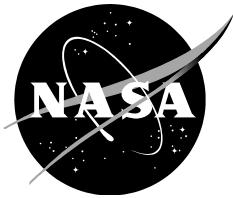


NASA/TP—2014-218302



# **Design Principles and Algorithms for Air Traffic Arrival Scheduling**

*Heinz Erzberger  
Ames Research Center  
Moffett Field, California*

*Eri Itoh  
Electronic Navigation Research Institute  
Tokyo, Japan*

---

**May 2014**

## NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

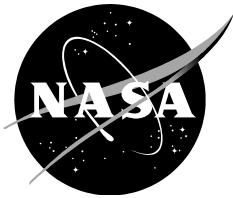
- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Phone the NASA STI Information Desk at 757-864-9658
- Write to:  
NASA STI Information Desk  
Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199

NASA/TP—2014-218302



# Design Principles and Algorithms for Air Traffic Arrival Scheduling

*Heinz Erzberger  
Ames Research Center  
Moffett Field, California*

*Eri Itoh  
Electronic Navigation Research Institute  
Tokyo, Japan*

National Aeronautics and  
Space Administration

*Ames Research Center  
Moffett Field, California*

---

**May 2014**

Available from:

NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320  
443-757-5802

National Technical Information Service  
5301 Shawnee Road  
Alexandria, VA 22312  
703-605-6000

## Table of Contents

<b>Abstract.....</b>	<b>1</b>
<b>Chapter 1.....</b>	<b>2</b>
1. Introduction.....	2
<b>Chapter 2.....</b>	<b>4</b>
2. Arrival Time Scheduling in Real Time.....	4
2.1 Airspace and Route Structure .....	4
2.2 In-Trail Distance Separations .....	5
2.3 First-Come-First-Served (FCFS) Arrival Sequence Orders .....	7
2.4 Position Shifting.....	8
2.5 "Freeze Horizon".....	9
2.6 Role of Freeze Horizon in Real-Time Scheduler Design.....	9
<b>Chapter 3.....</b>	<b>12</b>
3. Basic Algorithms for Scheduling Time of Arrivals .....	13
3.1 Description of the Basic Algorithms.....	13
3.2 Multiple Meter Gates to One Runway .....	13
3.3 Multiple Terminal Gates to Multiple Runways.....	22
<b>Chapter 4.....</b>	<b>27</b>
4. Advanced Algorithms.....	27
4.1 Real-Time Scheduling Algorithm with Limited Sequence and Runway Assignment Optimization.....	27
Freeze Horizon and Freeze-Horizon Time Interval .....	27
Optimization Horizon and Optimization Interval.....	27
Influence Horizon and Influence Interval .....	27
4.2 Estimating the Number of Trial Scheduling Combinations in the Comparison Set.	30
4.3 Adding a New Aircraft to the Schedule .....	30
<b>Chapter 5.....</b>	<b>33</b>
5. Strategy for Delay Absorption in the Presence of Time-Control Errors .....	33
5.1 Delay Allocation in Center and TRACON Airspace.....	33
5.2 Stochastic Simulation of Meter Gate Crossing Errors.....	35
5.3 Analysis of Results .....	36
5.4 Unification Principle .....	38
<b>Chapter 6.....</b>	<b>41</b>
6. Conclusion.....	41
<b>References.....</b>	<b>41</b>

## List of Figures

Figure 1. Airspace structure and arrival routes .....	5
Figure 2. Explanation of “freeze horizon” .....	10
Figure 3a. Adding in-trail constraints at the meter gates (step 1) .....	14
Figure 3b. Determining landing order (step 2) .....	16
Figure 3c. Determining the runway (STA)s (step 3) .....	17
Figure 3d. Delay distribution function (step 4).....	18
Figure 3e. Schedule time of arrivals at meter gates using delay distribution function (step 5)....	19
Figure 4a. Determining landing order of the first and second aircraft in the PAFCFS .....	21
Figure 4b. Determining landing order of the potential third aircraft in the PAFCFS.....	22
Figure 4c. Determining landing order by pushback of (STA <sub>FF</sub> )s using DDF in the PAFCFS.....	22
Figure 5. Choosing a non-FCFS order and runway assignment that minimizes slot loss.....	24
Figure 6a. Illustration of real-time scheduling for two runways.....	29
Figure 6b. Alternative runway assignment for A <sub>i</sub>	29
Figure 6c. Illustration of real-time scheduling algorithm: Adding a new aircraft A <sub>new</sub> to the scheduled list of frozen aircraft.....	31
Figure 7. Stochastic Monte Carlo simulation of scheduler .....	36
Figure 8. Average delay increment for 36 aircraft/hour for a single runway vs. d <sub>Tmax</sub> .....	37
Figure 9. Incremental fuel vs. d <sub>Tmax</sub> .....	38
Figure 10. DOC increment curves on the time-fuel coordinates .....	39

## List of Tables

Table 1. Minimum distance separation matrix for pairs of aircraft based on current FAA wake vortex separation standards at a runway threshold .....	6
Table 2. Examples of minimum time separation at a runway threshold when the true air speed (TAS) and ground speed is 130 knots (no wind) at the threshold.....	6
Table 3. FCFS ordered list of (ETA)'s .....	8
Table 4. Illustration of position-shifted sequence order relative to FCFS order.....	8

## **Abstract**

This report presents design principles and algorithms for building a real-time scheduler of arrival aircraft based on a first-come-first-served (FCFS) scheduling protocol. The algorithms provide the conceptual and computational foundation for the Traffic Management Advisor (TMA) of the Center/terminal radar approach control facilities (TRACON) automation system, which comprises a set of decision support tools for managing arrival traffic at major airports in the United States. The primary objective of the scheduler is to assign arrival aircraft to a favorable landing runway and schedule them to land at times that minimize delays. A further objective of the scheduler is to allocate delays between high-altitude airspace far away from the airport and low-altitude airspace near the airport. A method of delay allocation is described that minimizes the average operating cost in the presence of errors in controlling aircraft to a specified landing time. This report is a revision of an earlier paper first presented as part of an Advisory Group for Aerospace Research and Development (AGARD) lecture series in September 1995. The authors, during vigorous discussions over the details of this paper, felt it was important to the air-traffic-management (ATM) community to revise and extend the original 1995 paper, providing more detail and clarity and thereby allowing future researchers to understand this foundational work as the basis for the TMA's scheduling algorithms.

# **Chapter 1**

## **1. Introduction**

The urgent need for increasing the efficiency of the air-traffic-management (ATM) process has led to intense efforts in designing automation systems for air traffic control. The efforts have concentrated on two major technical challenges: designing an aircraft trajectory synthesizer/estimator and a real-time scheduler. The trajectory synthesizer/estimator computes flight times of aircraft to selected points, for example, meter gates and runway thresholds. The real-time scheduler processes these flight-time estimates along with required air traffic control's constraints (e.g., separation) to generate scheduled arrival times of aircraft to these points in order to make effective use of airport and runway capacity. The design of the trajectory synthesizer/estimator, though technically complex, can be accomplished by the application of well-established methods for navigation, guidance, and control of aircraft (refs. 1 and 2). In contrast, the design of the real-time scheduler for arrival traffic had minimal technical precedence to build upon at the time the original version of this paper was first published in 1995. It required blending operational knowledge of air traffic control with algorithms specifically created for scheduling arrival traffic and was, therefore, an appropriate subject for the Advisory Group for Aerospace Research and Development (AGARD) Lecture Series 200, entitled "Knowledge-Based Functions in Aerospace Systems" (ref. 3).

The scheduler described herein is incorporated in the Center/TRACON Automation System (CTAS), which was developed jointly by NASA Ames Research Center and the Federal Aviation Administration (FAA). The automation tools in CTAS consist of the Traffic Management Advisor (TMA), the Descent Advisor (DA), and the Final Approach Spacing Tool (FAST) (refs. 4 and 5). In combination these tools generate advisories that assist controllers in handling aircraft from about 40 minutes of flying time to an airport, until they reach the final approach fix. Several of its tools were evaluated in real-time simulation tests as well as field evaluation at the Denver and Dallas/Fort Worth airports (refs. 6 through 9). After completion of these proof-of-concept tests, the FAA chose the TMA element of CTAS for further development and embarked on a decade-long project to deploy TMA at all of its Centers and large TRACONs. This report focuses on the conceptual and algorithmic design of the real-time scheduler in the TMA.

This report is a revised and updated version of a paper originally published in 1995 as part of the AGARD Lecture Series 200 on "Knowledge-Based Functions in Aerospace Systems" (ref. 3). The revisions made herein include changes to figures and text designed to improve explanation of the technical concepts presented. The conceptual and algorithmic methods for real-time air traffic scheduling described in the original paper became the foundation for a software implementation referred to as the Dynamic Planner, which is a key component of TMA (ref. 10). The decision to revise the original paper was strongly motivated by discussions between the original author and the coauthor, who provided insightful critiques and numerous suggestions for improvements. The authors, during vigorous discussions over the details of this paper, felt it was important to the ATM community to revise and extend the original 1995 paper to provide more detail and clarity to help future researchers understand this foundational work as the basis of the TMA's scheduling algorithms.

This paper is organized as follows: Chapter 2 explains scheduling constraints in the real world. The reasons why the first-come-first-served (FCFS) principle is applied for handling the complexity in the arrival scheduling are explained. A "freeze-horizon" method is introduced to achieve efficient orders and means of control of arrival sequencing in the real-time process.

Chapter 3 describes the scheduling algorithms that generate schedules to the runway threshold while obeying FCFS sequence constraints at both meter gates and runways. Chapter 4 covers advanced algorithms including optimization of the arrival sequences in the real-time process. Additionally, this chapter describes a modification to the basic algorithms for handling one of the most important as well as frequently occurring events, namely the efficient insertion of a new arrival into the previously scheduled set of aircraft. Chapter 5 discusses strategies for delay absorption while accounting for two key factors: aircraft fuel consumption and accuracy of time control. Simulation results show the tradeoffs that exist between Center and TRACON delay distribution, aircraft direct operating cost, defined as the sum of the cost of time and the cost of fuel, and control accuracy. Concluding remarks are in Chapter 6.

## Chapter 2

### Nomenclature

$d_{it}$	= specified in-trail distance separation at a meter gate (not less than 5 nautical miles (n. mi.)
$D_{it}$	= specified in-trail distance separation at a runway threshold
$ETA$	= estimated time of arrival at a runway threshold
$(ETA)_s$	= more than one $ETA$
$ETA_{FF}$	= estimated time of arrival at a meter gate
$(ETA_{FF})_s$	= more than one $ETA_{FF}$
$v_g$	= estimated ground speed of trailing aircraft at a meter gate
$V_g$	= estimated ground speed of trailing aircraft at a runway threshold
$t$	= time
$t_{it}$	= time separation of trailing aircraft from leading aircraft at a meter gate
$T_{it}$	= time separation of trailing aircraft from leading aircraft at a runway threshold
$\Delta_F$	= time interval to fly from freeze horizon to the meter gate
$t_F$	= time when an aircraft crosses the freeze horizon

## 2. Arrival Time Scheduling in Real Time

### 2.1 Airspace and Route Structure

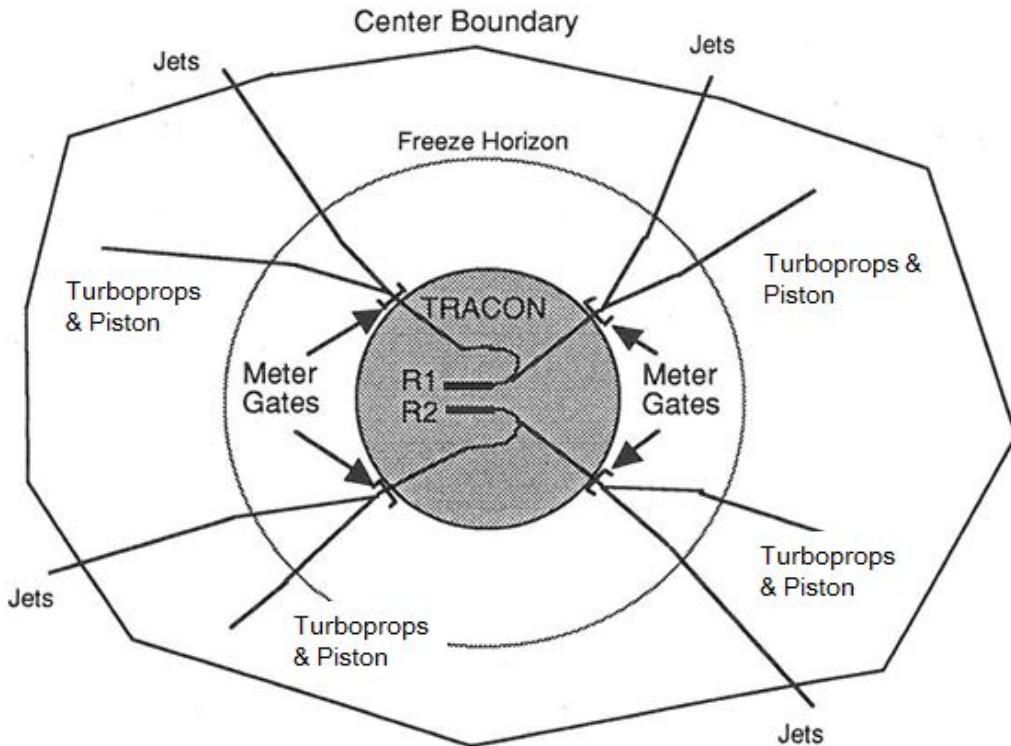
In order to serve the changing needs of airlines and air traffic control, the airspace and route structure surrounding a large airport have evolved into increasingly complex forms. Here we describe only those features that relate directly to the design of the real-time scheduler. The features are described for simple airspace, but the concepts and algorithms are applicable for the general case.

For the purpose of the scheduler design, the arrival airspace is divided into Center and TRACON regions. Whereas the Center region has an irregular outer boundary, the TRACON region is a roughly circular area about 35 n. mi. in radius around an airport and is completely surrounded by the Center airspace. Certain waypoints located on the boundary between the two regions are referred to as meter gates. During moderate and heavy traffic conditions when delays are expected, traffic is funneled through these gates as a means of controlling or metering the flow rate into the terminal area. In most terminal areas, arrival routes are merged at gates corresponding to the primary arrival directions. In this paper we use four gates, but this number is easily extensible.

Traffic flowing to each gate is often segregated into three independent streams by separating each stream vertically by at least 1000 feet in altitude at the gate-crossing point. This segregation is done to permit the three primary aircraft types, jets, turboprops, and piston, which have significantly different airspeed ranges, to cross the gates independently, thereby avoiding conflicts due to overtakes near the gates.

From each gate, routes that lead to all possible landing runways for each independent stream are defined in the TRACON airspace. For the design of the scheduler, the exact horizontal paths of the routes provide a structure from which the trajectory synthesizer can produce nominal flying times from each gate to all landing runways; these flying times must be provided as input.

Figure 1 illustrates the concepts of airspace structure, arrival routes, meter gates, and stream types as described previously. “freeze horizon”, which lies within the Center airspace in Figure 1, is explained in section 2.5.



**Figure 1.-Airspace structure and arrival routes.**

## 2.2 In-Trail Distance Separations

Scheduling constraints can be broadly classified into two types: in-trail distance-separation constraints and sequence-order constraints. Here the former is discussed.

In the design of the scheduling algorithm, the in-trail constraints play an especially important role because they determine the capacity of an airport and hence the maximum landing rate. The scheduling algorithm must be designed to meet, at a minimum, in-trail constraints both at the meter gates and on the final approach paths.

At the meter gates in-trail separations may be specified by separate parameters for each independent traffic stream in order to provide flexible control of the total flow rate into the TRACON. However, in the absence of flow-rate control at the meter gates, safety considerations require the minimum in-trail distance separations to be not less than 5 n. mi. Since scheduling is done in the time domain, all distances must be converted into equivalent time separations. In general, the conversion first determines the ground speed from estimates of airspeed and wind speed and then applies the following relation:

$$\left. \begin{aligned} t_{it} &= d_{it}/v_g \\ T_{it} &= D_{it}/V_g \end{aligned} \right\} \quad (1)$$

On the final approach path the minimum in-trail distance separations are a function of both aircraft weight class and landing order as determined by the FAA's wake vortex safety rules. Table 1 gives the values in matrix format. The table also gives examples of aircraft models falling in the different weight categories. As before, the distance separations in table 1 must be converted to equivalent time separations for use by the scheduler. The conversion process is complex and is given here only in outline. Generally, it involves modeling the airspeed profile of each type of aircraft and the wind speed on final approach and then integrating the equations of motion along the final approach path. The result of this process is the time separation matrix given in table 2 for the case of zero wind.

**TABLE 1. MINIMUM DISTANCE SEPARATION MATRIX FOR PAIRS OF AIRCRAFT BASED ON CURRENT FAA WAKE VORTEX SEPARATION STANDARDS AT A RUNWAY THRESHOLD**

		Trailing aircraft categories (n. mi.)				
		Super	Heavy	B757	Large	Small
Leader aircraft categories (n. mi.)	Super	2.5	6	7	7	8
	Heavy	2.5	4	5	5	6
	B757	2.5	4	4	4	5
	Large	2.5	2.5	2.5	2.5	4
	Small	2.5	2.5	2.5	2.5	2.5

**TABLE 2. EXAMPLES OF MINIMUM TIME SEPARATION AT A RUNWAY THRESHOLD WHEN THE TRUE AIR SPEED (TAS) AND GROUND SPEED IS 130 KNOTS (NO WIND) AT THE THRESHOLD**

		Trailing aircraft categories (seconds (sec))				
		Super	Heavy	B757	Large	Small
Leader aircraft categories (sec)	Super	69	166	194	194	222
	Heavy	69	111	138	138	166
	B757	69	111	111	111	138
	Large	69	69	69	69	111
	Small	69	69	69	69	69

For tables 1 and 2:

- Super = a separate designation that currently refers only to the Airbus A380
- Heavy = aircraft capable of takeoff weights of 300,000 pounds (140,000 kg) or more whether or not they are operating at this weight during a particular phase of flight
- Large = aircraft of more than 41,000 pounds (19,000 kg) maximum certificated takeoff weight, up to but not including 300,000 pounds (140,000 kg)
- Small = aircraft of 41,000 pounds or less maximum certificated takeoff weight

The numerical values given in these tables should be interpreted as representative examples for the purposes of this paper. They may be revised when new operational experience determines that vortex separation rules need to be modified to improve safety.

### **2.3 First-Come-First-Served (FCFS) Arrival Sequence Orders**

A sequence-order constraint specifies the order with respect to time that a group of aircraft must be scheduled to cross a time control point. The runway threshold and the meter gates are the points where sequence constraints are frequently enforced. They provide an essential mechanism for achieving scheduling efficiency, scheduling fairness, and controller preference. The sequence order that often meets the requirements of all three of these objectives simultaneously is the first-come-first-served (FCFS) order. It therefore plays the role of the standard or canonical sequence order against which the performance and acceptability of all other orders is referenced.

The basic objective of the scheduler in air-traffic-control automation is to match traffic demand and airport capacity while minimizing delays. As we shall see, this concise- and straightforward-sounding objective gives rise to a surprisingly complex algorithmic design problem when all necessary operational constraints are considered. This chapter presents an outline of the solution to this problem.

The dynamic nature of air traffic flow requires that the scheduler be designed to operate as a real-time process, which is defined in the following way. The scheduler must generate an updated schedule for the set of aircraft to be scheduled both periodically and in response to aperiodic events. The length of the periodic cycle is related to the basic radar update time interval, which is 4–12 seconds long. In Center airspace, experience has shown the scheduler update cycle must be a small multiple of the radar update cycle, or in the range of 6–12 seconds. Aperiodic events requiring an immediate update of the schedule are primarily due to controller inputs such as a change in airport configurations, a change in airport landing capacity, etc. While controllers prefer a nearly instantaneous response of the scheduler to such inputs, in practice a response within 10–15 seconds has been found to be acceptable and qualifies as real-time performance.

The objective of minimizing delays would require mathematical optimization to be performed by the scheduler in real time. However, at the time of the original publication of this paper in 1995, it was recognized that an algorithmic solution of the full scheduling optimization problem that includes all important constraints is infeasible and probably impossible in real time. In light of this situation, numerous studies have been done to synthesize practical algorithms that combine both adequate scheduling efficiency and short computation times, so as to maintain real-time performance. Subsequent work in optimization of arrivals schedules over last 20 years has shown minimal improvements to the FCFS approach when air traffic control's constraints are considered.

The scheduling principle underlying all practical real-time scheduling algorithms that have so far been developed is referred to as FCFS (refs. 4 and 11). In general, this principle generates “fair” schedules when delays must be absorbed. It is also known to be an optimum schedule for a simple constraint condition and performance criterion. In the discussions to follow, this principle is therefore the starting point for important aspects of the scheduler design. A precise definition of FCFS in the context of air traffic scheduling will be given later.

In addition to the FCFS principle, the scheduling problem is characterized by numerous constraints. The complexity of the scheduling algorithm that remains true to the FCFS principle is greatly increased by the presence of these constraints, as will be seen when the algorithm is derived.

Nevertheless, the scheduling algorithm described herein generates a feasible FCFS schedule without computationally lengthy iterative procedures, thereby achieving the precondition for real-time operation.

## 2.4 Position Shifting

The basic sequence-order constraints for which the scheduling algorithm will be derived consist of FCFS order at the runway and FCFS order for each independent stream at each meter gate. A sequence list of aircraft not in FCFS order is said to be position-shifted. Let  $\{ETA_{(i)}\}_N$  be the set of estimated times of arrival for the set of  $N$  aircraft  $\{A_i\}_N$  at the runway threshold, where the  $A_i$  are the aircraft identifiers. Then the FCFS order at this point is the time-ordered list of this set of ETAs arranged in a vertical column, as shown in table 3. By convention and for economy of notation, the earliest estimated time of arrival ( $ETA$ ) at the bottom of the list is associated with aircraft  $A_1$ , while the latest  $ETA$  at the top is associated with aircraft  $A_N$ .

A position-shifted order can be displayed graphically by placing the FCFS order list of ( $ETA$ )s and the position-shifted list of aircraft identifiers adjacent to each other and then connecting corresponding ( $ETA$ )s and aircraft identifiers with lines as shown in table 4. The crossed lines identify the aircraft that are position-shifted.

In Table 4,  $A_2$  and  $A_3$  are position-shifted by one, meaning that an order reversal of these two adjacent aircraft returns them to FCFS order. Higher-order position shifts would appear as multiple line crossings. If advancing an aircraft by  $k$  slots relative to FCFS is defined as a positive position shift of  $k$  and delaying it by  $m$  slots is defined as a negative position shift of  $m$ , then it can be shown that the algebraic sum of all position shifts of an arbitrarily position-shifted sequence order is zero.

The algorithm can easily be adapted to accommodate position-shifted sequence order at the runway or the meter gates. Position shifting is a technique for reducing delays by optimizing the landing sequence; it is discussed in Chapter 4.

**TABLE 3. FCFS ORDERED LIST OF (ETA)S**

$ETA(A_N)$ (latest)
:
$ETA(A_4)$
$ETA(A_3)$
$ETA(A_2)$
$ETA(A_1)$

**TABLE 4. ILLUSTRATION OF POSITION-SHIFTED SEQUENCE ORDER RELATIVE TO FCFS ORDER**

<u>FCFS</u>	<u>Position-Shifted Order</u>
$ETA(A_N)$	$A_N$
:	
$ETA(A_4)$	$A_4$
$ETA(A_3)$	$A_2$
$ETA(A_2)$	$A_3$
$ETA(A_1)$	$A_1$

## 2.5 “Freeze Horizon”

The freeze horizon gives air traffic controllers sufficient time and airspace to maneuver aircraft so they will cross the feeder gate/meter gate at the scheduled times.

In order to achieve an efficient sequence order in the real-time process and for other reasons that will be explained, scheduled times to cross the meter gate, initial landing times, and runway assignments must be made well before aircraft cross the meter gate. This process is achieved by specifying a freeze-horizon time interval,  $\Delta_F$ , which is the estimated time interval for aircraft to fly from the freeze horizon to the feeder/meter gate while they are following their nominal trajectories. Typical values for  $\Delta_F$  lie in the range of 15–20 minutes. For a value of time,  $t$ , when the estimated time remaining for an aircraft to cross the meter gate first becomes equal to (or less than)  $\Delta_F$ , it is said that the aircraft has crossed the freeze horizon,  $t_F$ , as shown in equation (2):

$$ETA_{FF} - t_F \leq \Delta_F \quad (2)$$

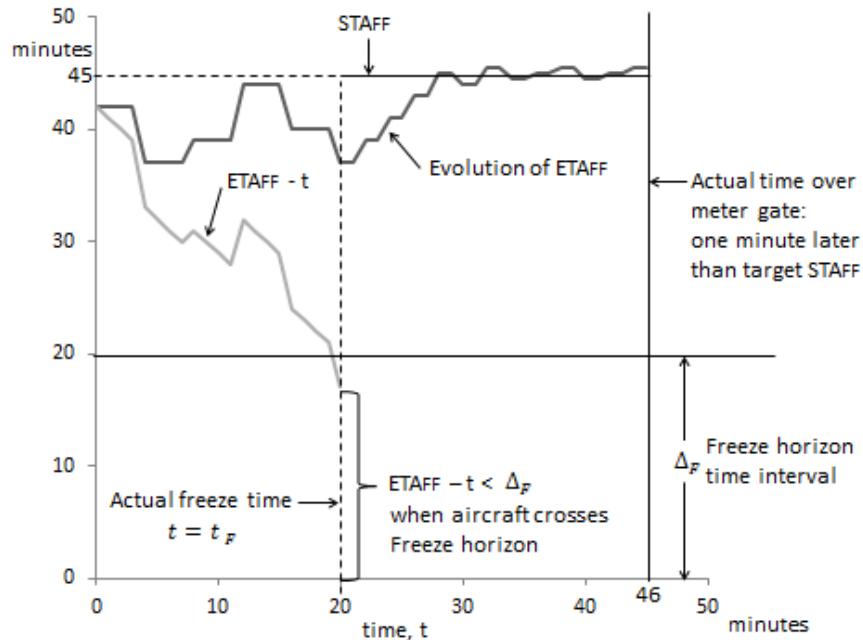
When that time is reached, the scheduler generates the scheduled time of arrival,  $STA_{FF}$ , at the feeder/meter gate. These concepts are illustrated in figure 2 for an example  $ETA_{FF}$  update history. Note that the change in the  $ETA_{FF}$  at each update reflects the noise in the trajectory-prediction process that generates the  $ETA_{FF}$ s. The jumps in the  $ETA_{FF}$ s at update times account for the fact that the crossing of the freeze horizon generally occurs at values of time that yield less than the specified interval  $\Delta_F$  for the freeze horizon.

Controllers require the freeze horizon to be held nearly constant, within 1 or 2 minutes of the nominal time. For example, a nominal time is 20 minutes in figure 2.

## 2.6 Role of Freeze Horizon in Real-Time Scheduler Design

Schedule-optimization problems are closely related to the well-known traveling salesman problem. Both types of problems give rise to search procedures that exhibit polynomial growth rates in computing time as the number of schedulable aircraft increases. Such procedures become computationally impractical to implement in real-time applications for all but a small number of schedulable aircraft.

As discussed in section 2.3, FCFS has a direct applicability measure of fairness. To shed further light on the nature of these problems, consider a FCFS ordered set of schedulable aircraft as shown in table 3. The optimization objective of interest in scheduling is the minimization of the sum of delays of all aircraft by position shifting and runway assignment. No computationally fast algorithms are known and none are thought to exist that can generate the optimum solution by operating sequentially on this time-ordered set, starting with the earliest  $ETA$  aircraft. In general, the character of the optimum-solution procedure is such that the choice of position shifts and runway assignments made at the beginning of the set cannot be made in isolation of, and are therefore interdependent with, such choices at the end of the set.



- 1)  $ETA_{FF}$  generally changes with each trajectory update due to prediction errors ( $t = 0$  to  $20$  in fig. 2). In figure 2, the update interval of  $ETA_{FF}$  before freeze horizon is 4 minutes (min).
- 2)  $STA_{FF}$  is calculated when  $ETA_{FF} - t \leq \Delta_F$  at crossing of the freeze horizon ( $t = 20$  in fig. 2).
- 3) Starting at freeze horizon, air traffic controllers issue clearances to the pilot that change the trajectory so as to converge  $ETA_{FF} \rightarrow STA_{FF}$  as aircraft approaches the feeder/meter gate ( $t = 20$  to  $46$  in fig. 2). After freeze, the update interval of  $ETA_{FF}$  is 2 min.
- 4) An aircraft actually crosses the feeder/meter gate at  $t = 46$  min, which is 1 minute later than the target  $STA_{FF}$ .

**Figure 2.-Explanation of “freeze horizon”.**

Now assume that the true optimum solution could be obtained for the whole set of schedulable aircraft converging on an airport by some superfast computer. Such a solution would, however, be of limited practical value in a real-time air-traffic-control environment for two reasons. First, unavoidable, unknowable, and time-varying errors in the computation of the ( $ETA$ )s upon which the optimum solution is based render the solution non-optimum even if it is computable. Since  $ETA$  errors grow approximately proportional with the time-to-fly to the airport, the degree of non-optimality grows with increasing time-to-fly to the meter gate, or airport. Second, even if the optimum solution were known, it may not be achieved in practice because of operational constraints (e.g., limited ability to perform numerous optimal control actions of air traffic controllers and pilots) inherent in the human-centered air-traffic-control process.

The necessity for a stable freeze horizon together with the inevitability of errors in ( $ETA$ )s enable a crucial simplification in the formulation of the scheduling-optimization problem. Instead of having to include a large number of aircraft in the combinatorial search (thereby creating computational overload) as originally envisioned, only the few aircraft that, at any time, are

within a narrow time range near the freeze horizon need to be considered in such a search. In practice, the number of such aircraft can be limited to fewer than five without incurring a significant loss in efficiency.

Thus, in conclusion, a careful examination of the actual operational environment for scheduling and control of arrival traffic permits a simplification of what initially appeared to be an intractable optimization problem to one that is computationally feasible, although not optimal, for a real-time scheduler.

## Chapter 3

### *Nomenclature*

$A, B$	=	identifier of meter gates
$A_i$	=	identifier of the $i$ th arrival aircraft at meter gate $A$
$A_{ik}$	=	identifier of $A_i$ aircraft, which is the $k$ th arrival aircraft at a runway
$B_j$	=	identifier of the $j$ th arrival aircraft at meter gate $B$
$B_{jl}$	=	identifier of $B_j$ aircraft, which is the $l$ th arrival aircraft at a runway
$C_p$	=	identifier of the $p$ th arrival aircraft at a runway
$C_{(R1)p}$	=	identifier of the $p$ th arrival aircraft at a runway $R1$
$C_{(R2)p}$	=	identifier of the $p$ th arrival aircraft at a runway $R2$
$d$	=	time delay between a scheduled time of arrivals at a runway threshold, $STA$ , and a required time of arrivals at a runway threshold, $RTA$
$d_{Tmax}$	=	maximum value of time delay that can be handled/absorbed in TRACON airspace
$DDF_C$	=	Center delay distribution function
$DDF_T$	=	TRACON delay distribution function
$ETA$	=	estimated time of arrivals at a runway threshold
$ETA_{FF}$	=	estimated time of arrivals at a meter gate
$(ETA_{FF})s$	=	more than one $ETA_{FF}$
$M$	=	total number of aircraft $A_i$ to be scheduled
$N$	=	total number of aircraft $B_j$ to be scheduled
$R1, R2$	=	identifier of runways
$RP$	=	identifier of preferred runway
$RTA$	=	runway time of arrival of an aircraft before separation constraints are applied
$(RTA)s$	=	more than one $RTA$
$RTA_{R1}$	=	$(RTA)s$ at a runway threshold of runway $R1$
$RTA_{R2}$	=	$(RTA)s$ at a runway threshold of runway $R2$
$RTA_{RP}$	=	$(RTA)s$ at a runway threshold of preferred runway
$STA$	=	scheduled time of arrivals at a runway threshold
$(STA)s$	=	more than one $STA$
$STA_{FF}$	=	$(STA)s$ at a meter gate
$(STA_{FF})s$	=	more than one $STA_{FF}$
$STA_{FFit}$	=	$(STA)s$ with in-trail constraints at a meter gate
$(STA_{FFit})s$	=	more than one $STA_{FFit}$
$S_l$	=	actual slot loss
$T_{it}$	=	minimum in-trail time separation at runway, dependent on aircraft weight classes of leading and trailing aircraft
$(T_{it})s$	=	more than one $T_{it}$
$T_t$	=	transition time of arrival aircraft in TRACON from a meter gate to a runway threshold
$(T_t)s$	=	more than one $T_t$
$t_{it}$	=	minimum in-trail separation of aircraft crossing a feeder gate
$T_{tR1}$	=	transition time of arrival aircraft in TRACON from a meter gate to a runway threshold of runway $R1$
$T_{tR2}$	=	transition time of arrival aircraft in TRACON from a meter gate to a runway threshold of runway $R2$

### 3. Basic Algorithms for Scheduling Time of Arrivals

#### 3.1 Description of the Basic Algorithms

This section describes the basic algorithms that generate schedules to the runway threshold while obeying FCFS sequence constraints at both meter gates and runways. The algorithms build the schedule by a non-iterative constructive procedure that translates directly into a rapidly executing software program. While the algorithms minimize aircraft required separation subject to constraints, they do not optimize any specific performance functions. Optimization approaches are briefly discussed in chapter 4.

#### 3.2 Multiple Meter Gates to One Runway

This section describes basic algorithms for arrival aircraft from multiple meter gates to a runway.

It is assumed that the schedulable aircraft are in Center airspace and some distance away from their meter gates. The basic input to the scheduler is the set of estimated times of arrival of all schedulable aircraft, computed to the appropriate meter gates. This set, designated by  $\{ETA_{FF}\}$ , is provided by the trajectory synthesizer algorithm (ref. 2). For the sake of simplicity but without loss of generality, the derivation is given for the case of two meter gates,  $A$  and  $B$ , and one runway. Aircraft assigned to these gates have associated identifiers  $\{A_i\}_M$  and  $\{B_j\}_N$ , respectively.

Thus  $M + N$  are the total numbers of aircraft to be scheduled. Let  $\{T_t(A_i)\}_M$  and  $\{T_t(B_j)\}_N$  be the set of TRACON transition times. They specify the nominal time intervals required for aircraft to fly from their respective meter gates,  $A$  or  $B$ , to the runway threshold. Therefore, the estimated times of arrival of aircraft  $A_i$  and  $B_j$  at the runway threshold can be written as

$$\left. \begin{aligned} ETA(A_i) &= ETA_{FF}(A_i) + T_t(A_i) \\ ETA(B_j) &= ETA_{FF}(B_j) + T_t(B_j) \end{aligned} \right\} (3)$$

The set of transition times  $\{T_t(A_i)\}_M$  and  $\{T_t(B_j)\}_N$  are input quantities also generated by the trajectory synthesizer algorithm.

A series of time lines, shown in figures 3a through 3e, are used to illustrate various steps in the development of the scheduling algorithm. Each figure in the series consists of vertical time lines arranged side-by-side representing a geographic scheduling point, either a meter gate or a runway. The transformation and procedures described in the various steps are represented graphically by lines connecting objects on adjacent time lines. The objects are generally the aircraft to be scheduled. By studying the figures in sequence, the reader can follow a specific scheduling problem from beginning to end, steps 1 to 5.

##### **Step 1: Apply in-trail separation constraints at meter gates.**

Let the set of scheduled times of arrival at the meter gates with in-trail constraints  $\{t_{it}\}$  be  $\{STA_{FFit}\}$ . Generate the  $(STA_{FFit})$ s sequentially at each meter gate starting with the earliest-to-arrive aircraft  $A_1$  and  $B_1$  at gates  $A$  and  $B$ , respectively:

$$\begin{aligned}
STA_{FFit}(A_1) &= ETA_{FF}(A_1) \\
STA_{FFit}(A_2) &= \text{Greater of } \begin{cases} ETA_{FF}(A_2) \\ STA_{FFit}(A_1) + t_{it}(A_1, A_2) \end{cases} \\
&\vdots \\
STA_{FFit}(A_M) &= \text{Greater of } \begin{cases} ETA_{FF}(A_M) \\ STA_{FFit}(A_{M-1}) + t_{it}(A_{M-1}, A_M) \end{cases}
\end{aligned}
\right\} (4)$$

$$\begin{aligned}
STA_{FFit}(B_1) &= ETA_{FF}(B_1) \\
STA_{FFit}(B_2) &= \text{Greater of } \begin{cases} ETA_{FF}(B_2) \\ STA_{FFit}(B_1) + t_{it}(B_1, B_2) \end{cases} \\
&\vdots \\
STA_{FFit}(B_M) &= \text{Greater of } \begin{cases} ETA_{FF}(B_M) \\ STA_{FFit}(B_{M-1}) + t_{it}(B_{M-1}, B_M) \end{cases}
\end{aligned}
\right\} (5)$$

Here, the in-trail constraint could be a constant or a function of the aircraft types in sequence. This step is illustrated in figure 3a.

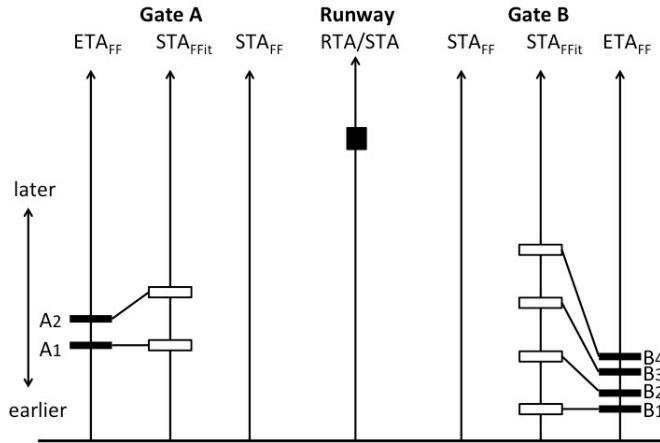


Figure 3a.-Adding in-trail constraints at the meter gates (step 1).

### **Step 2: Determine the runway threshold landing order.**

As previously stated, the overall objective is to generate a FCFS landing order at the runway. However, when in-trail constraints are present at the meter gates, such as those described in step 1, the definition of FCFS at the runway becomes ambiguous because arrivals enter the terminal airspace via multiple meter gates. The ambiguity is removed by choosing the ( $STA_{FFit}$ )s rather than the ( $ETA_{FF}$ )s when establishing the FCFS order at the runway. Simulation and analysis have shown this choice produces both a fairer schedule overall as well as one that is slightly more efficient.

The process begins by propagating the ( $STA_{FFit}$ )s forward in time from the gates to the runway by using the TRACON transition times. If  $RTA(A_i)$  and  $RTA(B_j)$  designate the runway times of arrival of aircraft  $A_i$  and  $B_j$ , then:

$$RTA(A_i) = STA_{FFit}(A_i) + T_t(A_i) \quad (6)$$

$$RTA(B_j) = STA_{FFit}(B_j) + T_t(B_j) \quad (7)$$

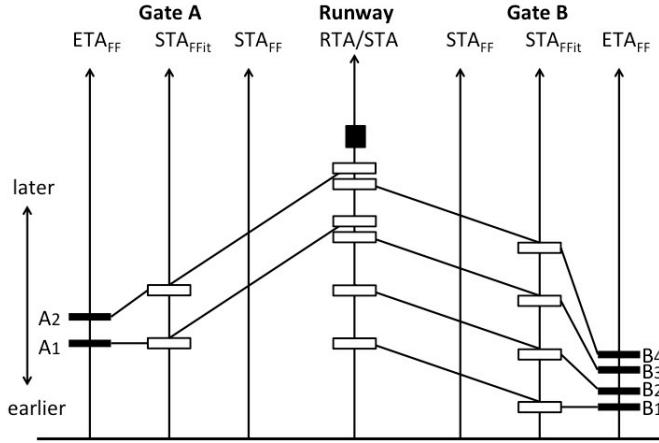
Repeating this process for all schedulable aircraft results in two sets:  $\{RTA(A_i)\}_M$  and  $\{RTA(B_j)\}_N$ . The times in these sets represent the earliest possible landing times of the schedulable aircraft when in-trail constraints at the gates are included but in-trail constraints on final approach are ignored.

Before the FCFS landing order is determined from the sets of ( $RTA$ )s, an order-rectifying procedure must first be performed for the following reason: Because different aircraft types can have substantially different TRACON transition times, the order of the sets  $\{RTA(A_i)\}_M$  and  $\{RTA(B_j)\}_N$  does not necessarily match the FCFS order at meter gates. That is, the  $RTA$  order could become position-shifted relative to the  $STA_{FFit}$  order for aircraft passing through the same gate. Such a position shift occurs when a sufficiently faster aircraft is able to overtake a slower aircraft on the same route within TRACON airspace. Overtakes of aircraft with similar performance flying from the same gate to the same runway on the same route are generally not acceptable to controllers and must be excluded by the scheduling algorithms. Overtakes are allowed for aircraft with dissimilar speed profiles flying on separate routes (e.g., jet and turboprop). Thus, it is necessary to check each set of ( $RTA$ )s for position-shifted sequences relative to their feeder gate crossing order. If such sequences are found, the  $T_t$  of each overtaking aircraft is increased by the smallest time increment that modifies the ( $RTA$ )s so as to restore them to FCFS ordered sequence. In practice this process requires the faster aircraft to reduce speed or increase its path length to the runway. It is now assumed that the  $\{RTA(A_i)\}_M$  and  $\{RTA(B_j)\}_N$  have already been rectified in this manner and are, therefore, in FCFS order.

The runway landing order list,  $\{C_p\}_{M+N}$ , is now obtained by merging the two sets of ( $RTA$ )s into a FCFS time-ordered sequence list:

$$\{C_p\}_{M+N} = \{A_{ik}, B_{jl}\}_{M+N} \quad (8)$$

where the second indices  $k, l$  indicate the landing order. The indices satisfy the inequalities  $k \geq i$ ,  $l \geq j$  over their range of values. Figure 3b illustrates the merging and ordering process. Note that no lines starting from the same gate that connect gate sequences and landing sequences cross, as required by the overtake prohibition. There is the special case in which  $RTA(A_i) = RTA(B_j)$ ; this equation can easily be solved by using another discriminator such as call-sign numeric order.



**Figure 3b.-Determining landing order (step 2).**

**Step 3: Compute scheduled times of arrival at the runway threshold.**

In this step the time separations between the unconstrained runway times, the (*RTA*)s, are stretched, when necessary, to conform to the minimum time-separation matrix given in table 2. This stretching yields the scheduled times of arrival, the (*STA*)s at the runway threshold.

The process involves inserting the appropriately chosen minimum time separation,  $T_{it}$ , from table 2, between pairs of aircraft in sequence starting with the first aircraft in the known landing order, and terminating with the last. This process can be written symbolically as shown in equation (9).

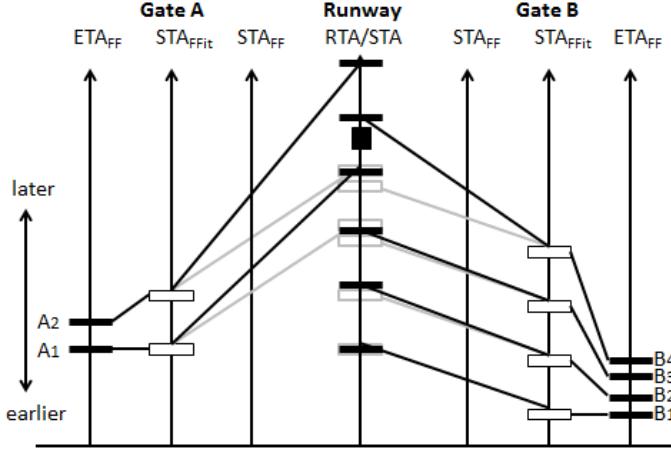
$$\begin{aligned}
 STA(C_1) &= RTA(C_1) \\
 STA(C_2) &= \text{Greater of } \left\{ \begin{array}{l} RTA(C_2) \\ STA(C_1) + T_{it}(C_1, C_2) \end{array} \right. \\
 &\vdots \\
 STA(C_{M+N}) &= \text{Greater of } \left\{ \begin{array}{l} RTA(C_{M+N}) \\ STA(C_{M+N-1}) + T_{it}(C_{M+N-1}, C_{M+N}) \end{array} \right.
 \end{aligned} \quad \left. \right\} (9)$$

If the (*RTA*)s are closely bunched, thus requiring the ( $T_{it}$ )s to be inserted for all or a portion of the schedulable aircraft, the process performed in equation (9) propagates a delay ripple for successive aircraft in the landing order. The delay ripple terminates when a sufficiently large time gap occurs between successive (*RTA*)s. The delay  $d(C_p)$  generated by equation (9) for the  $p$ th aircraft in the landing order can be written as:

$$\begin{aligned}
 d(C_p) &= STA(C_p) - RTA(C_p) \\
 &= STA(C_p) - STA_{FFit}(C_p) - T_t(C_p)
 \end{aligned} \quad (10)$$

In addition to the scheduling constraints already described, two other types of scheduling constraints, referred to as blocked intervals and reserved time slots on the runway, must also be handled by the scheduling algorithm. They are specified time intervals and virtual aircraft landing times that the scheduling algorithm must avoid when generating the (*STA*)s for the list of schedulable aircraft. The logic in equation (9) can be extended in to handle these constraints.

The processes described in this step are illustrated by the example in figure 3c. A blocked time interval has been included as a constraint.



**Figure 3c.-Determining the runway (STA)s (step 3).**

#### **Step 4: Develop the Delay Distribution Function.**

Whenever the total flow rate to a runway exceeds a certain maximum rate for a significant period of time, the separation constraints imposed by equation (9) will generate large delays. When that occurs it is said that the rate exceeds the runway capacity. Up to this point in the development of the algorithm, all delays, no matter how large, would be absorbed between the meter gates and the runway. However, a group of aircraft in sequence, each with delays of several minutes to absorb in the TRACON airspace, creates excessive workload for TRACON controllers and can produce potentially unsafe operational conditions. Center and TRACON traffic managers work diligently to control this congestion in the TRACON airspace. Analogously, the scheduling algorithm needs a mechanism for controlling congestion of TRACON airspace due to excessive delay buildup.

This step describes a procedure for distributing delay between Center and TRACON airspace. The procedure involves the use of two functions referred to as Center and TRACON delay distribution functions  $DDF_C$  and  $DDF_T$ , respectively, as follows:

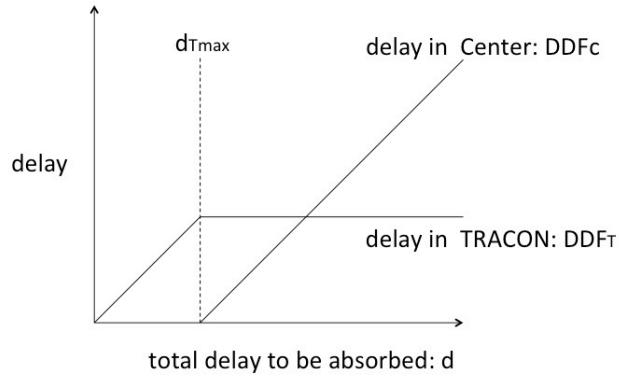
$$DDF_C(d) = \begin{cases} 0, & d \leq d_{Tmax} \\ d - d_{Tmax}, & d > d_{Tmax} \end{cases} \quad (11)$$

$$DDF_T(d) = \begin{cases} d, & d \leq d_{Tmax} \\ d_{Tmax}, & d > d_{Tmax} \end{cases} \quad (12)$$

where  $d_{Tmax}$  is a parameter that specifies the maximum delay an aircraft is permitted to absorb in the TRACON airspace. As required, the sum of the two functions just equals the delay to be absorbed:

$$DDF_C(d) + DDF_T(d) = d \quad (13)$$

for all values of  $d$ . The two functions are plotted in figure 3d. These functions are evaluated by substituting into them the delay,  $d$ , of each scheduled aircraft as computed by equation (10). Furthermore, the parameter  $d_{Tmax}$  is itself a function that depends on the meter gate through which an aircraft passes. The meter-gate dependency allows tailoring the delay-absorption parameter to the length of the nominal (un-delayed) path between meter gate and runway. In general, the shorter the nominal path length (more precisely, the TRACON transition time,  $T_t$ ) the less is the maximum delay that can be absorbed along the path. In chapter 5, a method for choosing appropriate values for  $d_{Tmax}$  is derived.



**Figure 3d.-Delay distribution function (step 4).**

#### **Step 5: Compute scheduled times of arrival at the meter gates.**

This step describes the procedures for combining the values of the Center delay distribution of step 4, the scheduled times of arrival at the runway of step 3, and the meter-gate sequence order of step 2 in order to generate the ( $STA_{FF}$ )s, the scheduled times of arrival at the meter gates.

In brief, the procedure consists of a push-back of the ( $STA_{FFit}$ )s by an amount of time calculated from the Center delay distribution. It may also be thought of as a backward propagation of delays from TRACON to Center airspace. The push-back is done sequentially for aircraft at each meter gate, starting with the earliest to cross its meter gate. This procedure preserves the feeder gate crossing order determined in step 1. The push-back procedure is done for all aircraft at each of the meter gates. It does not change the landing order determined in step 2 nor the landing times in step 3. Furthermore, the pushback procedure performed for the aircraft from one gate is independent of that performed for aircraft from other gates.

The procedure begins with the first aircraft to cross a chosen feeder gate. Let that aircraft be  $B_1$  at gate  $B$ , which is also the first to land in figure 3e. As the first aircraft scheduled to land, it has zero delay. The ( $STA_{FF}$ )s for all the aircraft crossing the meter gates  $A$  and  $B$  can then be generated sequentially for each gate, as shown in equations (14)–(19): equations (14)–(16) are for arrivals crossing meter gate  $A$ , equations (17)–(19) are for arrivals crossing meter gate  $B$ .

$$STA_{FF}(A_1) = ETA_{FF}(A_1) + DDF_c(A_1) \quad (14)$$

$$STA_{FF}(A_2) = \text{Greater of } \begin{cases} STA_{FF}(A_1) + t_{it}(A_1, A_2) \\ STA_{FFit}(A_2) + DDF_c(d(A_2)) \end{cases} \quad (15)$$

$$\begin{aligned} STA_{FF}(A_3) &= \text{Greater of } \begin{cases} STA_{FF}(A_2) + t_{it}(A_2, A_3) \\ STA_{FFit}(A_3) + DDF_c(d(A_3)) \end{cases} \\ &\vdots \\ STA_{FF}(A_M) &= \text{Greater of } \begin{cases} STA_{FF}(A_{M-1}) + t_{it}(A_{M-1}, A_M) \\ STA_{FFit}(A_M) + DDF_c(d(A_M)) \end{cases} \end{aligned} \quad (16)$$

$$STA_{FF}(B_1) = STA(B_1) - T_t(B_1) = ETA_{FF}(B_1) \quad (17)$$

$$\begin{aligned} STA_{FF}(B_2) &= \text{Greater of } \begin{cases} STA_{FF}(B_1) + t_{it}(B_1, B_2) = STA_{FFit}(B_2) \\ STA_{FFit}(B_2) + DDF_c(d(B_2)) \end{cases} \\ &= STA_{FFit}(B_2) + DDF_c(d(B_2)) \end{aligned} \quad (18)$$

$$\begin{aligned} STA_{FF}(B_3) &= \text{Greater of } \begin{cases} STA_{FF}(B_2) + t_{it}(B_2, B_3) \\ STA_{FFit}(B_3) + DDF_c(d(B_3)) \end{cases} \\ &\vdots \\ STA_{FF}(B_N) &= \text{Greater of } \begin{cases} STA_{FF}(B_{N-1}) + t_{it}(B_{N-1}, B_N) \\ STA_{FFit}(B_N) + DDF_c(d(B_N)) \end{cases} \end{aligned} \quad (19)$$

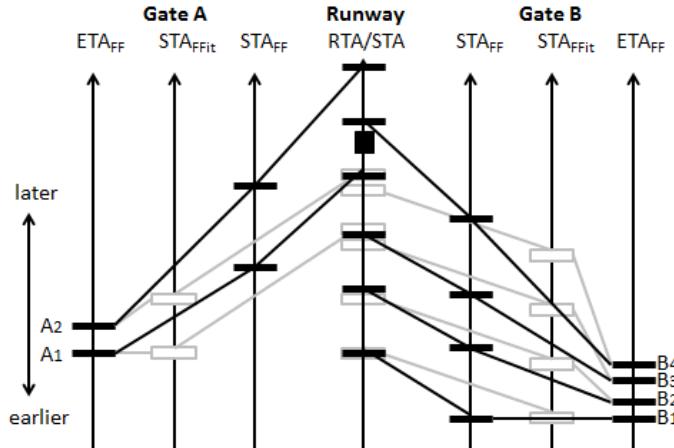


Figure 3e.-Schedule time of arrivals at meter gates using delay distribution function (step 5).

The gate that supplies the first aircraft to land will have zero delay and can be the only aircraft that has zero delay distribution. In this situation no other aircraft interfere, by separation constraints, at the meter gate or runway  $ETA$ , this being the first aircraft  $B_1$ . All other aircraft can

have non-zero delay distributions. In the example shown in figure 3e, that first aircraft originates from gate  $B$ . Therefore, the  $B_1$ 's  $STA_{FF}$  is equal to its  $ETA_{FF}$ , as shown in equation (17).

When aircraft are experiencing large delays, the second of the two quantities in the comparison test of equations (16) and (19) will be the greater of the two and thus will determine the  $STA_{FF}$ . However, in practice, the parameters  $t_{it}$  and  $DDF_C$  can assume such combinations of values that the first quantity becomes the greater of the two. By choosing the first quantity as the  $STA_{FF}$  in that case, the logical condition “greater of” ensures that the FCFS meter-gate sequence is preserved.

#### **Pushback-adjusted FCFS order method**

In the basic method described previously, first in steps 1 and 2, the landing orders of all schedulable aircraft are determined based on  $RTA$  at a runway threshold by forward propagation. Secondly, in steps 3–5, scheduled times of arrival at a runway threshold,  $STA$ , are propagated by forward propagation and at a meter gate,  $STA_{FF}$ , by backward propagation.

The pushback process described here suggests an alternate method for generating landing order and scheduled times. Instead of determining the landing order for all schedulable aircraft first, as in step 2, and then applying the delay distribution, in the alternate method the landing order is generated sequentially during the push-back process and is, therefore, referred to as the pushback-adjusted FCFS order method (PAFCFS). In the PAFCFS method the delay distribution for an aircraft that has been scheduled influences the landing order and time of the next aircraft to be scheduled; this situation is not the case in the original method. A difference in order between the methods is seen only if traffic flows from more than one feeder gate and/or the  $T_t$  are different.

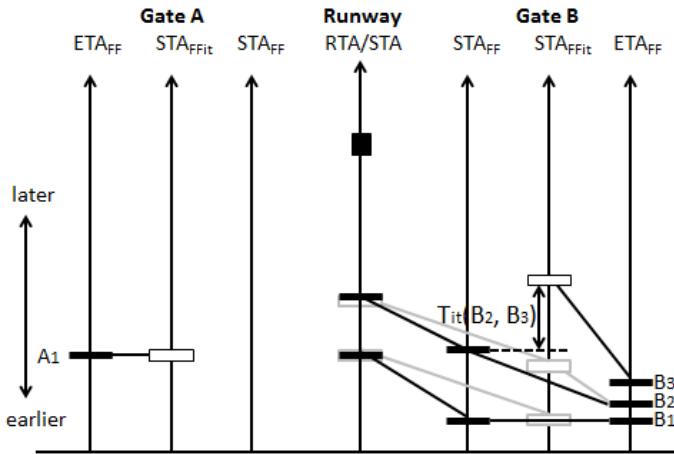
Figures 4a–4c illustrate the graphical construction of the schedule for the PAFCFS method. The pushback of meter-gate time is illustrated for  $A_1$  and  $B_3$ .

In the PAFCFS method, the landing order of the first and second aircraft is generally unchanged. Therefore, the ( $STA$ )s and ( $STA_{FF}$ )s for these two aircraft are still determined by equations (14), (15), (17), and (18) and their values remain unchanged. To determine the third aircraft to be scheduled to land, select the next aircraft in the FCFS sequence order at each meter gate. Following the example sequence in figure 4a, the next aircraft at gate  $A$  is  $A_1$ , and at gate  $B$  it is  $B_3$ . Then compute the in-trail meter gate times for these aircraft by equations (4) and (5):

$$STA_{FFit}(A_1) = ETA_{FF}(A_1) \quad (20)$$

$$STA_{FFit}(B_3) = STA_{FF}(B_2) + t_{it}(B_2, B_3) \quad (21)$$

Note that  $STA_{FF}(B_2)$  includes the center delay distribution from equation (13).



**Figure 4a.-Determining landing order of the first and second aircraft in the PAFCFS.**

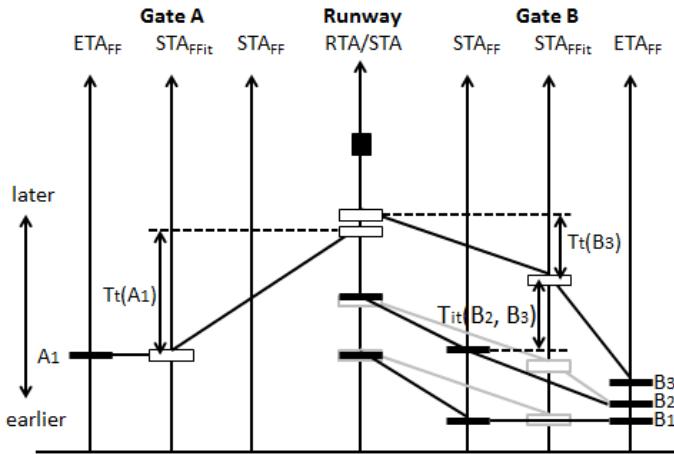
Next, compute the earliest unconstrained runway times for this pair:

$$RTA(A_1) = STA_{FFit}(A_1) + T_t(A_1) \quad (22)$$

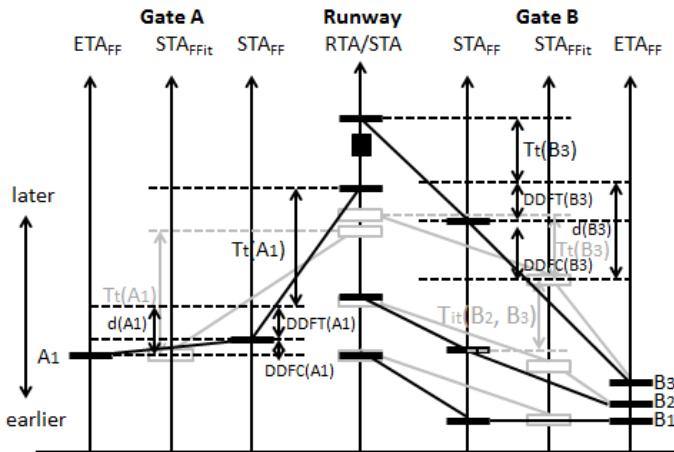
$$RTA(B_3) = STA_{FFit}(B_3) + T_t(B_3) \quad (23)$$

The next aircraft scheduled to land is now chosen between  $A_1$  and  $B_3$  as the one with the earlier  $RTA$ : *lesser of*  $\{RTA(A_1), RTA(B_3)\}$ .

In the example of figure 4b, the next aircraft is  $A_1$ , representing a change in order compared to the original method as shown in figure 3e. The computation of  $STA$ ,  $DDF_C$ , and  $STA_{FF}$  for the aircraft so selected ( $A_1$  in fig. 4b) now parallels the previously described method. Thus, in contrast to the original method where the landing order was determined at the end of step 2 for all aircraft, the landing order here is influenced by the pushback delay of the previously scheduled aircraft. Figure 4c shows the final schedule for this example.



**Figure 4b.-Determining landing order of the potential third aircraft in the PAFCFS.**



**Figure 4c.-Determining landing order by pushback of (STA<sub>FF</sub>)s using DDF in the PAFCFS.**

Comparison of the PAFCFS order with the original order using fast time simulations shows that the PAFCFS method tends to advance the landing order of aircraft from gates with lower flow rates relative to those from gates with higher flow rates. While this may be seen as less fair than the original method, the fast time simulation also shows it to yield slightly lower average delays. This PAFCFS approach was adopted by the implementation described in reference 12 and implemented in reference 9; it is employed in the current operational TMA because realistically traffic flows from more than one feeder gate, and transition times in TRACON ( $T_t$ ) are different.

### 3.3 Multiple Terminal Gates to Multiple Runways

In this section the basic algorithm is extended to handle the scheduling of aircraft to an airport with several landing runways. All large airports use at least two and as many as four landing runways simultaneously under certain traffic and weather conditions. Similar to the basic algorithm, the objective here is to generate efficient initial schedules for the multiple-runway case without time-consuming iterative procedures. These schedules and runway assignments can then

be used as the starting solution for optimizing procedures if real-time computational constraints permit. The guiding principle of the runway-assignment process as developed here is to assign and schedule aircraft sequentially to the runway that gives the earliest landing time while minimizing loss of full or fractional landing slots.

First it is necessary to generalize the definition of FCFS at the runway for the multiple-runway situations. Begin by completing step 1 for the list of schedulable aircraft. Then compute the unconstrained runway times of arrival, the ( $RTA$ )s as in equations (6) and (7) of step 2 for each runway. In order to avoid complex symbology, the following development assumes two landing runways, designated as  $R1$  and  $R2$ .

Thus, for the gate  $A$  and  $B$  aircraft, one can write:

$$RTA_{R1}(A_i) = STA_{FFit}(A_i) + T_{tR1}(A_i) \quad (24)$$

$$RTA_{R2}(A_i) = STA_{FFit}(A_i) + T_{tR2}(A_i) \quad (25)$$

$$RTA_{R1}(B_j) = STA_{FFit}(B_j) + T_{tR1}(B_j) \quad (26)$$

$$RTA_{R2}(B_j) = STA_{FFit}(B_j) + T_{tR2}(B_j) \quad (27)$$

Then, for each aircraft, define the preferred runway,  $RP$ , as the one having the lesser of the ( $T_t$ )s:

$$\begin{aligned} RP(A_i) &= \{R1 \quad \text{or} \quad R2\}, \\ &\text{corresponding to lesser of } \{T_{tR1}(A_i), \quad T_{tR2}(A_i)\} \end{aligned} \quad (28)$$

$$\begin{aligned} RP(B_j) &= \{R1 \quad \text{or} \quad R2\}, \\ &\text{corresponding to lesser of } \{T_{tR1}(B_j), \quad T_{tR2}(B_j)\} \end{aligned} \quad (29)$$

and similarly for aircraft from all other gates. Then, the FCFS order  $\{C_p\}_{M+N}$  is defined as the merged and time-ordered set of the corresponding ( $RTA_{RP}$ )s:

$$\begin{aligned} \{C_p\}_{M+N} &= \{A_{ik}, \quad B_{jl}\}_{M+N} \\ &\text{in order of increasing time of corresponding } RTA_{RP}, \\ &\{RTA_{RP}(A_{ik}), \quad RTA_{RP}(B_{jl})\}_{M+N} \end{aligned} \quad (30)$$

It would now be possible to generate scheduled landing times by inserting the appropriate minimum time separation,  $T_{it}$ , between successive aircraft landing on the same runway, similar to step 3 of the single-runway case. While such a schedule is feasible, it may also be inefficient for the following reason: At hub airports, traffic arrives in rushes from one or at most two directions, causing the one runway with the shortest transition time between the rush traffic meter gate and that runway to become overloaded. This overloading would occur because the FCFS order procedure defined previously leaves all aircraft assigned to their preferred runways even as delays build up on that runway. It is, however, possible to improve upon the preferred runway-

assignment procedure with little additional computation, thereby providing a more efficient starting condition for subsequent runway-assignment optimization steps.

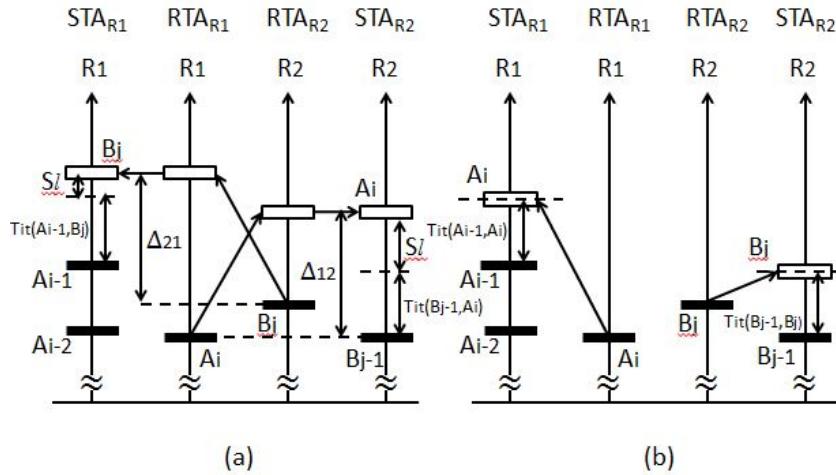
The improved procedure can be used either in conjunction with the preferred runway FCFS order defined previously or with the delay distribution adjusted order described in step 3. Assume to start with that the FCFS order of equation (30) is being followed. Let the next aircraft to be assigned and scheduled be  $A_i$  from gate  $A$ , and let the next aircraft to cross gate  $B$  be  $B_j$ . The preferred runways for these two aircraft are  $R1$  and  $R2$ , respectively. Let's assume  $A_i$  arrives at  $R1$  earlier than  $B_j$  arrives at  $R2$ . Then the RTA that determines the FCFS order follows:

$$RTA_{R1}(A_i) < RTA_{R2}(B_j) \quad (31)$$

If runway assignment is changed from preferred runway  $R1$  to non-preferred runway  $R2$ :

$$RTA_{R2}(A_i) = RTA_{R1}(A_i) + \Delta_{21}(A_i) \quad (32)$$

where  $\Delta_{21}(A_i) = T_{tR2}(A_i) - T_{tR1}(A_i) > 0$  is the increment in time for  $A_i$  to transition from gate  $A$  to the non-preferred runway compared to the preferred runway. Corresponding relationships can also be written for  $B_j$ . Figure 5 illustrates these concepts for an example sequence.



- (a) FCFS order ( $A_i$  scheduled first): Here expected (minimum) delay, *min. delay*, is defined as  $\text{min. delay} = (STA_{R1}(B_j) - RTA_{R2}(B_j)) + (STA_{R2}(A_i) - RTA_{R1}(A_i)) = 10$ . The sum of slot losses is  $\sum S_l = 3$ .
- (b) Non-FCFS order ( $B_j$  scheduled first): Here expected (minimum) delay, *min. delay*, is defined as  $\text{min. delay} = (STA_{R1}(A_i) - RTA_{R1}(A_i)) + (STA_{R2}(B_j) - RTA_{R2}(B_j)) = 7$ . The sum of slot losses is  $\sum S_l = 0$ .

**Figure 5.-Choosing a non-FCFS order and runway assignment that minimizes slot loss.**

The next step is to calculate the (*STA*)s for all combinations of aircraft next in sequence to cross any of the meter gates and all possible landing runways. The runway landing order list at runway *R1*,  $\{C_{(R1)p}\}_{M^*+N^*}$ , is now obtained by merging the two sets of (*RTA*)s into a FCFS time-ordered sequence list:

$$\{C_{(R1)p}\}_{M^*+N^*} = \{A_i, B_j\}_{M^*+N^*} \quad (33)$$

where  $M^* + N^*$  is the total number of aircraft that arrive at runway *R1*.

Each of the (*STA*)s at runway *R1* is calculated using the procedure described in step 3:

$$STA_{R1}(A_i = C_{(R1)p}) = \text{Greater of } \left\{ \begin{array}{l} RTA_{R1}(C_{(R1)p}) \\ STA_{R1}(C_{(R1)p-1}) + T_{it}(C_{(R1)p-1}, C_{(R1)p}) \end{array} \right\} \quad (34)$$

The four (*STA*)s for the example are shown at appropriate locations on the time line in figure 5. At any step in the scheduling process, the characteristics of the relationships between values of *STA* and values of *RTA* influence the strategy for making efficient runway assignments. That process yields two categories of problems, referred to as standard and potential “slot loss”. Slot loss is the loss of a landing opportunity due to an unavoidable delay imposition, which is explained as follows.

### **Standard category**

This category applies when preferred runway (*STA*)s are less than non-preferred runway (*STA*)s and/or all non-preferred runway (*STA*)s are larger than the corresponding (*RTA*)s.

If the FCFS order is being followed, then the next aircraft to be scheduled in that order is assigned to the runway giving the earliest *STA*. If, instead, the pushback adjusted FCFS order is being followed, the (*RTA*)s for aircraft yet to be scheduled are updated after an aircraft has been assigned and scheduled. Then the aircraft from the gate yielding the lowest *RTA* for any eligible runway becomes the next aircraft to be scheduled and is assigned the runway corresponding to the earliest *STA*. Either scheduling-order strategy provides acceptably efficient schedules and runway assignments for this category.

### **Potential slot-loss category**

This category applies when at least one *STA* for a non-preferred runway is equal to the corresponding *RTA* and is also less than the preferred-runway *STA*. This case can be written symbolically as

$$STA_{R2}(A_i) = RTA_{R2}(A_i) \quad (35)$$

$$STA_{R1}(A_i) > RTA_{R2}(A_i) \quad (36)$$

and is illustrated in figure 5a. In this example  $A_i$  is scheduled before  $B_j$  by the FCFS rule and is assigned to its lowest-delay runway R2. Then  $B_j$  is scheduled and assigned its lowest-delay runway R1. Both assignments result in slot losses, as shown in figure 5a.

The potential slot loss referred to here arises from the fact that scheduling an aircraft to a non-preferred runway incurs an unavoidable delay increment of  $\Delta$  seconds compared to scheduling it to the preferred runway. Therefore, the quantity  $\Delta$  establishes the maximum potential slot loss for a non-preferred runway assignment. However, the unavoidable delay increment  $\Delta$  is not a slot loss if the delay that must be absorbed in assigning an aircraft to a non-preferred runway is larger than  $\Delta$  for other reasons, such as meeting in-trail separation constraints with a preceding aircraft. The potential for a slot loss exists only when the earliest possible scheduled times to the two runways satisfy the condition in equation (35). The actual slot loss,  $S_l(A_i)$  as distinguished from the maximum potential slot loss, is computed as follows:

$$S_l(A_i = C_{(R2)p}) = RTA_{R2}(C_{(R2)p}) - STA_{R2}(C_{(R2)p-1}) - T_{it}(C_{(R2)p-1}, C_{(R2)p}) \quad (37)$$

A value of  $S_l > 0$  represents a fractional, or larger, landing-slot opportunity that is wasted unless an aircraft from another meter gate is available and can be scheduled instead of  $A_i$  to occupy a greater portion of that slot. If such an aircraft were available, for example  $B_j$  in figure 5b, then one of the FCFS scheduling-order disciplines that had selected  $A_i$  as the next aircraft to be scheduled would have to be relaxed so  $B_j$  can be scheduled before  $A_i$  instead. When this change is made for the same input conditions as in figure 5a, the runway assignments that minimize the sum of delays for the two aircraft are to assign them to their preferred runways, as shown in figure 5b. This strategy eliminates both slot losses but requires additional computations.

The significance of slot loss derives from its cumulative effect on delays for upstream aircraft during a period of delay buildup, such as at the beginning of a traffic rush. Under such conditions a slot loss can propagate into additional delays for all upstream aircraft until a sufficiently large time gap occurs between aircraft in sequence. Analysis of actual traffic during a rush at a large airport shows that this cumulative effect on delays of the upstream aircraft is between 2 to 4 times as much as the value of the slot loss. Thus, reducing slot loss, especially at the beginning of a rush, gives a large payoff in delay reductions.

As we have seen in the example illustrated in figure 5, allowing non-FCFS scheduling orders to be tried can mitigate the slot-loss problem. This approach can be generalized by computing all possible combinations of scheduling orders and runway assignments for pairs of aircraft. For the two aircraft in the example of figure 5, this computation yields eight combinations, of which two are redundant. Thus, we are left with six unique schedules and runway assignments. From these options we choose the one that minimizes the total delay for the two aircraft. This approach is an example of schedule and runway-assignment optimization discussed in the next chapter in greater detail.

In summary, the procedure described previously for assigning and scheduling aircraft to runways packs aircraft on runways as tightly as in-trail constraints permit, while also minimizing fractional slot losses that can arise from assigning aircraft to non-preferred runways. It avoids an unequal buildup of delays between different runways by shifting aircraft to non-preferred runways when it is efficient to do so. When potential slot losses are detected based on equations (35)-(37) it suggests allowing a limited number of position shifts to occur relative to FCFS order for aircraft from different meter gates in order to reduce slot losses and minimize delays.

# Chapter 4

## Nomenclature

$A_i$	=	identifier of the $i$ th arrival aircraft at meter gate $A$
$B_j$	=	identifier of the $j$ th arrival aircraft at meter gate $B$
$D$	=	delay equivalent cost
$ETA_{FF}$	=	estimated time of arrivals at a meter gate
$(ETA_{FF})_s$	=	more than one $ETA_{FF}$
$K$	=	upper limit of the number of trial schedules
$P$	=	number of aircraft
$Q$	=	number of landing runways
$STA_{FF}$	=	scheduled time of arrivals at a meter gate
$(STA_{FF})_s$	=	more than one $STA_{FF}$
$STA_{FFit}$	=	scheduled time of arrivals with in-trail constraints at a meter gate

## 4. Advanced Algorithms

### 4.1 Real-Time Scheduling Algorithm with Limited Sequence and Runway Assignment Optimization

Section 2.5 explained the need for incorporating a freeze horizon in the design of the real-time scheduler. This chapter extends the basic scheduling algorithm to include both a freeze horizon and a limited degree of schedule optimization that is computationally tractable in real -time.

In addition to the freeze horizon, the optimization horizon and the influence horizon play crucial roles in the real-time algorithm. The three horizons segregate the arrival aircraft into four sets based on the values of the  $(ETA_{FF})_s$  relative to these horizons. The  $ETA_{FF}$  time lines in figure 6 give representative examples of these sets.

#### Freeze Horizon and Freeze-Horizon Time Interval

The Freeze Horizon and Freeze-Horizon Time-Interval  $\Delta_F$  are explained in section 2.5. When the estimated time for an aircraft to fly to the meter gate, as derived from its current  $ETA_{FF}$ , becomes equal to or less than the freeze-horizon time interval, the scheduler freezes the runway assignment and landing sequence at its last computed values.

#### Optimization Horizon and Optimization Interval

The difference in time between the optimization horizon and the freeze horizon equals the optimization interval. Runway assignment and sequence optimization will be performed for the first  $P$  aircraft with  $(ETA_{FF})_s$  in this interval. After runway assignments and landing sequences have been determined for these  $P$  aircraft, they will be frozen simultaneously. The optimization interval has a relatively narrow time range of only 2–5 minutes in order to reduce the variability in the time when schedules are frozen. The narrowness of the time range also ensures the number of aircraft with  $(ETA_{FF})_s$  in the optimization interval will be small, thus reducing the complexity of the optimization.

#### Influence Horizon and Influence Interval

The influence interval is the difference between the influence horizon and the optimization horizon. Only aircraft with  $(ETA_{FF})_s$  less than the influence horizon will be allowed to influence the choice of the runway assignments and landing sequences for the  $P$  aircraft in the optimization

set. Aircraft with  $(ETA_{FF})$ s later than the influence horizon are excluded because they are still too far away and, therefore, their  $(ETA_{FF})$ s are too uncertain to allow these aircraft to influence the runway-assignment process at this time. Their influence will be felt later when these aircraft finally penetrate the influence horizon. Experience with the current level of  $ETA_{FF}$  accuracy suggests that the influence horizon should be about 10 minutes above (later than) the freeze horizon.

The three horizons divide the set of  $(ETA_{FF})$ s into four subsets as illustrated in figure 6. Aircraft below the freeze horizon have fixed  $(STA_{FF})$ s and runway assignments. In this region controllers handle the aircraft so as to move the  $(ETA_{FF})$ s toward coincidence with the corresponding  $(STA_{FF})$ s. As aircraft approach the meter gate, occasionally a controller may find it necessary to unfreeze and then reassign and re-sequence a particular aircraft or a group of aircraft in the freeze interval. Controllers try to avoid such procedures because they generally increase workload and create complex control problems.

The scheduling process is illustrated in figures 6a and 6b for an example scenario of three aircraft  $A_i, B_j, A_{i+1}, A_{i+2}$ , and  $B_{j+1}$  in the optimization interval. Runway assignment and a sequence optimization are to be performed for the first P of these aircraft. We choose  $P = 2$ , a realistic value for a real-time scheduler. It is carried out in three steps.

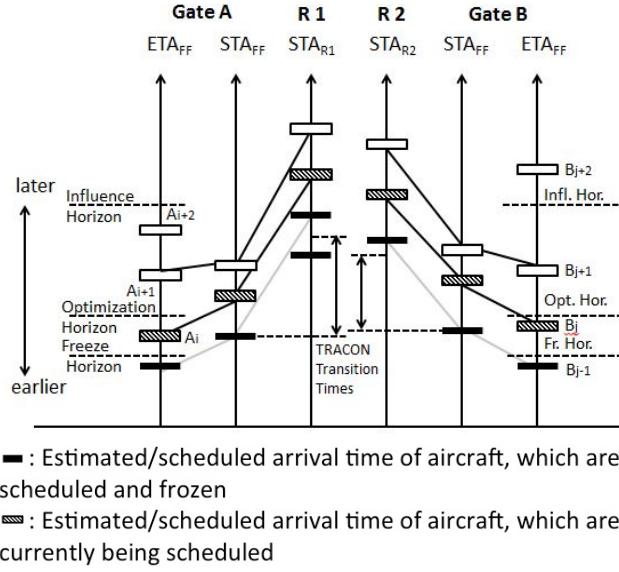
The first step generates the set of all runway assignments and scheduling orders for  $A_i$ , and  $B_j$ , producing what we will call the comparison set. Since  $A_i$  and  $B_j$  pass through different meter gates, there are no sequence-order constraints to be obeyed at the meter gates and, therefore, two scheduling orders are possible:  $A_i B_j$  and  $B_j A_i$ .

For each scheduling order all four pairs of runway assignments must be generated. For order  $A_i$  followed by  $B_j$  they are:

$$\begin{aligned} A_i \rightarrow R1, B_j \rightarrow R1 \\ A_i \rightarrow R1, B_j \rightarrow R2 \\ A_i \rightarrow R2, B_j \rightarrow R1 \\ A_i \rightarrow R2, B_j \rightarrow R2 \end{aligned}$$

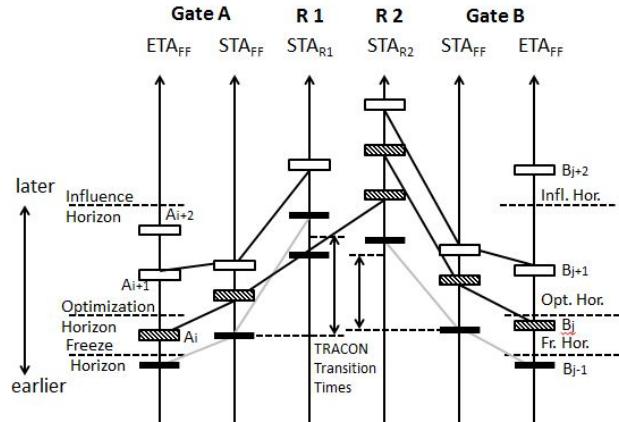
These four pairs of runway assignments, when combined with the two possible scheduling orders, produce eight trial schedule/runway-assignment options, which constitute the comparison set.

The second step of the process generates the runway  $(STA)$ s for each pair in the comparison set as well as for all other aircraft below the influence horizon. In figure 6a, these other aircraft are  $A_{i+1}, A_{i+2}$ , and  $B_{j+1}$ . They inherited their runway assignments from the initialization procedure previously described, or, if none was used, are assigned to their preferred runways. Since runway assignments are fixed for each comparison set, the basic algorithm can be applied to the determination of the runway and feeder-gate  $(STA)$ s in a straightforward way. Figures 6a and 6b illustrate two of these eight trial schedule/runway assignments. The first trial option corresponds to the FCFS order and nominal runway assignments for the two aircraft. The second trial, illustrated in figure 6b, assigns  $A_i$  to R2 but in front of  $B_j$  in the scheduling order.



Three horizons divide the set of ( $ETA_{FF}$ )s into four subsets. Two aircraft,  $A_i$  and  $B_j$ , are in the optimization interval. Initial trial schedule is shown.

**Figure 6a.-Illustration of real-time scheduling for two runways.**



**Figure 6b.-Alternative runway assignment for  $A_i$ .**

The third and final step determines the optimum runway assignment and landing order for  $A_i$  and  $B_j$  by selecting the minimum delay schedule from among the eight trial schedules of the comparison set. The delay equivalent cost,  $D$ , of each trial schedule,  $k$ , is defined as the sum of the ( $STA$ s) for all aircraft below the influence horizon:

$$D(k) = STA^k(A_i) + STA^k(B_j) + STA^k(A_{i+1}) + STA^k(B_{j+1}) + STA^k(A_{i+2}) \quad (38)$$

where in this example,  $k$  ranges from one to eight. The particular value of the index  $k$  that corresponds to the minimum of the  $D(k)$  establishes the optimum runway assignment and landing

order for  $A_i$  and  $B_j$ . When this step is completed, the scheduling status of  $A_i$  and  $B_j$  is changed to frozen. The real-time scheduler is now ready to receive a new set of updated ( $ETA_{FF}$ )s and process them in a similar manner.

#### **4.2 Estimating the Number of Trial Scheduling Combinations in the Comparison Set**

The number of distinct combinations of sequence orders and landing assignments for which trial schedules must be computed was shown in the preceding section to be 8 for the example of two landing runways and two aircraft in the optimization set. In order to assess the computational load for other cases of interest, it is useful to estimate the number of such trial schedules in the comparison set. If no limit is placed on the number of position shifts allowed, then the number of scheduling orders is  $P!$  for  $P$  aircraft in the optimization set. It should be noted that the scheduling order is the same as the landing order.

Let  $Q$  be the number of landing runways. Since each aircraft in a scheduling order of  $P$  aircraft may be independently assigned to any of the  $Q$  runways, the number of possible runway assignments for each scheduling order is  $Q^P$ . Therefore an estimate of what is essentially an upper limit of the number of trial schedules  $K$  that the scheduler must compute to determine the optimum is:

$$K = P! Q^P \quad (39)$$

Clearly,  $K$  exhibits an extremely fast growth rate. For example, if  $P$  and  $Q$  are both increased from 2 to 3,  $K$  increases from 8 to 162, which is near the limit of what a scheduler can handle in real-time at the time the original paper was published.

A current software implementation of the basic algorithm, which handles assignments to three runways, is designated for the  $P = 1$  case, and thus needs to generate only three trial schedules.

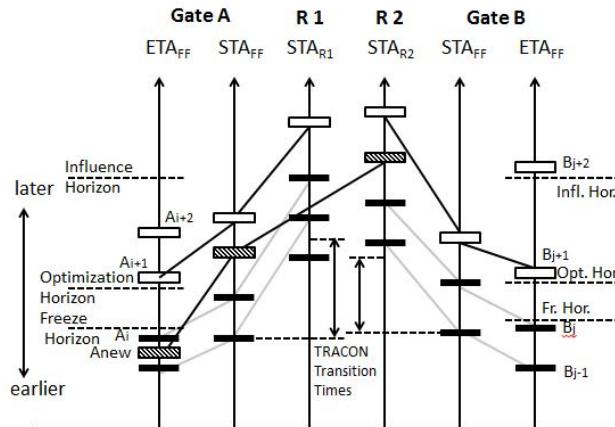
A modest improvement in scheduling efficiency can be obtained, especially for the  $P = 1$  case, by following the runway assignment of the freeze aircraft with a single position-shift trial involving this aircraft and the last-to-freeze aircraft. However, the delay reduction potential of position shifting is somewhat reduced when it follows runway-assignment optimization. This reduction occurs because runway-assignment optimization tends to assign aircraft with similar weight classes to the same runway, thus reducing the potential for delay reduction by position shifting for some situations.

#### **4.3 Adding a New Aircraft to the Schedule**

In addition to the basic algorithm that optimizes the schedules of aircraft near the freeze horizon and then transitions them from non-frozen to frozen status, the real-time scheduler also contains functions for handling a range of special scheduling events. Such events can be triggered by commands from controllers or by inputs from other components of the automation system. For example, a controller may issue a command to reschedule already frozen aircraft or reassign a group of frozen aircraft to a different runway. To handle the more complex events, for example runway configuration changes, the basic algorithm must be modified significantly. The management of these events in real-time and the creation of algorithms to generate the proper responses increase the complexity of the final software design significantly (measured by lines of computer code). Thus the software implementation of the full-function scheduler based on the algorithm described in this paper contains about 45,000 lines of C code (ref. 11).

Here we describe how the algorithm handles one such special event, namely the late arrival of a new aircraft with an  $ETA_{FF}$  that falls below the freeze horizon. A late arrival can be a recent departure from a nearby airport with short flying times to the hub airport. This event is signaled to the scheduler by the aircraft tracking- and trajectory-analysis modules of the automation system, which generate its  $ETA_{FF}$ . The scheduler must respond by merging this aircraft into the list of previously scheduled and frozen aircraft by a fair and efficient procedure.

The procedure for adding the newly born aircraft is an extension of the basic algorithm. First, the  $ETA_{FF}$  of the new aircraft is inserted into the existing  $ETA_{FF}$  list for the appropriate arrival gate in FCFS meter-gate sequence order. This procedure is illustrated in figure 6c for  $A_{new}$  arriving at gate A ahead of frozen aircraft  $A_i$ . Second, it is scheduled to the meter gate behind its lead aircraft  $A_i$  using the applicable meter gate in-trail constraint  $T_{it}$ . This step is not shown in figure 6c. Third, starting at the meter-gate time  $STA_{FFit}(A_{new})$  aircraft  $A_{new}$  is scheduled at the earliest time to a runway that is consistent with applicable meter gate-to-runway sequence constraints and, in addition, does not conflict with the schedules of any frozen aircraft. In general, the strategy we have adopted for handling late arrivals, i.e., those with initial ( $ETA_{FF}$ )s that fall below the freeze horizon, is to schedule them at the earliest time to any runway that does not require changing the ( $STA$ )s of frozen aircraft. It is important not to change frozen ( $STA_{FF}$ )s, since the corresponding aircraft may already have received their arrival/descent clearances. Changing them would increase both controller and pilot workload. Thus,  $A_{new}$  has been scheduled to R2 behind  $B_j$ , allowing  $B_j$  to retain its frozen STA and  $STA_{FF}$  as illustrated in figure 6c. Fourth, for the  $A_{new}$ 's STA, the corresponding  $STA_{FF}(A_{new})$  is determined by applying the required delay distribution. Because this  $STA_{FF}$  may conflict with a frozen  $STA_{FF}$  in more complicated cases than the one illustrated here, it may be necessary to revise the delay-distributed schedule in order to achieve the minimum in-trail constraint  $T_{it}$ . This step completes the process of scheduling the late arrival  $A_{new}$ . Note that aircraft within the influence interval were pushed out of the way to accommodate  $A_{new}$  since their schedules are allowed to change. Occasionally, events such as unexpected runway closures that necessitate unfreezing some or all frozen aircraft and then rescheduling the entire set of arrival aircraft can occur. Such events require additional modifications to the basic algorithm described here.



**Figure 6c. Illustration of real-time scheduling algorithm: Adding a new aircraft  $A_{new}$  to the scheduled list of frozen aircraft.**

In summary, during the flight history of an aircraft in Center airspace beginning with the start of active tracking and ending at the time of meter-gate crossing, the scheduler makes runway assignments for each aircraft twice. The first time is a preliminary assignment done at the start of active tracking. It ensures that every aircraft in the current schedulable list has an appropriately assigned runway. This preliminary assignment permits the scheduler to generate what might be called pseudo schedules, so named because they are updated periodically. They are used primarily to provide traffic managers with continuously updated estimates of expected delays. Controllers use these estimates, displayed in graphically convenient form, to formulate control strategies. The second time the assignment is made takes place just before the freeze horizon and involves the optimization procedure described previously. However, it should be noted that the initial assignment also influences the outcome of the optimization procedure because aircraft below the influence horizon that retain their original assignment still contribute to the value of the cost function given in equation (38).

While runway assignments are computed only twice, the ( $STA_{FF}$ )s are updated every 10-15 seconds prior to freeze. Experience with operating this scheduling algorithm in live traffic has shown that this update strategy achieves a reasonable balance between stability and responsiveness of the schedule to  $ETA_{FF}$  updates.

When an aircraft crosses a meter gate and enters TRACON airspace, it comes under the control of TRACON automation tools, such as the Final Approach Spacing Tool (FAST). At this time the aircraft is unfrozen and the FAST scheduler determines the final runway assignment and sequence order. If the traffic is being controlled accurately to the meter gates, the assignments made in TRACON airspace should conform mostly to those inherited from the Center scheduler. The next chapter examines the impact of control accuracy on the design of the scheduler in detail.

## Chapter 5

### Nomenclature

$ATA_{FF}$	=	actual time of arrival, which is a simulated arrival time given by adding a randomly generated meter-gate crossing time error $N_{pc}$ to the $STA_{FF}$ of each aircraft
$(ATA_{FF})_s$	=	more than one $ATA_{FF}$
$C_F$	=	cost factor converting fuel to direct operational cost (DOC)
$C_T$	=	cost factor converting time to DOC
DOC	=	direct operational cost, which is a cost function for optimizing delay distribution
$d_c$	=	Center delay, which is a delay time in the Center airspace
$d_T$	=	TRACON delay, which is a delay time in the terminal airspace
$d_{Topt}$	=	optimum delay distribution
$d_{Tmax}$	=	maximum value of time delay that can be handled/absorbed in TRACON airspace
$\Delta d$	=	delay increment
$ETA_{FF}$	=	estimated time of arrivals at a meter gate
$(ETA_{FF})_s$	=	more than one $ETA_{FF}$
$F$	=	additional fuel consumed in pounds because of feeder-gate crossing-time errors
$N_{pc}$	=	randomly generated meter-gate crossing time error
$STA$	=	scheduled time of arrival at runway
$(STA)_s$	=	more than one $STA$
$STA_{FF}$	=	scheduled time of arrivals at a meter gate
$(STA_{FF})_s$	=	more than one $STA_{FF}$
$T$	=	time to fly the trajectory in seconds

## 5. Strategy for Delay Absorption in the Presence of Time Control Errors

### 5.1 Delay Allocation in Center and TRACON Airspace

Whenever arrival traffic demand exceeds aircraft landing capacity over a 15-minute or longer time interval, a significant buildup of delay is likely to occur. After years of experience in dealing with such situations at large airports, traffic managers have learned how to anticipate the magnitude of a delay buildup and have devised standard procedures for absorbing the delay.

While traffic-management procedures in use today generally achieve smooth traffic flow even when delays have built up significantly, controversy lingers over what is the best procedure for delay absorption. The dividing chasm in the controversy is between pilots and airline operators on the one hand and controllers and traffic managers on the other.

Pilots and airline operators prefer delays to be absorbed close to the airport, even to the point where holding is required in the TRACON airspace at low altitude. They fear that early delay absorption far from the airport does not produce sufficient traffic pressure to achieve a high landing rate.

Traffic managers and controllers, on the other hand, contend that, on balance, it is more efficient to absorb most delay in the Center airspace far from the airport so as to maintain traffic flow in the TRACON smoothly and orderly. They further contend that delay-absorption strategies that lead to frequent holding in the TRACON airspace create high workload for controllers and risk chaotic traffic conditions that actually reduce landing rates.

We use the structure of the scheduling algorithm described in the preceding chapters, together with models of aircraft fuel consumption and accuracy of time control at the Center-TRACON boundary, to gain new insight into the delay-absorption problem. We describe a method of analysis that determines the value of delay distribution between Center and TRACON airspace such that the average direct operating cost of delay absorption for the arrival traffic is minimized.

The two factors that are the key to the analysis are aircraft fuel consumption and accuracy of time control. It is well known that the minimum fuel flow rate (lb/sec) of turbofan-powered aircraft is significantly less at cruise altitude than it is at sea-level altitude. Therefore, it is more fuel efficient to absorb delays at or near cruise altitude than it is at sea level. The performance manual of an aircraft contains the basic data needed to derive the relationship between fuel consumption and delay absorption at high and low altitude.

Such a relationship for a midsize passenger aircraft follows:

$$F = (120d_c + 180d_T) \frac{1}{60} \quad (40)$$

where  $d_c$  is the Center delay, which is assumed to be absorbed at 30,000 ft and  $d_T$  is the TRACON delay assumed to be absorbed at 3,000 ft. The quantity  $F$  is the additional fuel consumed in pounds due to delays  $d_c$  and  $d_T$  given in units of seconds. This simple relationship defines the empirical observation that fuel consumption at the high altitude is two-thirds of that at low altitude in the general case. If the total delay to be absorbed is  $d = d_c + d_T$ , then equation (40) shows that choosing  $d = d_c$ , and  $d_T = 0$  minimizes the additional fuel consumption for any delay  $d$ . It therefore follows that if the total delay to be absorbed can be determined when the aircraft is still at or near cruise altitude and a method for controlling the delay exists, the most fuel-efficient and, therefore, cost-efficient strategy is to absorb all delay in the high-altitude Center airspace and none in the low-altitude TRACON airspace.

This result leads directly to the question of how limits in the accuracy with which delays can be absorbed in Center airspace should change the proposed delay-absorption strategy.

This question is illuminated by examining the operation of the real-time scheduler. In the scheduling algorithm described in the preceding chapter, the final value of required delay absorption is determined at the time the  $STA_{FF}$  of an aircraft is frozen. Freezing occurs at the freeze horizon when an aircraft is approximately 19 minutes of flying time from its assigned meter gate. The meter gate, located at the boundary between Center and TRACON airspace, therefore, provides the dividing point for distributing the total delay between Center and TRACON airspace. And the delay distribution function, DDF, which is imbedded in the architecture of the basic real-time algorithm, provides the mechanism for allocating the delay between these airspaces on an aircraft-by-aircraft basis.

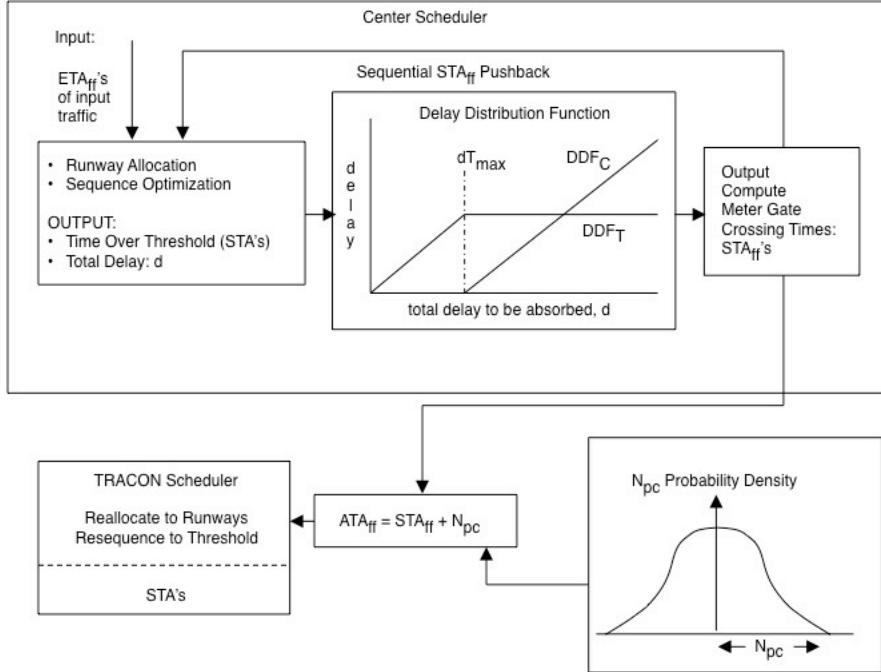
Thus, the basic information needed to study the question posed previously is to determine the expected accuracy of controlling an aircraft to cross the meter gate at time  $STA_{FF}$  when the aircraft is initially 19 minutes away from the meter gate.

Accuracy of control for both 19 and 30 minutes of flying time to the meter gates was recently estimated by analyzing over 3000 actual flights that landed at the Dallas/Fort Worth Airport (ref. 13). The estimates of accuracy were determined for both the legacy metering system used at the Fort Worth Center and the Center/TRACON Automation System, which was scheduled for field tests at the Fort Worth Center. Of particular interest here is a comparison of standard-deviation

errors between the current metering system and TMA/CTAS at 19 minutes to the meter gates. They are, respectively, 180 and 90 seconds (ref. 13). Before developing a quantitative method for studying the effects of these errors, it is important to understand qualitatively why these errors have the potential to increase delays. The errors can contribute to slot losses at the runway. This effect is most easily visualized when traffic is dense and all delay is being absorbed in the Center airspace. If an aircraft crosses the meter gate later than its prescribed  $STA_{FF}$ , all aircraft scheduled closely behind the late aircraft at the minimum time separation will have this time error also passed on to them. Since the time to fly from the meter gate to the runway is, by assumption, equal to the minimum time, it is generally difficult if not impossible to recover this slot loss completely by speeding up or short-cutting the path. Moreover, similar to the runway-assignment problem previously described, the total delay increment due to a fractional slot loss can be several times the magnitude of the original time error if the error is propagated to several trailing aircraft. Thus, the assumed benefits in fuel efficiency of absorbing all delays at the more fuel-efficient altitudes of Center airspace are diminished by increased delays and greater fuel consumption due to those meter gate crossing time errors.

## **5.2 Stochastic Simulation of Meter Gate Crossing Errors**

The effect of meter gate crossing errors was studied quantitatively by stochastic Monte Carlo simulation in (ref. 12). A simplified diagrammatic representation of this simulation is shown in figure 7. The upper part of the figure represents the basic scheduling algorithm. The diagram draws attention to the two distinguishing characteristics of the algorithm, namely the delay distribution function for allocating delays and the feedback-like effect of this function through the sequential pushback of the ( $STA_{FF}$ )s. The input to the algorithm is a set of ( $ETA_{FF}$ )s representing the simulated traffic scenario. They are generated by a random process that has been designed to match the statistical characteristics of a typical 90-minute traffic rush at the Dallas/Fort Worth airport. The simulation drives the algorithm with several thousand samples of such traffic rushes, all different from each other yet statistically identical. The performance of the algorithm is measured by calculating delay and fuel-consumption averages for thousands of such rush-traffic samples. Although the input traffic is statistically generated, this part of the simulation produces a deterministic set of ( $STA_{FF}$ )s and ( $STA$ )s for each randomly generated set of ( $ETA_{FF}$ )s.



**Figure 7.-Stochastic Monte Carlo simulation of scheduler.**

The lower part of figure 7 represents the stochastic simulation of meter gate crossing time errors. The simulation generates an actual time of arrival,  $ATA_{ff}$ , over a meter gate for each aircraft by adding a randomly generated meter gate crossing time error  $N_{pc}$  to the  $STA_{ff}$  of each aircraft, the latter being provided by the simulation of the basic scheduling algorithm. The statistical properties of  $N_{pc}$  are chosen to match the empirically determined probability distribution of meter gate crossing errors.

Although the errors were found to be nearly normally distributed, they are approximated here by the convolution sum of three uniformly distributed random variables having the general shape shown in the figure. This approximation eliminates the unrealistic large outliers of a normal distribution. The ( $ATA_{ff}$ )s now provide the input to what is referred to in the figure as the TRACON scheduler. This scheduler is identical to the basic scheduler but with  $d_{Tmax}$  set to zero. By reassigning and re-sequencing aircraft at the time they actually cross the meter gates, the TRACON scheduler compensates, to the degree that is possible, for the adverse effects of the meter gate crossing errors. Moreover, the twice-repeated application of the sequencing and runway-assignment algorithm, first at the Center freeze horizon and then at the TRACON boundary, more accurately reflects actual operations.

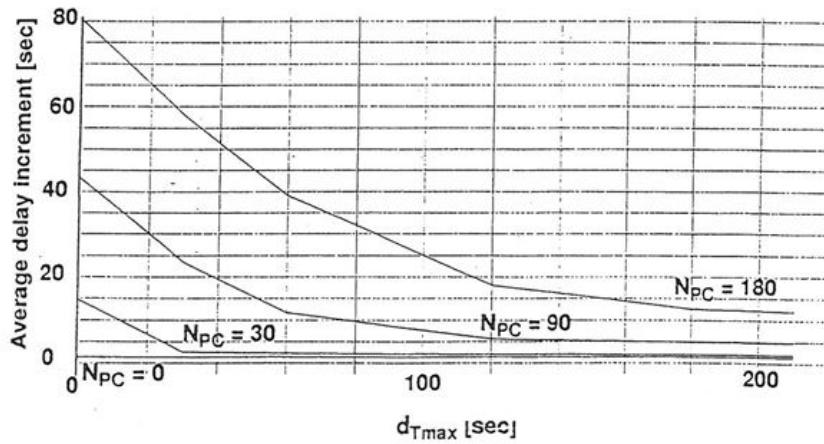
The two parts of the simulation, where the output of the first becomes the input to the second, generate runway threshold (STA)s whose values reflect both the efficiencies gained by sequencing and runway-assignment optimization and the time-control errors in meter gate crossing times.

### 5.3 Analysis of Results

The stochastic Monte Carlo simulation tool briefly described in the preceding section will now be used to investigate the quantitative relationship between delay distribution strategies, meter gate time control errors, and scheduling efficiency.

These relationships will be presented here for the single-runway case. This case is not only important in its own right, but it also reveals the essential characteristics of these relationships more clearly than the multi-runway case would. The route structure modeled in the simulation consists of four meter gates with two independent traffic streams converging on each gate. One stream contains a mix of large and heavy jets, the other only large turboprops. The streams are independent by virtue of a required large altitude separation between them at the crossing point. Independence implies that there are no in-trail separation restrictions between aircraft in different streams converging on the same gate. The input traffic rate is 36 aircraft/hour, which is slightly above the maximum sustainable traffic level. There will thus be a significant buildup of delays at this traffic level. All data points used in plotting of curves represent *the average of* over 1000 randomly generated traffic samples, each of which contains 54 aircraft in a 90-minute rush period, or 36 aircraft/hour.

The results, plotted in figures 8–10, focus exclusively on the effects of meter gate crossing errors. The first of the figures, figure 8, plots the delay increment  $\Delta d$  as a function of the TRACON delay distribution variable,  $d_{Tmax}$ , with meter gate crossing errors,  $N_{pc}$ , as a parameter. It is seen that the origin of coordinates corresponds to  $\Delta d = 0$ ,  $d_{Tmax} = 0$ , and  $N_{pc} = 0$ . The average delay obtained for the simulated traffic scenario at these operating conditions was 280 seconds.



**Figure 8.-Average delay increment for 36 aircraft/hour for a single runway vs.  $d_{Tmax}$ .**

For each of the three non-zero values of  $N_{pc}$ , the delay increment  $\Delta d$ , decreased strongly with increasing values of  $d_{Tmax}$ . For the highest value of  $N_{pc}$ , 180 seconds, which corresponds to the crossing errors of the current operational systems, the reduction in the delay increment is especially striking, declining from 80 seconds at  $d_{Tmax} = 0$  to only 11 seconds at  $d_{Tmax} = 180$  seconds. This result confirms the ability of TRACON delay distribution to compensate almost completely for slot loss due to meter gate time control errors. At the two lower values of  $N_{pc}$ , the delay increments are less to start with and decline to correspondingly lower values as  $d_{Tmax}$  is increased. The  $N_{pc} = 30$  seconds case establishes the practical lower limit of errors, which could be achieved with advanced controller tools such as Descent Advisor and/or improved on-board flight management systems. The middle value of  $N_{pc} = 90$  seconds can be achieved with the Traffic Management Advisor. At  $N_{pc} = 0$ , delay distribution has no effect on delay increment, as expected.

The asymptotic limits of this family of curves suggest a simple rule of thumb for choosing the optimum delay distribution: choose  $d_{Tmax}$  equal to  $N_{pc}$ . There is, however, a practical upper limit

on  $d_{Tmax}$  of about 180 seconds. It is determined by the maximum delay that can reasonably be absorbed in TRACON airspace.

A significant difference in the effect of TRACON delay distribution exists between the single- and multi-runway cases. In the multi-runway case, a non-zero  $d_{Tmax}$  can be used to reduce delays even for  $N_{pc} = 0$ . Analysis of this case shows that delay distribution in the TRACON, combined with optimization of runway assignments, can mitigate delays due to meter-gate in-trail constraints and/or slot losses, thus reducing overall delays.

A substantially different picture emerges from figure 9, which plots the increment in fuel consumption  $\Delta F$  as a function of the same two variables as in figure 8. The incremental fuel consumption at  $d_{Tmax} = 0$  and  $N_{pc} = 180$  seconds is 230 pounds for the average aircraft in the traffic sample. This amount represents a significant economic penalty in fuel consumption resulting directly from time errors at the meter gates. Initially, the fuel consumption strongly declines as  $d_{Tmax}$  increases. However, the distinguishing feature of the curves is that they reach a clearly defined minimum with respect to the variable  $d_{Tmax}$ . Beyond the minimizing value of  $d_{Tmax}$  the fuel consumption begins to rise again and becomes asymptotic to the  $N_{pc} = 0$  curve. In this case, the rule of thumb for choosing the fuel-optimum value of  $d_{Tmax}$  is  $2/3 N_{pc}$ . This result reflects the influence of the fuel-consumption trade-off relation, equation (40). It shows that high values of TRACON delay distribution exact a fuel-cost penalty that weighs against the benefits of incremental delay reduction shown in figure 8.

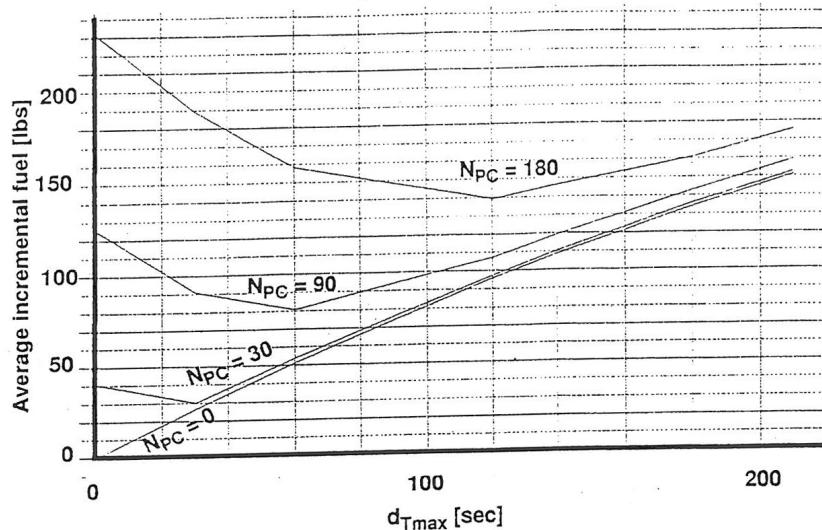


Figure 9.-Incremental fuel vs.  $d_{Tmax}$ .

#### 5.4 Unification Principle

In the preceding section we saw that minimizing delay requires a substantially different delay distribution than that required to minimize fuel consumption. The unification principle introduced in this section is an attempt to reconcile this difference by using direct operating cost (DOC) as a cost function for optimizing delay distribution. DOC plays an important role in the economic analysis of airline operations. It is defined as the sum of the cost of time and the cost of fuel as follows:

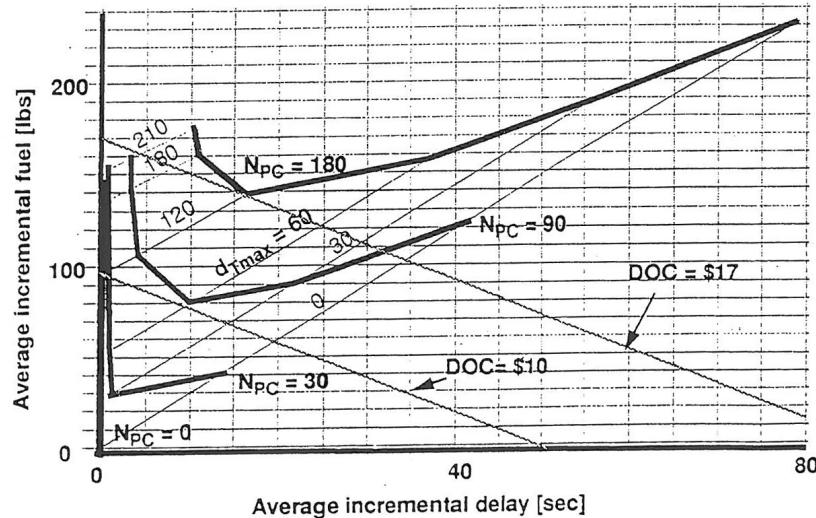
$$DOC = TC_T + FC_F \quad (41)$$

where  $T$  is the time to fly a trajectory in seconds,  $F$  is the fuel consumption of a trajectory in pounds, and  $C_T$  and  $C_F$  are cost factors for converting time and fuel to  $DOC$  measured in dollars. Airline operations analysts can provide representative values for the cost factors  $C_T$  and  $C_F$ , applicable to the average aircraft in an airline fleet. These cost factors may differ between airlines and may also change with fuel costs. Representative values for these factors were obtained from a U.S. airline for a typical aircraft in its fleet. From these values the following relationship was derived:

$$F = 10DOC - 2T \quad (42)$$

Here we have chosen  $F$  as the dependent variable,  $T$  as the independent variable, and  $DOC$  as a parameter. This representation is most convenient for the analysis to follow.

To prepare for the application of equation (42), figures 8 and 9 have been combined in a two-parameter family of curves sometimes referred to as a carpet plot. In this carpet plot, figure 10, fuel and time increments may both be considered dependent variables plotted along vertical and horizontal axes, respectively. The independent variables are the parameters  $d_{Tmax}$  and  $N_{pc}$ .



**Figure 10.-DOC increment curves on the time-fuel coordinates.**

The unification principle may now be defined as the process by which the  $DOC$  relationship given by equation (42) is used to select the delay distribution that minimizes the increment in  $DOC$  for the average aircraft during the rush-traffic period. Note that since equation (42) is linear in all variables, incremental variables can directly replace the original variables in equation (42) without changing its form.

The selection process can be explained by superimposing the  $DOC$  increment curves derived from equation (42) on the time-fuel coordinates of figure 10. Then it becomes evident that the unification principle is satisfied at the point of tangency of a linear  $DOC$  curve with a specific  $N_{pc}$  curve. The  $DOC$  line that is tangent to the curve of a selected value of  $N_{pc}$  gives the lowest possible  $DOC$  increment corresponding to that value of  $N_{pc}$ . It therefore gives the optimum operating point for the selected value of  $N_{pc}$ . The final step is to select the delay distribution parameter,  $d_{Tmax}$ , corresponding to the optimum operating point. That is done by interpolating on values of constant  $d_{Tmax}$  curves to find a curve that passes through the optimum operating point.

That value of  $d_{T_{max}}$  establishes the optimum delay distribution,  $d_{Topl}$ . For the case of  $N_{pc} = 180$  sec,  $d_{Topl} = 120$  sec and for  $N_{cp} = 90$  sec,  $d_{Topl} = 60$  sec.

The difference in the incremental *DOC* between any two values of meter gate time errors has an economic interpretation. It represents the cost penalty of operating an arrival management system at the higher meter gate time error compared to operating it at the lower value. Conversely, this difference also gives the average cost saving per landing that would be obtained by implementing a new technology that reduces the meter gate time errors by a specified amount. For example, by reducing the time error from the current value of 180 seconds to 30 seconds, attainable with an advanced controller tool such as Descent Advisor, each landing aircraft would, on average, reduce its operating cost by \$14.

# **Chapter 6**

## **6. Conclusion**

The algorithm described here uses a generalized first-come-first-served (FCFS) scheduling protocol that incorporates important operational constraints on sequencing and is computationally fast enough to be implemented in a real-time environment. The paper also describes how to improve the performance of the basic algorithm by searching for more optimized landing schedules and runway assignments that reduce overall delays while still meeting computational constraints for real-time operation. The algorithm provides the conceptual and computational foundation for the Traffic Management Advisor, an operational tool used by controllers to manage arrival traffic at major airports. The algorithm uses a two-stage process to generate efficient schedules and runway assignments.

In the first stage it generates landing times and runway assignments that produce the earliest landing times subject to required minimum time spacings between aircraft that were scheduled at earlier times. This stage yields the total delay that each new arrival aircraft must absorb prior to landing. In the second stage, aircraft are assigned times to cross a feeder fix (also known as a meter gate) using a delay distribution function, which specifies how much of the total delay aircraft must absorb in Center airspace prior to crossing the feeder fix, and how much they must absorb in TRACON airspace prior to landing. Thus, the dual-stage scheduling procedure of the algorithm generates coordinated schedules for all aircraft at both feeder fixes and runways.

A stochastic simulation was developed to model the effect of time errors of aircraft when crossing the feeder fix. The simulation explored the tradeoffs between crossing-time errors, delay distribution parameters, and performance measures, such as average delay, fuel consumption, and direct operating costs. Analysis of these tradeoffs provided a method for selecting the delay distribution function so as to minimize a cost for specified values of crossing-time errors. It was found that larger crossing-time errors increase costs and require more delay to be absorbed in TRACON airspace as a countermeasure against potential slot losses resulting from these errors. The analysis method thus gives a quantitative measure for aircraft operating cost savings that result from improvements in trajectory tracking performance.

## References

- [1] Erzberger, Heinz; and Tobias, Leonard: A Time Based Concept for Terminal-Area Traffic Management. In "Efficient Conduct of Individual Flights and Air Traffic," pp. 52-1–52-14, AGARD CD-410, June 1986.
- [2] Slattery, R.A.: Terminal Area Trajectory Synthesis for Air Traffic Control Automation. Proc. of Am. Control Conf., June 1995.
- [3] Erzberger, Heinz: Design Principles and Algorithms for Automated Air Traffic Management. In AGARD Lecture Series 200: Knowledge-Based Functions in Aerospace Systems, Madrid, Paris, San Francisco, Nov. 1995.
- [4] Erzberger, Heinz; and Nedell, William: Design of Automated System for Management of Arrival Traffic. NASA TM 102201, June 1989; Engle, Laurie: Conflict Detection Tool, Addendum to TM 102201, NASA Ames Research Center, Oct. 1989.
- [5] Erzberger, H.; Davis, T.J.; and Green, S.M.: Design of Center-TRACON Automation System. Proc. of AGARD Guidance and Control Panel 56th Symposium on Machine Intelligence in Air Traffic Management, Berlin, Germany, 1993, pp. 52-1–52-14.
- [6] Davis, T.J.; Erzberger, H.; Green, S.M.; and Nedell, W.: Design and Evaluation of an Air Traffic Control Final Approach Spacing Tool. J. Guidance, Control, and Dynamics, vol. 14, no. 4, July-August, 1991, pp. 848–854.
- [7] Green, S.M.; Vivona, R.A.; and Sanford, B.: Descent Advisor Preliminary Field Test. Proc. AIAA Guidance, Navigation, and Control Conf., Baltimore, Md., Aug. 7–9, 1995.
- [8] Davis, T.J.; Krzeczkowski, K.J.; and Bergh, C.C.: The Final Approach Spacing Tool. Proc. 13th IFAC Symposium on Automatic Control in Aerospace, Palo Alto, Ca., Sept. 1994.
- [9] Swenson, H. N.; Hoang, T.; Engelland, S.; Vincent, D.; Sanders, T.; Sanford, B.; and Heere, K.: Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center. 1<sup>st</sup> USA/Europe Air Traffic Management Research & Development Seminar, Saclay, France, June 17–19, 1997.
- [10] Krzeczkowski, K.J.; Davis, T.J.; Erzberger, H.; Lev-Ram, I.; and Bergh, C.C.: Knowledge-Based Scheduling of Arrival Aircraft in the Terminal Area. Proc. AIAA Guidance, Navigation, and Control Conf., Baltimore, Md., Aug. 1995.
- [11] Neuman, Frank; and Erzberger, Heinz: Analysis of Delay Reducing and Fuel Saving Sequencing and Spacing Algorithms for Arrival Traffic. NASA TM 103880, Oct. 1991, NASA Ames Research Center.
- [12] Wong, Gregory L: The Dynamic Planner: The Sequencer, Scheduler, and Runway Allocator for Air Traffic Control Automation. NASA TM 2000-209586, Apr. 2000.
- [13] Ballin, M.G.; and Erzberger, H.: An Analysis of Aircraft Landing Rates and Separations at Dallas/Ft. Worth Airport. NASA TM 110397, July 1996.