-based operations benefit from state-of-the-art FMS RTA functions.
- Data-link should be the preferred way of communicating with aircraft, the manual coordination with the non-connected aircraft was error prone and is too laborious to be used in day to day practice.
- As aircraft derived ETA tend to fluctuate over time, it is beneficial to update the ETA information as frequently as possible.
- For controllers to keep track of progress, during the execution phase regular ETA updates should be requested and presented on the HMI.
- Accurate on-board wind information is vital for the quality of the down-linked ETAs and therefore also for the quality of the planning process.
- One of the success criteria for the planning system was the use of existing and proven

#Flights in planning group	1	2	3	4	5	6
#Flights in execution group	13	12	11	10	9	8
#Equations for linear program	42	82	120	156	190	222
Calculation time (s)	0.002	0.011	0.052	0.297	1.91	14.10

Table 2 Some worst-case calculations from the synthetic tests with ETA very close to each other.

technology as much as possible, both for the planning system and the data-link.

References

[1] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha and D. Abramson, *Displacement problem and dynamically scheduling aircraft landings*. Journal of the Operational Research Society 55, pp. 54-64, 2004

[2] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha and A. Abramson, *Scheduling aircraft landings - The static case*. Transportation Science 34, pp. 180-197, 2000

[3] C. R. Brinton, *An implicit enumeration algorithm for arrival aircraft scheduling*. Proceedings of the 11th Digital Avionics Systems Conference, pp. 268-274, 1992

[4] V. Ciesielski and P. Scerri, *An anytime algorithm for scheduling of aircraft landing times using genetic algorithms*. Australian Journal of Intelligent Information Processing Systems 4, pp. 206-213, 1997

[5] V. Ciesielski and P. Scerri, *Real time genetic scheduling of aircraft landing times*. Proceedings of the 1998 IEEE International Conference of Evolutionary Computation, New York, pp. 360-364, 1998

[6] V.H.L. Cheng, L.S. Crawford and P.K. Menon, *Air traffic control using genetic search techniques*. Proceedings of the 1999 IEEE International Conference on Control Applications, pp. 249-245, 1999

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

[8] N. De Gelder, D. Nieuwenhuisen and E. Westerveld, *AIRE-II Final result report*. LVNL S&P/2012/047, 2012

[9] J.V. Hansen, *Genetic search methods in air traffic control*. Computers & Operations Research 31 (3), pp. 445-459, 2004

[10] D. Nieuwenhuisen, *Optimizing nightly Schiphol traffic trough time based operations*. Air Transport and Operations Symposium (ATOS), 2012

[11] H. Pinol and J.E. Beasley, *Scatter search and bionomic algorithms for the aircraft landing problem*. European journal of operational research 171, pp. 439-462, 2006

[12] J. Sgall, *On-line Algorithms (Lecture Notes in Computer Science)*. A. Fiat and G. Woeginger, Springer Berlin, pp. 196-231, 1998

[13] http://lpsolve.sourceforge.net/5.5/

5 Copyright Statement