UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING, SCIENCE & MATHEMATICS

School of Mathematics

Aircraft Arrival Management

by

Adam Robert Brentnall

Thesis for the Degree of Doctor of Philosophy

January 2006

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE & MATHEMATICS

SCHOOL OF MATHEMATICS

Doctor of Philosophy

AIRCRAFT ARRIVAL MANAGEMENT

by Adam Robert Brentnall

This Thesis is based around the Air Traffic Control Arrival Management problem of scheduling the landing of aircraft on runways, where aircraft must respect minimum separation distances based on wake-vortex criteria. Existing scheduling approaches and methods of assessing their effects on Air Traffic Control are reviewed. Several polynomial-time dynamic programming algorithms are proposed for determining optimal landing sequences. Six sequencing algorithms and four delay-sharing strategies are linked into a discrete-event simulation model of Stockholm Arlanda arrival airspace. The procedures for generating traffic samples, and important output performance indicators, are validated against 16 recorded traffic samples of arrivals from autumn 2003 through hypothesis tests, confidence intervals and tests of dynamic behaviour. Several statistical methods are used to analyse experiment output from the Stockholm Arlanda model. These include graphical methods, EDFIT analysis, regression metamodels, variance metamodels and logit models. A series of detailed experiments on the model do not find tremendous benefits to Air Traffic Control airport runway capacity from advanced sequencing, above the benefits that occur from using first-come first-serve sequences. However, changes to the Air Traffic Control system are found in holding time, time in approach sectors and stability of the advice generated.

Contents

List of Tables

List of Figures

Acknowledgements

I have been guided throughout this work by my supervisor at the University of Southampton, Professor Russell C.H. Cheng, and Mr. Alan Drew of the Eurocontrol Experimental Centre in Bretigny, France. Russell has directed my attention to various operational research skills to help react to the Air Traffic Control knowledge of Alan. I am very grateful to them both. Many thanks to my advisor, Professor Chris Potts, for helping with the formulation of the dynamic programs described in Chapter 2. I thank Peter Martin, program manager of EVP (European reference ATM Validation Platform) at Eurocontrol, for arranging my studentship and facilitating the research. I also thank the Swedish Civil Aviation Administration LFV (Luftfartsverket) for their hospitality, and supplying some data relating to Stockholm Arlanda airport. I acknowledge the financial support of the European Commission and the EPSRC. The head of the graduate school in mathematics, Dr. David Chillingworth, provided guidance with some grammatical issues. I would like to acknowledge the friendship and support of some fellow students: Paul Gardner, James Dartnall, Marta Cabo Nodar, Edgar Possani, Christine Currie, Muge Saadetoglu, Naomi Powell, Ebert Brea and Israel Vieira. My love and thanks to Estelle, for turning up during Chapter 1, marrying me between Chapters 3 and 4, and supporting me throughout.

List of abbreviations

ACC	Area Control Centre
AFTN	Aeronautical Fixed Telecommunications Network
AMAN	Arrival Manager
ANOVA	Analysis Of Variance
ATC	Air Traffic Control
ATM	Air Traffic Management
CAA	Civil Aviation Authority
CDF	Cumulative Distribution Function
CFMU	Central Flow Management Unit
COMPAS	Computer Oriented Metering Planning and Advisory Program
CPS	Constrained Position Shifting
CTAS	Center TRACON Automation System
DES	Discrete Event Simulation
DP	Dynamic Program
EDD	Earliest Due Date
ERD	Earliest Release Date
EDF	Empirical Distribution Function
EDFIT	Empirical Distribution Integral Test
EEC	Eurocontrol Experimental Centre
EST	Estimate data
FAST	Final Approach Spacing Tool
FCFS	First-Come First-Serve
FIFO	First-In First-Out
FPL	Flight Plan data
GOF	Goodness Of Fit
IAF	Intermediate Approach Fix
ICAO	International Civil Aviation Organization
ID	Identification
MAESTRO	Means to Air Expedition and Sequencing of Traffic with Research of Optimization
MPS	Maximum Position Shift
NATS	National Air Traffic Services
OLS	Ordinary Least Squares
OSYRIS	Orthogon System for Real-time Inbound Sequencing
RHP	Rolling Horizon Procedure
ROPAC	Route-Oriented Planning and Control
RW	Runway
SD	Standard Deviation

TMA	Terminal Area
TMA	Traffic Management Advisor (in CTAS context)
Track	Radar Track data
TRACON	Terminal Radar Approach Control
USA	United States of America
WLS	Weighted Least Squares
WV	Wake Vortex

Chapter 1

Introduction

Air Traffic Control (ATC) Arrival Manager (AMAN) computer-driven decision support tools have been demonstrated to increase airport runway capacity at a number of locations (Eurocontrol 2000*a*). They work by advising controllers on landing sequences and relevant control actions, for aircraft arrivals up to a certain distance away from an airport. Some landing sequences may produce less delay, or land more aircraft per unit time than First-Come First Serve (FCFS) because separation distances between aircraft depend on aircraft wake-vortex categories. Arrival aircraft sequencing based on these minimum separations has been little tested, and Eurocontrol (2000*a*) (recommendation R-2) recommends that further work be undertaken to study the benefits and feasibility of sequencing algorithms that attempt to optimize the minimum separations between aircraft. The Eurocontrol Experimental Centre (EEC), established in 1963 to conduct research and development in Air Traffic Control, has a project underway to investigate use of an AMAN with this functionality. Their investigation is being made through real-time simulation experiments with air traffic controllers on an experimental AMAN system with one sequencing algorithm. This Thesis investigates the effects of alternative sequencing algorithms and related delay-sharing strategies. A caveat applies to all findings presented: they are based on data used in the modelling process, not on a detailed operational study.

A list of the abbreviations used in this work may be found in the section preceding this introduction. The Aircraft Arrival Management problem is reviewed and motivated in Chapter 2. Formal definitions are made of the sequencing and delay-sharing problems based on the aircraft arrivals situation. Some systems that have already been developed in Europe and USA are described. Sequencing techniques that have not been used in operational AMAN systems are also reviewed. Some of these algorithms have been tested using models of the aircraft arrival process, in the Terminal Area (TMA) surrounding an airport.

Advanced sequencing techniques have been found to increase airport capacity dramatically in a number of the models. However, it is the contention of this Thesis that these models have not paid enough heed to validation or variation, and so some doubt is cast on their conclusions. Specific issues with some published work are highlighted.

Chapter 3 considers the problem of scheduling the landing of aircraft on runways, where aircraft must respect minimum separation distances based on their weight. Several polynomial-time dynamic programming algorithms are proposed for determining optimal landing sequences. Three different machine job models are made. In the first, dynamic programs are developed to sequence aircraft out of holds onto several runways for any regular objective function. In the second, dynamic programs are developed to sequence aircraft onto a single runway based on their release dates, to minimize makespan and total tardiness (assuming each job's release and due date are the same). In the final model, dynamic programs are developed to sequence aircraft based on their approach stream FCFS order and release dates onto several runways for any regular objective function. In all the models deadline constraints may be incorporated, but the second approach requires that deadline constraints vary linearly with corresponding release dates. The chapter concludes by describing how the Constrained Position Shifting (CPS) constraints of Psaraftis (1980), where aircraft may not be moved more than M positions either side of their FCFS position, may be incorporated into all the dynamic programs.

A Visual Basic, discrete-event, terminating simulation model was developed and is described in Chapter 4. This model of pre-TMA airspace is used to investigate scheduling and delay-sharing strategies when landing aircraft at airports. Analysis is undertaken on Stockholm airport but may also be carried out in future on alternative airspace. The conceptual model sub-systems, assumptions, inputs and outputs are all described. Six sequencing algorithms are implemented in the model; three dynamic programs from Chapter 3, two FCFS rules and a heuristic that represents a potential algorithm for an operational AMAN system. Four delay-share strategies are implemented; all delay in hold, delay as late as possible, delay as early as possible and delay evenly throughout the route. Not all the strategies are compatible with all sequencing algorithms, a total of 18 combinations may be run. Five implemented re-sequence strategies are listed.

Statistical validation procedures are used in Chapter 5 to lend credibility to the simulation's results. The simulation model is set up using data to represent

Stockholm Arlanda from a database with 28 different days of Eurocontrol Central Flow Management Unit (CFMU) historical flight plans, for arrivals into Stockholm Arlanda in 2001. The simulation model is compared to 16 traffic samples, recorded on aircraft arriving at Stockholm Arlanda airport in autumn 2003. Hypothesis tests are carried out to compare the statistical models that generate aircraft arrivals, with the real data. Hypotheses of no difference cannot be rejected (individually) at the 0.05 level. Model outputs delay and landing rate are also examined. Mean positive delay from the real data, and the model are compared using goodness-of-fit tests. No significant differences are found at the 0.05 level. A 95% bootstrap confidence interval on the mean difference (Davison & Hinkley 1997) between the two mean positive delays also covers zero, at [-0.159, 0.319] minutes. Landing rate is a time-dependent performance indicator, so a slightly unusual hypothesis test is used to validate, based on graphical analysis. This test can not be rejected at the 0.05 level. A sensitivity analysis of landing rate is carried out by fitting an Ordinary Least Squares (OLS) regression model (Kleijnen 1995) to predict actual landing rate, using the simulation landing rate. The model landing rate coefficient returns significant with a 95% confidence interval covering 1. Dynamic behaviour of landing rate is investigated using some subjective graphical time series methods. No difference is noticeable. Overall validation of the simulation model is based on Fishers composite test (D'Agostino & Stephens 1986). This can not be rejected at the 0.05 level. The chapter concludes by summing up the strengths and weaknesses of the analysis.

Chapter 6 reviews the statistical methodology used in the experiments of Chapters 7 and 8. Three main factors are investigated in this terminating simulation model: sequencing algorithm, delay-sharing strategy and traffic description. Run time is largely determined by the sequencing algorithm, but is short, often less than a minute. Analysis methods that make use of a large number of replicates at each design point are thus applicable, and some are reviewed. Ideas underlying the application of Design of Experiment methodology to a computer simulation model of this sort are outlined. The Empirical Distribution Integral Test (EDFIT) method (Cheng & Jones 2004) is used to analyse trends in output distribution revealed by graphical analysis. Significance levels of factors are found through Monte-Carlo simulation, in the form of EDFIT tables. These tables are developed for the general case of an unbalanced design, in a similar manner to Analysis of Variance (ANOVA) tables. Statistical models are used to better understand the impact of input factors on distribution summaries. Linear regression models for means may be fitted by a number of methods, but a large number of repeats per design point makes Weighted Least Squares (WLS) with

weights estimated from the output more attractive than is traditional. Similarly, variance models may be fitted using design point sample means, as the relative loss in efficiency from this approach is small. Models for skewness, or any other distribution summary, are also proposed. All the models may use resampling routines, based on the large number of design point repeats, to test for significance when standard asymptotic result assumptions do not hold. Finally, logit and beta models are outlined to further examine a single observation per simulation run, by recording the proportion of data points that miss thresholds or targets.

The series of experiments reported in Chapter 7 are run to see what effect change to sequencing algorithms and delay-share strategies may have on arrivals to Stockholm Arlanda airport. The experiments become progressively more complex, as new factors are introduced. Under the range of traffic conditions simulated, nothing is gained through improved sequencing algorithms for delay and landing rate performance indicators: sequencing FCFS at runway performs as well as any other. However, system behaviour is found to vary by sequencing algorithm and delay-sharing strategy. Holding time and its variability is reduced by delaying aircraft before the TMA. As traffic intensity increases, the gain in reducing the mean time holding increases, but the gain in reduction in variability of hold time decreases. Delaying aircraft pre-hold results in more traffic for controllers in sectors further back from the airport, even when delaying as late as possible without holding. This may have implications on other Air Traffic Control issues, such as slot allocation for aircraft departing at different airports that need to fly in the sectors affected. Delay-to-lose advice through time is found to be more stable when delaying aircraft in holds, than earlier in airspace. The CPS constraint is shown to be a good method to limit the variability of advice from advanced sequencing algorithms, to that found from a FCFS at runway algorithm. Three general conclusions are drawn from the experimental results. First, improved sequencing techniques should not be regarded as a panacea to reduce delay and increase landing rate because the ability to realize these benefits depends on arrival airspace and traffic characteristics. Second, different sequencing algorithm and delay-share strategies in an AMAN system may cause different system behaviour. Last, choice of sequencing algorithm and delay-share strategy will affect stability of advice to controllers, and quality of information to other users.

Chapter 8 investigates the effect an Arrival Manager might have on airport runway capacity, by looking at delay and landing rate performance indicators. The wake-vortex category traffic mix range examined in Chapter 7 is not too large because only small, realistic changes to the validated mix are made. This chapter

investigates a much wider range to look for benefits to airport capacity when the wake-vortex mix is more varied. No clear reduction in mean delay is found when sequencing aircraft FCFS in comparison with alternative methods, under the traffic conditions simulated. However, the distribution of aircraft delay is found to change using the statistical methods based on proportions. One algorithm delays a greater proportion of aircraft above a threshold of 15 minutes, two others delay more aircraft between 13 and 15 minutes. Landing rate is found to be affected only when traffic is saturated with sufficient wake-vortex mix, to a point where algorithms have enough choice of sequence position to make a difference. This does not happen in the simulation model when arrival rates are around airport runway capacity. The dynamic nature of sequence updating is shown to produce situations where an optimal deterministic algorithm may produce sub-optimal sequences, and be bettered by a heuristic. In situations when it is possible to increase landing rate over FCFS, the addition of a CPS constraint, that aircraft may only be sequenced a maximum of 3 positions either side of their FCFS position, is found to make increase in landing rate impossible at some wake-vortex mix levels. In this case, aircraft are shown to bunch together locally as a difference is found in maximum landing rate. Overall, no obvious increase in the airport runway capacity of the simulation model airspace surrounding Stockholm Arlanda is found. System behaviour changes when landing sequence is altered, but there is not enough evidence to support a claim that the simulation airspace model may better cope with more arrivals than it would when sequencing aircraft FCFS.

Chapter 2

The Air Traffic Control Arrival Management problem

Question Which aircraft should land next?

The Air Traffic Control (ATC) Arrival Management problem is summed up by this question. Section 2.1 describes why the question is of particular importance today and introduces the problem in more detail. The rest of the chapter is geared towards understanding the utility that modelling and analysis of the problem may bring. The aircraft arrivals situation is described in Section 2.2 and models of the arrival sequencing and delay sharing sub-problems are defined in Sections 2.3 and 2.4. Some real Air Traffic Control Arrival Manager (AMAN) systems have already been developed. Section 2.5 reports how they have worked and reasons why they were built. The theoretical sequencing problem is well studied and many approaches are reported in Section 2.6. Assessment of how well these algorithms may perform in the dynamic Air Traffic Control environment has been less well studied, but some of the findings from various approaches to this are discussed in Section 2.7. The chapter concludes in Section 2.8 by stating how this work will fit into preexisting work.

2.1 Introduction

The primary purpose of Air Traffic Control is to ensure that aircraft fly to their destination in a safe, orderly and expeditious manner. In 2003, European aircraft traffic increased 2.8% over the previous year (Performance Review Unit 2004) and it is forecast to increase between 2% and 5% in 2004 (STATFOR 2004). Air Traffic Control must make the most use of its existing facilities, while maintaining very high safety levels, in order to prepare for increases in demand. Gilbo (1993) and Gilbo (1997) develop mathematical models of airport runway capacity because

they see it as the bottleneck in Air Traffic Control capacity. Many others hold this view, including most of the authors referenced in Section 2.6 such as Fahle & Wong (2003). Methods of utilizing existing airport capacity are especially important because airport capacity is a politically and environmentally sensitive issue (BBC News 2004).

Air Traffic Control has always adapted in order to maintain a safe and efficient service. One way it has coped with the increase in demand has been through the use of new technology. Technologies such as the Radio Telephone (RT), Radar, Distance Measuring Equipment (DME) and Instrument Landing Systems (ILS) that are used today enable more aircraft to fly safely than was once thought possible (Graves 1998). Arrival Manager tools are also thought to have the potential to aid ATC with increased demands. These computer-driven support tools advise controllers on a landing sequence and consequent control actions for aircraft up to a certain distance away from an airport (Eurocontrol 2000*a*). One aim of such a tool is to improve use of system capacity at the airport. Others include potential improvements in aircraft delay and punctuality, reducing fuel consumption, helping controller workload and providing information to other users of the system.

Generating sequence and control advice might work towards these aims for a number of reasons. Traditionally controllers have sequenced arrivals First-Come First-Serve (FCFS), and this is how the majority of controllers still work (Graves 1998, Bianco, Dell'Olmo & Giordani 1999, Carr, Erzberger & Neuman 2000). An AMAN might be set up to mimic this sequence process. In displaying the FCFS sequence and the recommended control actions needed to maintain minimum separations between aircraft, the amount of information available to a controller increases. This could lead to improvements (Barco-Orthogon 2002). Additionally, improvements may be found if the advised landing sequence is optimized. Sequence dependent minimum separations based on aircraft wake-vortex category mean that some landing sequences may produce less delay, or land more aircraft per unit time than FCFS (see Section 2.2).

The Eurocontrol Experimental Centre (EEC) investigates potential methods of increasing capacity. They currently have a project underway examining the effects of using an AMAN tool on ATC. The interest is in exploring use of such a tool with controllers, to see if some of the aims described have a basis in reality. The project is being carried using an experimental AMAN system, OSYRIS (Barco-Orthogon 2002). This work fits into the broad scheme of the EEC project.

A high-level goal is to examine whether the aims of an AMAN are achievable and to quantify any improvements or side-effects that may result from such a system.

2.2 Arrivals situation

When a pilot arrives in the Terminal Area (TMA) of an airport they have taken their aircraft from their destination over a sequence of navigating beacons, communicated with air traffic controllers about their route as they passed through ATC sectors and now they want to land (Graves 1998). A schematic of this process may be found in Figure 4.3. An immediate question for the controllers in the TMA is what order aircraft should land. Factors influencing this decision may include safety considerations based on knowledge of the current traffic situation in the TMA, runway configuration, weather conditions, not wishing to bias decisions by aircraft operator and the Wake-Vortex (WV) category of aircraft (Venkatakrishnan, Barnett & Odoni 1993, Carr, Erzberger & Neuman 1999). After a sequence decision has been made the pilot will receive their instructions and follow ATC through a final standard approach route before touchdown (Graves 1998).

The landing order of aircraft can make a difference to efficiency measures such as delay. This is due to the wake vortices that follow aircraft as they fly. Aircraft are placed in wake vortex categories by the ATC authority responsible for control of landing aircraft at the airport. Different ATC authorities have similar categories, but there may be slight variations in the number of categories, or classifications of aircraft in those categories. Based on the wake vortex characteristics of each class, a matrix is generated to advise controllers on the minimum separation distance that must be applied between aircraft categories. Much of the work done on sequencing aircraft reported in Section 2.6 has focused around optimizing the minimum separations between aircraft. The main reason that people have focused on this is that it is believed to be a major factor in determining runway capacity (Dear & Sherif 1991, Venkatakrishnan et al. 1993). Optimize minimum separations and more aircraft may be able to use the runway.

Ultimate control of the landing sequence rests with controllers in the TMA. Most of the authors that have assessed different sequences have focused on the effect they may have on aircraft in the TMA (Section 2.7). However, use of AMAN technology permits controllers in approach sectors before the TMA to be involved because landing sequences can be produced before aircraft enter the TMA. If good communication exists between controllers and all receive the same landing

sequence advice then aircraft may be controlled towards the landing sequence further away from the airport.

2.3 The scheduling problem

The arrival aircraft scheduling problem is to determine the landing sequence of aircraft onto single or multiple runways, then assign the aircraft landing times. Previous work has designed algorithms to schedule aircraft on to single and multiple runways to optimize objective functions. The objective functions of these algorithms have been to minimize: sequence makespan C_{MAX}, sum of weighted tardiness $\sum w_j T_j$, sum of completion times $\sum C_j$, general non-decreasing linear functions, general non-linear functions and functions based on priority of individual aircraft. Constraints on the sequences have included sequence-dependent times between aircraft $S_{i,j}$, time-windows of earliest and latest land time $[r_j, \overline{d}_j]$ and Constrained Position Shifting (CPS). This section presents the general problem in natural language and machine job formulations.

2.3.1 Natural language formulation

Objective The Arrival Management scheduling problem is to determine the order aircraft land on single or multiple runways so that they optimize an objective function. Objective functions of interest may include delay (e.g. sum of weighted tardiness) or use of the runway (e.g. sequence makespan).

Constraints

1. Aircraft type. Each aircraft belongs to one of X wake-vortex categories.
2. Wake vortex matrix. The minimum time between landing two aircraft of type i and type j on a runway is given by a matrix $S_{i,j}$, i.e. there are sequence dependent minimum time separations between landing aircraft on the same runway. Some of the operational systems such as RATE-PC or MAESTRO do not consider sequence dependent separations when generating their sequence land times (Eurocontrol 2000*b*). However, all of the algorithms described in Section 2.6 explicitly consider this constraint.
3. Preferred land time. Each aircraft j has a preferred landing time d_j. Algorithms with objectives that are based on delay implicitly include a preferred land time concept. Some authors such as Psaraftis (1980) have considered the preferred land time to be the same as the earliest land time.

For others, including Beasley, Sonander & Havelock (2001), preferred time may be later than earliest land time.

4. Deadline. Each aircraft j has a deadline $\overline{d}_j$ on the time it must land by. Not all of the algorithms in Section 2.6 include this constraint, e.g. Bianco et al. (1999). However, all aircraft have a limited amount of time they may spend in the air. Eventually they will run out of fuel.

5. Earliest land time. Each aircraft j has an earliest possible land time, or release date $\hat{r}_j$ which may be later or equal to the preferred land time. The algorithms in Psaraftis (1980) did not include earliest possible land times. In this case all aircraft sequenced are available to be sequenced now. This constraint is necessary if an AMAN tool is to provide sequence advice before aircraft enter the TMA.

6. Precedence constraints. If aircraft are placed in holding patterns, then they must not be sequenced out of their hold before aircraft that preceded them (Graves 1998). This gives rise to precedence constraints. Also, aircraft that use the same Intermediate Approach Fix (IAF) point tend to follow similar arrival routes. If overtaking aircraft with similar routes is forbidden then precedence constraints on the order of arrival to IAF points may be used. None of the authors who have proposed algorithms for sequencing aircraft arrivals have explicitly included precedence constraints in their models of the scheduling problem.

7. Constrained Position Shifting (CPS). These constraints do not allow aircraft to be switched more than M positions from their FCFS position. They are developed by Psaraftis (1980). Dear & Sherif (1991) believe that they are a good way to ensure that the sequence is stable as it is updated over time.

8. Aircraft weighting. Some aircraft may be more important to schedule than others. This preference may be absolute or quantitative.

2.3.2 Machine job formulation

The problem above may be described in machine job parlance if we take aircraft to be called jobs and runways machines. Other authors such as Psaraftis (1980), Venkatakrishnan et al. (1993) and Bianco et al. (1999) also believe that the machine job form is appropriate to sequencing arrival aircraft. The following objectives and constraints are taken from the natural language formulation above.

Objective The objective is to find a sequence that minimizes $\sum g_j(C_j)$or $\max\{g_j(C_j)\}$, where g_j is an arbitrary non-decreasing function of the completion time C_j of job j. This type of objective function includes makespan C_{MAX}, sum of weighted tardiness $\sum w_j T_j$, sum of completion times $\sum C_j$, maximum tardiness L_{MAX} and number of tardy jobs $\sum U_j$.

Constraints and definitions

1. Each job belongs to one of X types.
2. The processing time of a job of type i followed by job type j on each machine is given by $S_{i,j}$.
3. Each job j has a time by which it is due to be processed. For job i this is denoted d_j.
4. Jobs have a deadline $\overline{d}_j$, the latest time they may complete.
5. Jobs have a release date $\hat{r}_j$, the earliest job completion time. This is slightly different to the usual release date definition of earliest job start time.
6. Jobs in the same stream may be subject to precedence constraints.
7. Jobs must not be switched more than M positions from their FCFS position.
8. Each job has a weight w_j indicating its importance if this importance may be quantified.

The classification of scheduling problems used by Pinedo (1995) defines machine scheduling problems with notation of [machine description] | [problem constraints] | [objective function]. In this notation the machine scheduling problem described above is represented as a $m|CPS, prec, \overline{d}_j, \hat{r}_j, S_{i,j}| \sum g_j(C_j)$ problem, where m is the number of parallel machines.

2.3.3 Alternatives

The scheduling problem definition above does not share all the features of those previously studied. The first difference is that the formulation assumes sequence dependent separations where the triangle inequality holds. Some papers published also make the same assumption, e.g. Bianco et al. (1999). Algorithms developed with this assumption will in general only be applicable for sequencing arrival traffic, not a mix of arrival and departures. For airports such as London-Heathrow with a designated runway for arrivals, these algorithms are

appropriate. However, for others such as London-Gatwick where the runway operates in a mixed-mode of departures and arrivals these algorithms could not be run to consider departures in conjunction with arrivals.

The machine job form of the scheduling problem works with non-decreasing linear objective functions. Beasley et al. (2001) work towards more complicated non-linear objectives. Indeed, this is a central reason as to why they develop their population-heuristic. However, all other papers in Section 2.6 choose non-decreasing linear objectives, and the most advanced sequencing algorithms in operational AMAN systems (in COMPAS and OSYRIS) use linear objectives related to delay. One may argue that non-decreasing objective functions are more appropriate to the real world problem from a human-factors point of view. Hopkins (1995) believes that for decision-support tools such as an AMAN to benefit air traffic control, controllers must be able to interrogate the system about its decisions and so understand why the solutions have been chosen. So even if non-linear objective functions produce better schedules for controllers, they may not be accepted by controllers because they don't understand how they were formed. In any case, the issue of whether non-linear or linear objective functions are best is not clear and perhaps each is as justifiable as the other.

A final difference is that precedence constraints are not used in the algorithms found in the literature. This is surprising since they are based on a physical aspect of landing aircraft. Venkatakrishnan et al. (1993) even examine the performance of their algorithm with respect to the overtakes it generates for the precedence order because:

> "...they [the controllers] are worried about having to handle too many overtakes, deviations from the first-come, first-serve (FCFS) sequence for aircraft within the *same* stream (the same entry fix)."

Some authors, such as Fahle & Wong (2003), have distanced control actions from the sequence decision process claiming that the two are independent. This work will consider sequencing with and without the control related precedence constraints.

2.4 The delay-share problem

2.4.1 Background

The delay-share problem is to determine how best to implement sequences that have been generated. For instance, suppose an aircraft is one hour away from

landing at an airport and its position in the landing sequence dictates that it must land 10 minutes later than it would flying its normal flight path. How should the aircraft lose the 10 minutes?

The experimental OSYRIS system developed for the EEC (Barco-Orthogon 2002) uses a letter of agreement approach where each ATC sector i inside the AMAN boundary agrees to absorb a maximum delay $\tilde{d}(i)$. If the total delay D_{tot} needed to be absorbed for an aircraft is greater than the sum of maximum delays $\sum_{i=1}^{k} \tilde{d}(i)$ for the k ordered approach sectors of the aircraft, then maximum delay is absorbed in all the sectors up to sector k (the last) which is assigned its maximum plus the residual D_{resid}. i.e. the revised exit times $rt(i)$ of each sector i depend on the original forecast exit time $ot(i)$ in the following way:

$$rt(j) = \begin{cases} ot(j) + \sum_{i=1}^{j} \tilde{d}(i) & 0 < j < k \\ ot(j) + \sum_{i=1}^{k} \tilde{d}(i) + D_{resid} & j = k. \end{cases}$$

If $D_{resid} = 0$ then two strategies have been designed for use in OSYRIS. The first spreads delay evenly through the route of the aircraft:

$$rt(j) = ot(j) + \sum_{i=1}^{j} \frac{\tilde{d}(i)}{\sum_{i=1}^{j} \tilde{d}(i)} D_{tot} \quad 0 \leq j \leq k.$$

The second absorbs delay as late as possible:

$$rt(j) = \begin{cases} ot(j) & D_{tot} - \sum_{i=j+1}^{k} \tilde{d}(i) \leq 0 \\ ot(j) + D_{tot} - \sum_{i=j+1}^{k} \tilde{d}(i) & D_{tot} - \sum_{i=j+1}^{k} \tilde{d}(i) > 0. \end{cases}$$

The RATE-PC delay-share strategy may be seen as a mix of the late as possible and even-spread strategies in OSYRIS. It tries to keep the TMA full with around 10 - 11 aircraft. If aircraft cannot be taken any more by the TMA then controllers in sectors feeding the TMA (the ACC) are given a delay to absorb. This can be done linearly up to a maximum capacity point when aircraft will use a holding pattern (Eurocontrol 2000*b*).

The MAESTRO system at Copenhagen assigns delays to aircraft over the sectors in a similar manner to the OSYRIS even-spread strategy. If total delay is greater than may be absorbed with linear absorption through sectors then orbital holding patterns are used (Eurocontrol 2000*b*).

The CTAS system does not work in the same way as these AMAN systems - landing sequences are formed when the FAST component takes over around 35nm

from the airport. Any delays here are absorbed in upper sectors through vectoring or orbital holding. Aircraft enter the lower sectors in the TMA at times appropriate to the landing sequence (Eurocontrol 2000*b*).

2.4.2 Problem formulation

The AMAN systems described in Section 2.4.1 use strategies to share delay. All the strategies implicitly or explicitly satisfy the following constraints on how aircraft may be delayed, given a landing sequence.

1. Sector definitions. The layout of the sector will have an impact on how much delay is feasible to apportion to an aircraft. For instance, the sector may have departure flights, enroute flights or other complicating factors that make it unlikely delay may be absorbed.

2. Flight paths. The route of the aircraft through a sector may have an impact on delay sharing. Specifically, the amount of time the aircraft may normally spend flying through the sector, and how feasible it is for delay to be absorbed through vectoring or speed control.

3. Aircraft. Aircraft may have special characteristics which affect the amount of delay that may be absorbed in a sector. For instance, aircraft cannot be slowed down beyond a certain point.

4. Controller workload. Air Traffic controllers have a maximum amount of time they may absorb for a plane flying through their sector which is a function of time. Controllers may be able to absorb a lot of delay if they are not busy, or none if they are.

If an objective function were added to the constraints an optimization problem would result. For instance, delay could be shared to minimize a function of amount of orbital near-airport holding, a controller workload measure or uneconomic flight profiles. However, the main delay sharing objective described in Eurocontrol (2000*a*) is to:

> "...improve optimal flow of traffic towards airport, prevent holes in the sequence and overloading".

There is no evidence to suggest that simple strategies will not help achieve this aim. This work therefore sees strategies rather than optimization as a means to the delay sharing aims.

2.5 Developed AMAN systems

Several AMAN systems have already been developed and tested. Some have been, or are operational today, others were developed but never integrated into a real control system. There follow descriptions of the major AMAN tools developed. Each description includes the aims of the system and how the AMAN works towards them.

At Stockholm Arlanda, software called RATE-PC was developed in Visual Basic by the Swedish CAA to run on a desk-top computer (Eurocontrol 2000*a*, Eurocontrol 2000*b*). It was based on an earlier paper-based RATE method. The RATE method was introduced to increase runway capacity, improve working position for controllers and provide information to users (e.g. how much delay to expect). It worked using Estimated Times of Arrival (ETAs) at the runway; calculated using a trajectory prediction program. The estimated FCFS sequence that resulted was used as the landing sequence. A maximum landing rate was estimated using a spreadsheet that took into consideration weather, runway configuration and traffic mix. Using the landing rate, aircraft were assigned slots to land, so planned times between successive aircraft were not based on minimum separations. Controllers were advised on the delays that aircraft needed to absorb in order to land in their slot time.

The Swedish CAA plan to move to a new control centre in 2004. At this centre they will stop using the RATE-PC method and switch to the MAESTRO (Means to Air Expedition and Sequencing of Traffic with Research of Optimization) AMAN. The MAESTRO system is in use at a number of places in Europe; at Copenhagen-Kastrup, Malmo-ACC, Paris-Orly and its birthplace Paris-Charles de Gaulle (Eurocontrol 2000*a*, Eurocontrol 2000*b*). Copenhagen installed MAESTRO in order to improve controller workload, make better use of airspace, reduce fuel consumption, split delays between Swedish and Danish sectors and provide data for users. The AMAN works by advising controllers on control actions for landing aircraft in a FCFS sequence, based on maximum arrival rates (not separation criteria). This advice is made further away from the runway than in RATE-PC, in enroute sectors. Delay is absorbed evenly through all the sectors to the runway. MAESTRO has a series of states for aircraft from where the aircraft's sequence position may change, to where its position is frozen, to where the controller may no longer change the landing position.

The Traffic Management Advisor (TMA) tested at Dallas Ft. Worth is a sub-component of the Centre TRACON (Terminal Radar Approach Control) Automation System (CTAS). The CTAS system was built to reduce stress and workload in controllers, reduce delays and increase safety (Eurocontrol 2000*a*, Eurocontrol 2000*b*). The TMA system is used for aircraft arrivals 35nm to 200nm back from the airport. It does not use a landing sequence to advise control actions, rather it uses estimated sequence times over navigating beacons. The sequence advised over these points is FCFS , unless controllers add constraints in which case the algorithm in Wong (2000) works to make sure constraints are obeyed. Delays that aircraft need to lose in order to arrive in the generated sequences are then advised to controllers. The component of CTAS that aids controllers merging sequences of arriving traffic close to an airport in the terminal area is called FAST. The component of FAST that generates sequence and runway advisories is termed pFAST. The landing sequences it forms are based on fuzzy-logic, with parameters set after real-time simulations with controllers (Davis, Isaacson, Robinson III, den Braven, Lee & Sanford 1997).

The first version of the COMPAS (Computer Oriented Metering Planning and Advisory Program) system in use at Frankfurt went operational in 1989. The general goal of COMPAS is to optimize use of available runway capacity (Eurocontrol 2000*b*). The sequencing algorithm tries to do this by minimizing total delay time. Initially a FCFS sequence is generated, then time conflicts are sought out. If a time conflict between aircraft is found, a branch-and-bound algorithm runs on a "serve the earliest conflict first" principle to order the aircraft so that total delay time is minimized. Controllers are then advised on the sequence, schedule and control information. Delay is tried to be absorbed as far back from the airport as possible. The system is not dynamic as aircraft are added to the landing sequence as they enter the system's airspace - no updating of landing sequence is performed on aircraft after this point.

Another system named FAST (Final Approach Spacing Tool) was being developed by National Air Traffic Services (NATS) for London-Heathrow (Eurocontrol 2000*a*, Eurocontrol 2000*b*) but the project was discontinued for a number of reasons that included problems with wind parameters. Its aim was to assist controllers in achieving minimum separation spacing on descent into the airport. It did this by using a landing sequence defined by a controller and then gave advisories on two turning points between holding stacks and runway based on conditions at the moment in time using information about wind, weather and runway configuration.

The AMAN that the EEC is experimenting with has been bought from Barco-Orthogon. It is called OSYRIS (Orthogon System for Real-time Inbound Sequencing). The philosophy underlying it is similar to MAESTRO and COMPAS, and it shares the general efficiency and controller oriented aims. One difference between this system and the others is its sequencing algorithm. This is a modified form of constrained position shifting (Barco-Orthogon 2002), an idea found in Psaraftis (1980). The idea is that aircraft have a maximum number of places they may switch from their FCFS, and by enumerating all possibilities the experimental OSYRIS algorithm chooses the solution that minimizes the objective. The objective may be to minimize total delay or deviation from schedule (i.e. landing early has the same cost as landing late). Two strategies are available for sharing any delay that may result in the experimental system - delay absorption as late as possible and delay absorption evenly spread.

2.6 Sequencing algorithms literature review

The algorithms used in developed AMAN systems have all been quite basic, most based on FCFS slot allocation rather than wake-vortex separations. Different approaches and algorithms have been developed to sequence landing aircraft in various journals. These are described here.

Some authors have taken a deterministic machine job scheduling view of the arrival aircraft sequencing problem, where aircraft are regarded as jobs. The special characteristics of jobs in a machine environment enabling use of this model with aircraft are sequence-dependent processing times (or set-up times) $S_{i,j}$, release dates r_j, due dates d_j and deadlines $\overline{d}_j$. The sequence-dependent separation matrix $S_{i,j}$ corresponds to the minimum wake vortex separations between aircraft. Release dates r_j and deadlines $\overline{d}_j$ mean that each aircraft j has a time window $[r_j, \overline{d}_j]$ in which it may land. The due date d_j is the time that the aircraft would prefer to land. In this model of sequencing aircraft, runways are machine resources - only one job may be processed on each machine at any moment in time. The tractability of a machine job approach arises from the relatively small number of aircraft wake-vortex classifications - typically there are between 3 and 5 categories depending on the ATC authority.

Psaraftis (1980) presented a Dynamic Program (DP) to minimize $\sum g_j(C_j)$, where g_j is a general non decreasing cost function of job completion time C_j, for the single-machine problem where all jobs have identical release and due dates. Using the notation for machine job scheduling problems from Pinedo (1995), he

tackled a $1|S_{i,j}|\sum g_j(C_j)$ problem. The algorithm sequences aircraft that are all ready to land now, are due to land now, have no constraint on the maximum time they may wait to land and will all land on the same runway. The dynamic program had complexity $N^2\prod_{i=1}^{N}(1+k_i^{max})$ where N is the total number of job types and k_i^{max} the number of jobs of type i. Psaraftis (1980) also gave a dynamic program for the same problem but with priority constraints. The author termed this problem as one of Contrained Position Shifting (CPS), where a job may not be shifted more than a Maximal Position Shift (MPS) of M places from its original FCFS position. The CPS dynamic program had complexity no worse than that of the $1|S_{i,j}|\sum g_j(C_j)$ DP. The two approaches were applied to an example of 15 aircraft of three types waiting to land onto a single runway. Comparisons were made between the MPS used and the optimum without CPS. Finally, Psaraftis suggested modifications to the DP formulations that would allow the problem $m|S_{i,j}|\sum g_j(C_j)$ of m machines to be tackled. Venkatakrishnan et al. (1993) described a revised version of the non-CPS DP in Psaraftis (1980) that introduced a heuristic element, adding aircraft in the dynamic program based on lower bounds of time windows on aircraft land time. i.e. this heuristic tackled the $1|r_j,\bar{d}_j,S_{i,j}|\sum g_j(C_j)$ problem.

Bianco, Ricciardelli, Rinaldi & Sassano (1988) worked on the $1|r_j,S_{i,j}|C_{MAX}$ problem. This single-machine problem caters for the situation when not all jobs may start immediately - there exist release dates r_j, and there are sequencing dependent separations between jobs. Its objective is to minimize makespan C_{MAX}, i.e. the time the last job will finish. One may note that the problem is equivalent to the traveling salesman problem with additional time constraints. It also corresponds to the problem of maximizing use of the runway in the aircraft sequencing problem, where aircraft have earliest land times but no latest. A mixed integer program was formulated with upper bounds, lower bounds and dominance criteria for use in a branch and bound algorithm. Computational results of this algorithm were presented for 10, 15 and 20 job problems. Bianco et al. (1999) modelled jobs with the same characteristics as Bianco et al. (1988) on a single machine. The objective function changed to one of minimizing the sum of completion times $\sum C_j$, i.e. the $1|r_j,S_{i,j}|\sum C_j$ problem. This problem is equivalent to the cumulative traveling salesman problem with additional time constraints. An exact dynamic program, three lower bounds and two heuristic algorithms were formulated. Computational results for the heuristics and lower bounds were presented for 10, 20, 30, 40 and 50 job problems. Computational results were also assessed for the aircraft sequencing problem. The heuristic algorithms were compared to FCFS for two traffic samples of 30 and 44 aircraft of 4 types

approaching the TMA. In these samples the heuristics generated significant reductions in mean aircraft delay over FCFS.

A number of papers written on general machine job scheduling with sequence dependent setups have not specifically proposed algorithms for the AMAN scheduling problem but could be used in such a framework. Ovacik & Uzsoy (1994) produced decomposition heuristics to tackle the $1|r_j, S_{i,j}|L_{MAX}$ problem. The idea behind the heuristics was termed Rolling Horizon Procedure (RHP). Here the scheduling problem was decomposed into a series of smaller sub-problems. The limited size of these sub-problems allowed exact procedures to be used to solve them. The overall solution was then found by gluing together the sub-problem solution segments. The authors developed an exact branch-and-bound algorithm to solve the sub-problems and examined the tradeoff between computation time and solution quality for different size sub-problem. They also compared the solution quality of the RHP heuristic with the Earliest Due Date (EDD) rule and a modified EDD rule with a local search procedure. Their experimentation found the RHP procedure always produced better results than the other methods. Ovacik & Uzsoy (1995) applied the RHP procedure for the multiple parallel machine equivalent $m|r_j, S_{i,j}|L_{MAX}$ problem. Three heuristics were developed. The first assigned jobs to machines based on the EDD rule, then applied the same RHP procedure with the branch-and-bound algorithm described earlier. The second heuristic assigned jobs to machines by moving through time and machines. The set of jobs available to be scheduled at each time point is ordered using the optimal branch-and-bound algorithm. A number of these, based on an input parameter are then be added to machine i. The process continues to the next time point, a new set of jobs considered, the optimal order made and a number added to machine $i + 1$ and so on, until all the jobs have been assigned to machines. The third heuristic followed the same pattern as this except that as it moved through time it checked if by sequencing the jobs on any other machine, the first job (and therefore all jobs) would complete earlier. If it would, the jobs are added to the job list for that machine. All the heuristics were compared to the EDD rule and another EDD augmented with local search procedures. They outperformed these benchmarks, with the second and third heuristic procedures performing the best in general for the experiments run. If applied to arrival aircraft scheduling these RHP algorithms work to minimize the maximum lateness of aircraft, where aircraft may have an earliest land time but no deadline constraint. The parallel machine case corresponds to multiple runways, the single machine a single runway.

Uzsoy, Martinvega, Lee & Leonard (1991) worked on a job shop problem where jobs must be processed at different workcentres. They split up this problem by sequencing jobs at each workcentre. This lead to development of an optimal branch-and-bound algorithm and a local search heuristic to tackle the $1|prec, S_{i,j}|L_{MAX}$ problem. These algorithms worked on the $1|prec, q_j, S_{i,j}|C_{MAX}$ problem where delivery times q_j are needed after job j is processed. By setting $q_j = K - d_j$ where $K = \max_j(d_j)$ the authors noted that this problem is equivalent to $1|prec, S_{i,j}|L_{MAX}$. However, Ovacik & Uzsoy (1994) found that the branch and bound algorithm for this problem did not perform well for more than 15 jobs. Applied to sequencing aircraft these algorithms produce a sequence to minimize maximum delay to an aircraft assuming that all aircraft are ready to land now, and there exist precedence constraints on the order they may land. Uzsoy, Lee & Martinvega (1992) formulated a heuristic algorithm for the $1|prec, r_j, S_{i,j}|C_{MAX}$ problem. This would seek to land aircraft in as short a time as possible, based on variable earliest land times, precedence constraints between aircraft and landings on a single runway. Another heuristic was developed for the $1|prec, S_{i,j}|\sum U_j$ problem. Here all aircraft sequenced are ready to land now, subject to precedence contraints and the objective is to minimize the total number of aircraft with delay. Exact dynamic programs were given for $1|prec, S_{i,j}|L_{MAX}$ (the same problem as Uzsoy et al. (1991)) and $1|prec, S_{i,j}|\sum U_j$. The DP for $1|prec, S_{i,j}|L_{MAX}$ had computational complexity $O(m^2(N+1)^m T)$ where N is the maximum number of jobs in any lot (each lot is ordered by precedence), m the number of lots and T is an upper bound on the completion time. The DP for $1|prec, S_{i,j}|\sum U_j$ had computational complexity $O(m^2(N+1)^m n^S)$ where S the number of distinct setup time values. Asano & Ohta (1996) looked at the $1|\bar{d}_j, r_j, S_{i,j}|\sum E_j$ problem, where E_j is the earliness of job j, and formulated an optimal branch-and-bound algorithm. Applied to aircraft the objective function is to minimize the amount of time aircraft land early, subject to aircraft having and earliest land time and a deadline. Computational efficiency of the algorithm was improved through a dominance relation and a lower bound.

The aircraft sequencing problem where the triangle inequality between minimum separations breaks down has also been considered. Discussions with Air Traffic controllers at Eurocontrol have indicated that for arrival aircraft this is rarely, if ever, a problem. However, it may become relevant when departures are considered in conjunction with arrivals. For example, a departure may fit in between two arrivals, but the minimum separation between the two arrivals may be greater than the sum of minimum separations between the first arrival - departure - second arrival (see Figure 2.1). Algorithms that have the feature of

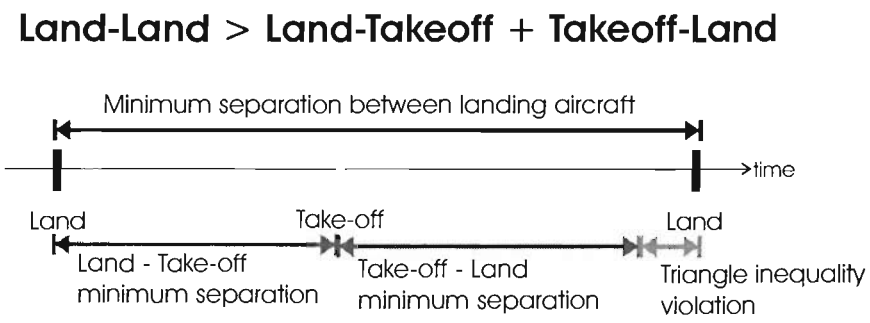


Figure 2.1: Triangle inequality violation example

working when the triangle inequality breaks down may thus be used for sequencing arrival aircraft, departing aircraft or a mixture of both.

Ernst, Krishnamoorthy & Storer (1999), Beasley, Krishnamoorthy, Sharaiha & Abramson (2000), Beasley et al. (2001), Fahle & Wong (2003) and Beasley, Krishnamoorthy, Sharaiha & Abramson (2004) all considered the general problem where separations are enforced between all aircraft, not only successive. A Mixed Integer Program (MIP) model of this problem was first formulated by Beasley et al. (2000) with a general linear objective function. Some computational results were presented. Ernst et al. (1999) developed a lower-bounding method for this formulation with a linear objective measuring weighted differences between target time and sequenced time. This method was employed in an exact branch-and-bound algorithm and a meta-heuristic that used a genetic algorithm. Computational results were assessed. Another genetic algorithm heuristic tackling the same formulation as in Beasley et al. (2000) but for a non-linear or linear objective function was presented by Beasley et al. (2001) with more results comparing quality of solution and speed. Fahle & Wong (2003) presented further heuristic constraint programming and local search methods to tackle a similar MIP formulation to Beasley et al. (2000) with a linear objective measuring weighted differences between target time and sequenced time. The computational results from these techniques were compared with exact methods.

The branch-and-bound algorithms developed in Trivizas (1998) were based on the same idea of Constrained Position Shifting (CPS) as Psaraftis (1980), but referred to as Maximal Position Shifting. Different versions may be used in cases where the triangle inequality is, or is not, violated. If the triangle inequality is violated the sequencer deals with this by keeping track of the number of departures that left after the last arrival - this approach does not check all the separations, but a smaller subset. The algorithms worked towards objectives of

either sequence makespan C_{MAX} or sum of weighted tardiness $\sum w_j T_j$.

All of the sequencing methods discussed so far have had as their objective functions some measure of efficiency - they are interested in minimizing some measurement of delay and/or maximizing the landing rate. While this may be good for the system as a whole it may not necessarily be what the airlines want. The possibility of allowing airlines to express relative arrival priorities to ATC was investigated in Carr et al. (2000). Carr et al. (1999) explored the problem of how to do this in a fair way - some aircraft may be delayed when other aircraft from the same operator have greater priority. Delay-exchange algorithms were presented. Milan (1997) also described two priority rules for sequencing aircraft. The first prioritized aircraft by passengers. Data such as the number of passengers, the cost of passenger delay and the number of seats on each aircraft were taken into account. The second priority rule prioritized aircraft based on their expected delay.

Authors have commented on differences between the static scheduling problem at one particular moment in time, and the dynamic scheduling problem as time evolves. The branch-and-bound algorithm described by Trivizas (1998) was updated dynamically through a simple dynamic program which ran periodically when a number of new aircraft enter the system. Dear & Sherif (1991) promoted constrained position shifting as a generic methodology applicable to any sequencing technique in order to limit the changes to sequence when re-sequencing occurs. In some sense Beasley et al. (2004) used a constrained position shifting idea. They modelled change to sequences as having a cost, and proposed to include these costs in any run of an algorithm. This was demonstrated on the genetic algorithm from Beasley et al. (2001). Venkatakrishnan et al. (1993) implemented their sequencing algorithm dynamically in two ways. Their first approach was to run the algorithm on those aircraft more than 5 minutes from touchdown aircraft every time a new aircraft entered the terminal area. The second approach re-sequenced when new aircraft entered the system, but shrank the time-windows on when the aircraft may land linearly as aircraft approached the runway. Dynamic use of algorithms is not a problem limited to sequencing aircraft. The Rolling Horizon Procedures in Ovacik & Uzsoy (1994) and Ovacik & Uzsoy (1995) were developed to be used in a dynamic environment. They worked by splitting up the problem into sub-problems; those jobs available to be sequenced now, and those jobs forecast to be sequenced at points in the future. Only the decisions related to the current decision point are implemented, the other decisions may be revised at the next decision point.

2.7 Assessment of sequencing algorithms

The algorithms developed in Section 2.6 are all based on models of the arrival process. The optimization models make a number of simplifying assumptions that may produce unexpected results when applied to the real problem.
Most algorithms have been developed to run given a set of aircraft waiting to land. This has been termed the static problem. A number of authors have evaluated their algorithms on the static problem only. These include Psaraftis (1980), Ernst et al. (1999), Bianco et al. (1999) and Fahle & Wong (2003). The real problem facing controllers is more dynamic, as aircraft arrive sequences need to be updated. This has been termed the dynamic problem (Venkatakrishnan et al. 1993). This section describes the methods taken to assess how the different sequences and their modelling assumptions may affect the ATC system. That is, how the algorithms assumption of perfect information affects its performance in a dynamic environment. The published work only considers the effects of implementing sequences on aircraft in the TMA around an airport. Very little work has been done to investigate effects of sequencing further back from the TMA.

Simulation A discrete-event simulation model of Rome airport was developed and validated by Andreussi, Bianco & Ricciardelli (1981). Its purpose was to be used to evaluate different scheduling strategies, but little experimentation of different strategies appears to have been carried out on it. Trivizas (1994) used the MPS algorithm from Trivizas (1998) to compare runway capacity against different maximum position shifts in the algorithm. This was done by using a detailed 4D simulation model, TMSIM, developed by Massachusetts Institute of Technology, USA. This simulation flies each aircraft from departure to destination modelling routing processes, communication and conflict detection and resolution. The model was validated by air traffic controllers visually and also checked for errors on minimum separation between landings. Results from experiments on one traffic sample of take-offs and landings at Frankfurt and another at Chicago O'Hare indicated that substantial improvements in capacity and delay may be achieved with increasing values of maximum shift. Unfortunately, there was no indication of what might happen with other traffic samples. Both traffic samples used were dominated by medium type aircraft, so it may be that the improvements found were due to improvements in sequencing departure aircraft.

Dear & Sherif (1991) evaluated FCFS and their CPS algorithm methodology using a fast-time simulation model of a TMA, but model validation was not reported. It seems unlikely the simulation was used to model a real airport.

The arrival process of aircraft at the system boundary was modeled as a Poisson process with constant rate. Three types of traffic sample were generated and multiple runs each with 500 aircraft were made on the model. Mean delay results from FCFS and the CPS algorithm were compared. The differences were sometimes quite impressive, there was a 76% reduction in delay for a particular aircraft class, and the CPS algorithm always reduced delay. However, the authors did not include any tests for statistically significant differences. Beasley et al. (2004) used a similar approach to test out their dynamic updating procedure using the algorithms developed by Beasley et al. (2000) and Beasley et al. (2001). Thirteen traffic samples were investigated with the number of possible runways between 1 and 5. The traffic samples were generated from a Poisson process with constant rate where aircraft appeared 10 minutes before the time they were due to land. The algorithms were compared to FCFS rules by their objective function values. The smart algorithms always did at least as well as FCFS: in a dynamic environment between 36% and 55% improvement in objective function values was found. However, it is difficult to interpret the significance of the results for a real airport since the process of generating arrival traffic seems sensible, but was not validated for any airspace.

Beasley et al. (2001) compared the landing sequence from a real traffic sample of 20 aircraft arrivals into London-Heathrow, with a sequence from their algorithm. The methodology used to make the comparison is philosophically debatable. The authors set recorded land times in the traffic sample to be target land times. A delay indicator was calculated using the difference between these targets, and the times aircraft would have landed if they were separated by the deterministic separation matrix. It may be argued that this delay measure is meaningless, since by definition delay in the real landing sequence is now zero. The traffic sample delay indicator was compared to a landing sequence, with deterministic separations, formed by the algorithm. The comparison is made in this way in an attempt to make it fair, by removing differences between real-life separation distances and the deterministic separation matrix used for the algorithm. However, the procedure also limits interpretation of its results. Consider a comparison between one traffic sample of n aircraft with landing times $\boldsymbol{C}^{(d)} = \{C_1^{(d)}, C_2^{(d)}, \ldots, C_n^{(d)}\}$ and an alternative sequencing algorithm's landing times $\boldsymbol{C}^{(a)} = \{C_1^{(a)}, C_2^{(a)}, \ldots, C_n^{(a)}\}$. The purpose of the analysis is to compare $\boldsymbol{g}(\boldsymbol{C}^{(d)})$ to $\boldsymbol{g}(\boldsymbol{C}^{(a)})$, where $\boldsymbol{g}$ is an arbitrary function of the aircraft landing times, such as total delay. The authors comparison problem is that the process driving $\boldsymbol{C}^{(d)}$ is not the same as that for $\boldsymbol{C}^{(a)}$. The authors do not choose to model the separation between aircraft in real life to make a direct comparison fair.

Rather they compare $\boldsymbol{g}(\boldsymbol{t}(\boldsymbol{C}^{(d)}))$ to $\boldsymbol{g}(\boldsymbol{C}^{(a)})$, where $\boldsymbol{t}(\boldsymbol{x})$ is a transformation of landing times $\boldsymbol{x}$. The transformation used for $j = 1, \ldots, n$ is:

$$t(x_{p(j)}) = \begin{cases} x_{p(1)} & \text{if } j = 1 \\ \max\{t(x_{p(j-1)}) + S(p(j-1), p(j)), \hat{r}_{p(j)}\} & \text{otherwise} \end{cases}$$

where $S(i, k)$ is a deterministic separation matrix of the minimum time between aircraft i and k, $p(j)$ is the position in $\boldsymbol{x}$ of the aircraft with sequence position j and $\hat{r}_j$ is the earliest land time possible for aircraft j. The comparison of $\boldsymbol{g}(\boldsymbol{t}(\boldsymbol{C}^{(d)}))$ to $\boldsymbol{g}(\boldsymbol{C}^{(a)})$ is only meaningful if conclusions may still be drawn on the real-world measure $\boldsymbol{g}(\boldsymbol{C}^{(d)})$. However, this is not the case because the transformation used is not invertable. That is, it is not possible to perform the operation $t^{-1}(t(x_{p(j)})) = x_{p(j)}$ for $j = 2, \ldots, n$ because $x_{p(j)}$ is not part of the transformation's definition. It follows that no inference may be made on $\boldsymbol{g}(\boldsymbol{C}^{(d)})$ from $\boldsymbol{g}(\boldsymbol{t}(\boldsymbol{C}^{(d)}))$. This result may also be explained in more general terms. The procedure is debatable because the general philosophy of modelling has been reversed. Models are generally built, based on a series of assumptions, so that inferences may be made on real world data. In the comparison procedure, the real world data has been constructed in a way that inferences may be made on a model of landing sequences.

The smart algorithm is reported to fare better than various objectives calculated from the new actual sequence, which led the authors to deduce (Beasley et al. 2001):

> "Were this to be repeated across time such a saving would have the potential for Heathrow to cope with (approximately) one extra landing per h. This would be a significant improvement."

It was stated that other data sets were considered but not presented for reasons of brevity, so this conclusion was not drawn from only the single, 20 aircraft traffic sample analysis that was actually presented. But the analysis procedure casts a serious doubt on the conclusions.

Perhaps the most rigorous examination of a sequencing algorithm has been made by Venkatakrishnan et al. (1993). They got around the problem of comparing simulation results with real traffic sample landing sequences, by using an empirical statistical model of landing time intervals to form algorithm sequence landing times. Target land times were set as the earliest possible land time once an aircraft had entered the TMA. Data analysis was carried out on a traffic sample of 18 days traffic at heavy traffic periods for 3 to 4 hours at Logan airport. It found

that landing time intervals was influenced by two factors - wake vortex category of landing aircraft pairs and runway configuration. The model was calibrated using half of the data so that given a pair of landing aircraft a lookup table existed with the expected mean, standard deviation and 25th percentile of landing times between these aircraft. The model was validated using the remaining half of the data. New estimates were found and appropriate tests for simultaneous comparison of estimates of the two sets of parameters led to the conclusion that the null hypothesis of no difference could not be rejected at the 5% level. Comparison between algorithms and real-life data was based on the assumption that since the separation between aircraft in real life was explained by the statistical model, using its mean value separation matrix with the new algorithm would result in a like with like comparison. i.e. any difference in performance would be attributable to sequencing algorithm only. This was a good idea, but some information may have been lost by using the mean of the statistical model. Their modified CPS algorithm was run for the static case with perfect information, and in a dynamic fashion, as aircraft arrived in six real traffic samples. The algorithm performed best in the static case. Performance of the algorithm in the most realistic dynamic case was also reported to compare favorably with what happened on the day. In three data sets there were reductions in delay of about 30% compared to actual sequences, but in another there was an increase by about 18%. From a statistical viewpoint the significance of a reduction in delay is dubious - the data set is size six. Consider the results found in Table XIV of the paper detailing the minimum cumulative delay for all algorithms. The DASP-2 column represents the implementation of the algorithm in a dynamic environment, with limited choice of sequence switch made by reducing the time windows on which aircraft may land as they approach the runway. The mean difference between DASP-2 and actual results is -9.83 minutes. So the mean difference is in DASP-2's favour. However, a bootstrap experiment (Efron & Tibshirani 1998) resampling 6 observations from the 6 pairs and recording the mean difference 10,000 times returned a 90% empirical confidence interval of [-22.5,3.66] minutes. Or putting it another way, from another bootstrap experiment (B=10,000) on the percentage difference, a 90% confidence interval returned [-29.37,0.2148] percent. That is, both intervals cover zero and the hypothesis that there is no improvement cannot be rejected. The paper does go into some detail explaining why the unexpected results happened, but still leaves the impression that improved sequencing reduces delays. The very last line of the paper reads:

> "Ideally, the combination of improved sequencing knowledge will result in improved capacity, diminished delay, less need for new airports, and no reduction in air safety."

The major limiting factor with the paper is that only six traffic samples were compared and it thus hard to consider randomness. This means that the question generating the separation times from the empirical distributions. This would have enabled a more rigorous understanding of the difference. Another disappointment is that the results were not compared to a FCFS algorithm. There remains the possibility that there is no difference between FCFS and their algorithm, or FCFS and the controllers.

Carr et al. (2000) and Carr et al. (1999) applied the airline-priority based algorithms inside a simulation model of airspace around an airport. Validation of the model was not commented on. The analysis in Carr et al. (1999) paid close attention to the time advances given by the airline priority algorithms. These gave a measure how successful each scheme could be - the algorithm with the most time advances was better as airlines would be more happy with the resulting schedule. Carr et al. (2000) used a similar simulation model to compare other performance measures from a delay-exchange algorithm with FCFS. Again, validation of the model was not commented on. The authors claimed that compared to FCFS, the delay-exchange algorithm produced a schedule closer to airlines preferences while maintaining levels of delay.

Queuing theory Bolender & Slater (2000) used queuing theory from an $M|D|n$ queue and validated the analytical results from this model against a $M|M|n$ simulation. They found that the transient analysis from a $M|D|n$ queue can give reasonable results in predicting average delays when capacity is known. Results from different sequencing rules on these models indicated that as traffic intensity increased, delays decreased if heavy and light aircraft were sequenced on different runways. Milan (1997) gave a more complicated analytic queuing model to model flight delays. The queuing process in the air traffic network was analyzed in time cycles, where a batch of aircraft would be served at the airport in each cycle. From this, expressions for expected delays were found. The model was built to calculate delay when the different priority sequencing methods previously discussed were used, but no validation appears to have been done to see if it gave reasonable results. The model was fed with input parameters derived from a variety of sources. Based on model outputs from these inputs the author recommended that sequencing priority rules were chosen with caution as they will have different effects on objectives such as equitable distribution of delay or minimize flight

27

delay. The priority algorithms reduced total delays, whereas FCFS was more equitable in its distribution of delay.

Other methods for TMA sequencing In Wong, Li & Gillingwater (2002), an optimization model was developed to predict air and ground delays of arrivals and departures given a flight timetable. Predictions from the model were validated against a simulation model of flights arriving and taking off at an airport under two sequencing rules - FCFS and arrival-first (i.e. arrivals had priority over departures). The authors concluded that this validation process meant that the model could be used to assess appropriateness of scheduled time tables. Mohleji (1996) hypothesized that a more structured method of controlling aircraft from an approach fix to the runway is needed to improve landing sequences. The author believed predefined paths would make it is easier to form sequences, minimize separations and increase runway capacity through a Route-Oriented Planning And Control (ROPAC) concept. Analytic expressions to estimate flying times were developed to evaluate this concept and current setups. The current setup at an airport was compared to the ROPAC concept using one traffic sample. This sample gave good results for the ROPAC concept.

Pre-TMA delay evaluation An alternative concept of delay-share strategy as a means to shift aircraft delay back from orbit holding near the airport to enroute holding, rather than absorbing all delay in the TMA has been around for a number of years. Indeed, the MAESTRO system shifts delay back from the TMA (Eurocontrol 2000*a*, Eurocontrol 2000*b*). In 1996 Eurocontrol produced a Monte-Carlo simulation model to test the effect of simple delay-share rules when uncertainty exists on the time aircraft fly over points. Results from an initial study based on thousands of repeat runs suggested that:

> "Simple rules can be constructed that markedly reduce airport holding whilst having little impact on landing capacity."[1]

Outside of internal work such as this, nothing appears to have been published on the consequences of aircraft sequencing with delay-sharing strategy in enroute sectors.

[1]Internal email between Eurocontrol employees Mike Moore and Peter Martin, Thursday, April 24, 2003

2.8 What this work will contribute

Much has been already done on means to advise controllers landing sequences and the control actions necessary to implement them for aircraft up to a certain distance away from an airport. Operational systems have been developed, smart algorithms based on mathematical models of the sequencing problem have been formulated and some assessed using other models. However, much remains to be done.

- Models of the sequencing problem have not incorporated some constraints, so new algorithms may be developed.
- All the work published in journals has looked at the problem of sequencing once aircraft are close to the runway, in the TMA. Some real world systems work further back and so delay sharing strategies to implement generated sequence advice may be developed.
- The success of real systems has been mixed. The CTAS system has had operational problems, whereas the MAESTRO system is being used at more and more airports. What are potential benefits for ATC in implementing improved landing sequences? How confident can one be in these potential improvements?
- Work done to assess the dynamic implementation of smart sequences in the TMA has been undertaken by several authors. However, the issue on whether the use of such algorithms will improve airport capacity is not as clear cut as has been presented. The models developed all have flaws that may impact on drawing meaningful conclusions. Some have been generic models without application to a specific airspace, others have not taken enough account of variability and in most cases the validity of the model has been questionable.
- Some interesting remarks by authors may be further investigated. Venkatakrishnan et al. (1993) describe the reason why their algorithm was beaten by an actual landing sequence from controllers:

  > "This is apparently a situation where the experience of the controllers may overcome a disadvantage the algorithm has because it does not look ahead."

  A related remark in Beasley et al. (2004) explains how the dynamic problem (termed displacement problem) may produce unexpected results for sequencing algorithms:

> "It can happen, as here, that solving a displacement problem heuristically leads to decisions that are better (in terms of aircraft yet to appear - which are unknown) than decisions made by solving the same displacement optimally."

This work will address some of the gaps in arrival management research. Consequences of smart sequencing techniques on Air Traffic Control are investigated using a computer simulation model of real airspace. A caveat applies to any conclusions drawn from the model: they are based on data used in the modelling process, not on a detailed operational study. However, statistical validation methods lend credibility to the model output. Design of experiment ideas are used to take variation into account when looking for significant differences due to algorithms, algorithmic constraints delay-share strategies and traffic conditions on pre-TMA airspace. The package of algorithms, simulation, validation, design of experiments and output analysis is made to investigate how the ATC system reacts to changes in sequencing algorithms, optimization criteria and delay-sharing strategy. It is hoped that output from this work will be of use to decision makers at the Eurocontrol Experimental Centre and beyond.

Chapter 3

New algorithms

The Arrival Management sequencing problem described in the previous chapter has been well studied. However, there are some gaps in the arrival sequencing research. This chapter attempts to fill some of them. New dynamic programming algorithms are developed for general machine scheduling problems, but with the sequencing of aircraft arrivals in mind. Section 3.1 describes the modelling assumptions made for three different approaches and sets up the notation used in the subsequent dynamic programs. Mathematical formulations of the three sets of dynamic programs and their computational complexities are given in Sections 3.2 to 3.4. The algorithms work on a variety of single and multiple machine scheduling problems with sequence dependent processing times and other assumptions on precedence order, deadlines and release dates. The chapter finishes with Section 3.5 on how all the dynamic programs presented may incorporate the Constrained Position Shifting (CPS) constraints of Psaraftis (1980).

3.1 Description of algorithms

The dynamic programs presented in this chapter are polynomial-time dynamic programs working on deterministic machine job scheduling problems. The ideas underlying the dynamic programs are not new, for instance Potts & Kovalyov (2000) review similar dynamic programs for scheduling with batching, but the algorithms are believed to be novel. Three machine scheduling models of the sequencing problem are made. These have been developed to land aircraft based on the problem definition in Section 2.3.

3.1.1 K-Stacks model

Model When aircraft are placed into orbital holding patterns the aircraft leave their hold First-In-First-Out (FIFO) (Graves 1998), so there exist precedence

constraints on the sequencing problem. If sequencing decisions are only made on aircraft stacked in holding patterns, then all aircraft may be seen to be available to land now, and so the earliest completion times, or release dates $\hat{r}_j$, are all zero. There exist deadlines $\overline{d}_j$ on the last time an aircraft may land, or a job may complete. Using the notation in Pinedo (1995) this is a $m|prec, \overline{d}_j, S_{i,j}|obj$ problem, where there are m identical parallel machines and the objective function obj may be based on $\max\{g_j(C_j)\}$ or $\sum g_j(C_j)$, where g_j is an arbitrary non-decreasing function of the completion time C_j of job j. Two important objective functions of this form related to sequencing aircraft are sequence makespan C_{MAX} (i.e. maximize utilization of runway) and sum of weighted tardiness $\sum w_j T_j$ (i.e. minimize sum of weighted delays). Dynamic programs will be presented for single and multiple machine problems with these objective functions and the more general objective function $\sum g_j(C_j)$. This model tackles an important sub-problem of Arrival Management machine job sequencing problem described in Section 2.3.2, where constraints 5 and 7 are relaxed.

Definitions The set of n jobs to be sequenced is divided into K ordered sets, with N_i $(i = 1, \ldots, K)$ in each set. The jobs in each set are ordered by precedence, i.e. job j must be sequenced in advance of job $j + 1$. Each job belongs to one of X types. Let $g(i, j)$ denote the job type of job j in job list i, where jobs are ordered by precedence. The processing time of a job of type j followed by job type k is given by $S_{j,k}$. Each job j in set i has a time by which it is due to be processed $d_{i,j}$, a deadline $\overline{d}_{i,j}$ and a weight $w_{i,j}$ indicating its importance with respect to the tardiness $T_{i,j}$.

3.1.2 The global approach

Model The second sequencing model ignores holding and subsequent precedence constraints. Knowledge available to the scheduler is earliest $\hat{r}_j$, preferred d_j and latest $\overline{d}_j$ landing times. This is a $m|\overline{d}_j, \hat{r}_j, S_{i,j}|obj$ problem where there are m identical parallel machines and the objective function obj may be based on $\max\{g_j(C_j)\}$ or $\sum g_j(C_j)$, where g_j is an arbitrary non-decreasing function of the completion time C_j of job j. Dynamic programs will be presented for the single machine problem with objectives minimize makespan C_{MAX}, and if earliest land time and preferred land time are equivalent, minimize sum of tardiness $\sum T_j$. This model is a sub-problem of the more general Arrival Management machine job sequencing problem described in Section 2.3.2, relaxing constraints 6 and 7.

Definitions Each job in the set of n jobs to be sequenced belongs to one of X types. There are N_i type i jobs. The processing time of a job type j following a job type i is given by $S_{i,j}$. The X job type sets are ordered by earliest release time so the release time for job j in ordered set i is $\hat{r}_{i,j}$. Each job j in set i also has a deadline $\overline{d}_{i,j}$ and a time that it is due to be processed $d_{i,j}$.

Lemma Jobs i, j of the same job type a have the property:

$$\hat{r}_i \leq \hat{r}_j \tag{3.1}$$

If $C(\alpha)$ is the evaluation of objective function C_{MAX} or $\sum T_j$ (where $\hat{r}_i = d_i$) for a job sequence α then $C(\sigma i \gamma j \sigma') \leq C(\sigma j \gamma i \sigma')$.

Proof Makespan Let α_l and α_f denote the job type for the last and first jobs in a sequence α. Let $\hat{r}_{\alpha_l}$ and $\hat{r}_{\alpha_f}$ denote the release time for these jobs. Now,

$$\begin{aligned} C(\sigma i) &= \max(\hat{r}_i, C(\sigma) + S_{\sigma_l,a}) \\ C(\sigma j) &= \max(\hat{r}_j, C(\sigma) + S_{\sigma_l,a}). \end{aligned}$$

Since condition (3.1) holds it follows that

$$C(\sigma i) \leq C(\sigma j).$$

$C(\sigma i \gamma)$ is determined by the earliest time the first job in γ may complete:

$$\max(\hat{r}_{\gamma_f}, C(\sigma i) + S_{a,\gamma_f}).$$

Similarly, the earliest time the first job in $C(\sigma j \gamma)$ may complete is:

$$\max(\hat{r}_{\gamma_f}, C(\sigma j) + S_{a,\gamma_f}).$$

And so:

$$C(\sigma i) \leq C(\sigma j) \Rightarrow C(\sigma i \gamma) \leq C(\sigma j \gamma).$$

Consider sequence $C(\sigma j \gamma i)$. As condition (3.1) holds and job j is already sequenced

$$C(\sigma j \gamma i) = C(\sigma j \gamma) + S_{\gamma_l,a}.$$

Now,

$$C(\sigma i \gamma j) = \max(\hat{r}_j, C(\sigma i \gamma) + S_{\gamma_l,a}).$$

As $\hat{r}_j < C(\sigma j\gamma)$ and $C(\sigma i\gamma) \leq C(\sigma j\gamma)$

$$C(\sigma i\gamma) + S_{\gamma_l,a} \leq C(\sigma j\gamma) + S_{\gamma_l,a} \quad \Rightarrow \quad C(\sigma i\gamma j) \leq C(\sigma j\gamma i).$$

$C(\sigma i\gamma j\sigma')$ is determined in a similar manner to $C(\sigma i\gamma)$ by the earliest time the first job in σ' may complete. Applying the same arguments as before
$C(\sigma i\gamma j\sigma') \leq C(\sigma j\gamma i\sigma')$.

Proof Sum of Tardiness Let $D(\sigma)$ be the makespan of a sequence σ. From the proof of C_{MAX} above we know that

$$D(\sigma j\gamma i) \quad \geq \quad D(\sigma i\gamma j).$$

From this we can say that the sum of tardiness of jobs in the σ' sequence of $\sigma i\gamma j\sigma'$ is less than or equal to the sum of tardiness of jobs in in the σ' sequence of $\sigma j\gamma i\sigma'$. Similarly,

$$D(\sigma j\gamma) \quad \geq \quad D(\sigma i\gamma),$$

and the sum of tardiness for jobs in the γ part of sequence in $\sigma i\gamma$ is less than or equal the sum of tardiness for the γ part of the $\sigma j\gamma$ sequence.
The decrease in the tardiness of job j found by choosing sequence $\sigma j\gamma\ i$ instead of $\sigma i\gamma\ j$ is

$$\max(D(\sigma i\gamma) + S_{\gamma_l,a}, \hat{r}_j) \quad - \quad \max(D(\sigma) + S_{\sigma_l,a}, \hat{r}_j). \tag{3.2}$$

The increase in tardiness for job i in sequence $\sigma j\gamma\ i$ over $\sigma i\gamma\ j$ is

$$D(\sigma j\gamma) + S_{\gamma_l,a} \quad - \quad \max(D(\sigma) + S_{\sigma_l,a}, \hat{r}_i). \tag{3.3}$$

Now, Equation (3.2) - Equation (3.3) ≤ 0 so it follows that switching i and j can only have the effect that

$$C(\sigma i\gamma j\sigma') \quad \leq \quad C(\sigma j\gamma i\sigma').$$

3.1.3 Approach stream model

Approach Streams The final sequencing model does not allow change to be made to the FCFS sequence aircraft fly into IAF points through precedence constraints. One benefit of this is that aircraft that fly to the same IAF generally fly similar routes into the airport and so problematic overtakes between aircraft with similar routes are avoided. Each aircraft j has an earliest land time $\hat{r}_j$ and a latest $\overline{d}_j$. This is a $m|prec, \overline{d}_j, \hat{r}_j, S_{i,j}|obj$ problem where there are m identical parallel machines

and the objective function obj may be based on $\max\{g_j(C_j)\}$ or $\sum g_j(C_j)$, where g_j is an arbitrary non-decreasing function of the completion time C_j of job j. Dynamic programs will be presented for single and multiple machines for two important objective functions related to sequencing aircraft - sequence makespan C_{MAX} and sum of weighted tardiness $\sum w_j T_j$, and a general non-decreasing objective function. This model tackles a sub-problem of the more general Arrival Management machine job sequencing problem described in Section 2.3.2, relaxing constraint 7.

Definitions The set of n jobs to be sequenced is divided into K ordered sets, with N_i $(i = 1, \ldots, K)$ in each set. The jobs in each set are ordered by precedence, i.e. job j must be sequenced in advance of job $j + 1$. Each job belongs to one of X types. The processing time of a job of type j followed by job type k is given by $S_{j,k}$. Each job j in set i has a time that it is due to be processed $d_{i,j}$, an earliest completion time $\hat{r}_{i,j}$, a deadline $\overline{d}_{i,j}$ and a weight $w_{i,j}$ indicating its importance with respect to the tardiness $T_{i,j}$.

3.2 Dynamic program formulations for the K-Stacks problem

The dynamic programs in this section use the precedence order of jobs to find optimal solutions. Feasible sequences must have jobs in precedence list order, so jobs are added to the dynamic programs in their list order. By enumerating all the possibilities the dynamic programs are able to find optimal sequences.

3.2.1 Dynamic program for $1|prec, S_{i,j}|C_{MAX}$

State variables h_i are used to keep track of the number of jobs added from each precedence list i. Basic definitions are found in Section 3.1.1. Define $f(h_1, h_2, \ldots, h_K, l)$ to be the minimum makespan of a schedule with the first h_i jobs from list i already scheduled and l the last job's list number for $i = 1, \ldots, K$.

Initialization

$$\begin{aligned} f(1, 0, \ldots, 0, 1) &= 0 \\ f(0, 1, \ldots, 0, 2) &= 0 \\ &\vdots \\ f(0, 0, \ldots, 1, K) &= 0. \end{aligned}$$

Recursion For $h_i = 0, 1, \ldots, N_i \qquad i = 1, 2, \ldots, K \qquad l = 1, 2, \ldots, K$

$$f(h_1, h_2, \ldots, h_K, l) = \min_{l'=1,2,\ldots,K} \left\{ f(h'_1, h'_2, \ldots, h'_K, l') + S_{g(l',h'_{l'}),g(l,h_l)} \right\}$$

where

$$h'_i = \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l. \end{cases}$$

Optimum

$$\min_{l=1,2,\ldots,K} \quad f(N_1, N_2, \ldots, N_K, l).$$

3.2.2 Dynamic program for $1|prec, \bar{d}_j, S_{i,j}|C_{MAX}$

The dynamic program in Section 3.2.1 can be modified for deadlines, by setting the function value to infinity for the states where the job will be processed after its deadline. The recursion equation becomes:

$$f(h_1, h_2, \ldots, h_K, l) = \begin{cases} \min_{l'=1,2,\ldots,K} \left\{ f(h'_1, h'_2, \ldots, h'_K, l') + S_{g(l',h'_{l'}),g(l,h_l)} \right\} & \text{if } \leq \bar{d}_{l,h_l} \\ \infty & \text{otherwise.} \end{cases}$$

3.2.3 Dynamic program for $1|prec, S_{i,j}|\sum w_j T_j$

This dynamic program uses state variables $y_{j,k}$ to record the number of type k jobs following type j. These variables permit calculation of the makespan of each state, and thus calculation of the contribution to weighted tardiness from the job added. Basic definitions are found in Section 3.1.1. Define $f(h_1, h_2, \ldots, h_K, y_{1,1}, y_{1,2}, \ldots, y_{X,X}, l)$ as the minimum weighted tardiness for a schedule with h_i jobs sequenced from list i, $y_{j,k}$ job type k following job type j, with l the last list a job was sequenced from. Let all the variables in a state denoted $(h'_1, \ldots, l')$ have a prime on them.

Initialization

$$\begin{aligned} f(1, 0, \ldots, 0, 0, \ldots, 0, 1) &= 0 \\ f(0, 1, \ldots, 0, 0, \ldots, 0, 2) &= 0 \\ &\vdots \\ f(0, 0, \ldots, 1, 0, \ldots, 0, K) &= 0. \end{aligned}$$

Recursion For $h_i = 0, \ldots, N_i \quad i = 1, \ldots, K \quad j = 1, 2, \ldots, X$ $k = 1, 2, \ldots, X \quad l = 1, 2, \ldots, K$ such that $h_{l'} > 0$ and $y_{g(l',h'_{l'}),g(l,h_l)} > 0$

$$f(h_1, h_2, \ldots, h_K, y_{1,1}, y_{1,2}, \ldots, y_{X,X}, l) =$$
$$\min_{l'=1,\ldots,K} \left\{ f(h'_1, \ldots, l') + w_{l,h_l} \left(\max \left\{ \sum_{j=1}^{X} \sum_{k=1}^{X} y_{j,k} S_{j,k} - d_{l,h_l}, 0 \right\} \right) \right\}$$

where

$$h'_i = \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l \end{cases}$$

$$y'_{j,k} = \begin{cases} y'_{j,k} - 1 & \text{if } j = g(l', h'_{l'}) \text{ and } k = g(l, h_l) \\ y_{j,k} & \text{otherwise.} \end{cases}$$

Optimum

$$\min_{y_{j,k},l} \; f(N_1, N_2, \ldots, N_K, y_{1,1}, y_{1,2}, \ldots, y_{X,X}, l)$$

where for $j = 1, \ldots, X$, $k = 1, \ldots, X$; $y_{j,k} = 1, \ldots, X$ and $l = 1, \ldots, K$.

3.2.4 Dynamic program for $1|prec, S_{i,j}|\sum g_j(C_j)$

The formulation of the dynamic program for $1|prec, S_{i,j}|\sum w_j T_j$ may be generalized for any objective function $\sum g_j(C_j)$, where g_j is an arbitrary non-decreasing function of the completion time C_j of job j. Job index j may be deduced from l and h_l. The recursion equation becomes:

$$f(h_1, \ldots, l) = \min_{l'=1,2,\ldots,K} \left\{ f(h'_1, \ldots, l') + g_{l,h_l} \left(\sum_{j=1}^{X} \sum_{k=1}^{X} y_{j,k} S_{j,k} \right) \right\}.$$

3.2.5 Dynamic program for $1|prec, \bar{d}_j, S_{i,j}|\sum g_j(C_j)$

Deadline constraints can be included in the formulation by setting the function f to infinity for this case. That is, the recursion equation for state $(h_1, h_2, \ldots, h_K, y_{1,1}, y_{1,2}, \ldots, y_{X,X}, l)$ if $\sum_j \sum_k y_{j,k} S_{j,k} > \bar{d}_{l,h_l}$ is:

$$f(h_1, h_2, \ldots, h_K, y_{1,1}, y_{1,2}, \ldots, y_{X,X}, l) = \infty.$$

3.2.6 Dynamic program for $m|prec, S_{i,j}|\sum w_j T_j$

A dynamic program for the multiple machine problem may also be developed, building on ideas from Section 3.2.3. Suppose there are m machines. A state variable $y_{z,j,k}$ is used to store number of type k jobs following type j jobs on machine z. A further m variables v_z are used to record the last job type on each machine z; this information cannot be deduced from the last list l sequenced when there is more than 1 runway. Basic definitions are found in Section 3.1.1. Let $f(h_1, h_2, \ldots, h_K, y_{1,1,1}, y_{1,1,2}, \ldots, y_{1,X,X}, y_{2,1,1}, \ldots, y_{m,X,X}, v_1, v_2, \ldots, v_m, u, l)$ be the minimum weighted tardiness for a schedule with h_i jobs sequenced from list i, $y_{z,j,k}$ the number of job type k following job type j on machine z, v_z the last job type sequenced on machine z, with l the last list a job was sequenced from and u the last machine added to. Define $v_z = 0$ when no job is scheduled on machine z. Set $v_u = g(l, h_l)$. Then a dynamic program may be formulated as follows.

Initialization For $i = 1, \ldots, K$ and $j = 1, \ldots, K$

$$h_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

and for $u = 1, \ldots, m$ and $z = 1, \ldots, m$

$$v_z = \begin{cases} g(i,1) & \text{if } u = z \\ 0 & \text{otherwise} \end{cases}$$

$$f(h_1, h_2, \ldots, h_K, 0, 0, \ldots, 0, v_1, \ldots, v_m, u, i) = 0.$$

Recursion For $h_i = 0, \ldots, N_i$ $i = 1, \ldots, K$ $j = 1, \ldots, X$ $k = 1, \ldots, X$ $z = 1, \ldots, m$ $u = 1, \ldots, m$ $l = 1, \ldots, K$

$$f(h_1, h_2, \ldots, h_K, y_{1,1,1}, y_{1,1,2}, \ldots, y_{m,X,X}, v_1, \ldots, v_m, u, l) = \min_{l', v'_u, u'} \left\{ f(h'_1 \ldots, l') + w_{l,h_l} \left(\max \left\{ \sum_{j=1}^{X} \sum_{k=1}^{X} y_{u,j,k} S_{j,k} - d_{l,h_l}, 0 \right\} \right) \right\}$$

where $l' = 1, \ldots, K$, $v'_u = 0, \ldots, X$, $u' = 1, \ldots, m$ and

$$h'_i = \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l \end{cases}$$

$$y'_{z,j,k} = \begin{cases} y_{z,j,k} - 1 & \text{if } v'_u \neq 0 \text{ and } z = u \text{ and } j = v'_u \text{ and } k = v_u \\ y_{z,j,k} & \text{otherwise} \end{cases}$$
$$v'_z = v_z \text{ if } z \neq u \ .$$

Optimum

$$\min_{y_{z,j,k}, v_z, u, l} f(N_1, N_2, \ldots, N_K, y_{1,1,1}, y_{1,1,2}, \ldots, y_{m,X,X}, v_1, \ldots, v_m, u, l)$$

where for $z = 1, \ldots, m$, $j = 1, \ldots, X$, $k = 1, \ldots, X$; $y_{z,j,k} = 1, \ldots, X$, $v_z = 0, \ldots, X$, $u = 1, \ldots, m$ and $l = 1, \ldots, K$.

3.2.7 Dynamic program for $m|prec, S_{i,j}|\sum g_j(C_j)$

The DP formulation for the $m|prec, S_{i,j}|\sum w_j T_j$ problem may be generalized for any objective function $\sum g_j(C_j)$, where g_j is an arbitrary non-decreasing function of the completion time C_j of job j. The recursion equation becomes:

$$f(h_1 \ldots, u, l) = \min_{l', v'_u, u'} \left\{ f(h'_1 \ldots, u', l') + g_{l,h_l} \left(\sum_{j=1}^{X} \sum_{k=1}^{X} y_{u,j,k} S_{j,k} \right) \right\}$$

where $l' = 1, \ldots, K$, $v'_u = 0, \ldots, X$ and $u' = 1, \ldots, m$.

3.2.8 Dynamic program for $m|prec, \overline{d}_j, S_{i,j}|\sum g_j(C_j)$

Deadline constraints can also be included in the problem. That is, for $u = 1, \ldots, m$ the recursion equation for state $(h_1, h_2, \ldots, h_K, y_{1,1,1}, y_{1,1,2}, \ldots, y_{m,X,X}, v_1, \ldots, v_m, u, l)$ if $\sum_j \sum_k y_{u,j,k} S_{j,k} > \overline{d}_{l,h_l}$ is:

$$f(h_1, h_2, \ldots, h_K, y_{1,1,1}, y_{1,1,2}, \ldots, y_{m,X,X}, v_1, \ldots, v_m, u, l) = \infty.$$

3.2.9 Computational complexity

Assume that the number of machines m, and the number of ordered sets K are fixed. The dynamic programs for $1|prec, S_{i,j}|C_{MAX}$ and $1|prec, \overline{d}_j, S_{i,j}|C_{MAX}$ have at most n stages. Each stage has at most n^{K-1} states (since the state variables h_i have K-1 degrees of freedom). Therefore computational complexity is at most $O(n^K)$. The dynamic programs for $1|prec, S_{i,j}|\sum g_j(C_j)$ and $1|prec, \overline{d}_j, S_{i,j}|\sum g_j(C_j)$ have at most n stages. Again, each stage has at most n^{K-1} combinations of the h_i variables. There are at most n^{X^2} combinations of the $y_{j,k}$ variables. So computational complexity is at most $O(n^{K+X^2})$. The dynamic programs for $m|prec, \overline{d}_j, S_{i,j}|\sum g_j(C_j)$ and $m|prec, S_{i,j}|\sum g_j(C_j)$ have at most n stages. Each

stage has at most n^{K-1} combinations of the h_i variables, and n^{mX^2} of the $y_{z,j,k}$ variables, so has total complexity bounded by $O(n^{K+mX^2})$.

3.3 Dynamic program formulations for the global approach

The lemma in Section 3.1.2 shows that optimal sequences, for the problems in this section, must have jobs of the same type ordered by earliest release date. Dynamic programs are developed to make use of this property. By enumerating all the possible sequences with this property, optimal sequences are found.

3.3.1 Dynamic program for $1|\hat{r}_j, S_{i,j}|C_{MAX}$

The lemma in Section 3.1.2 shows that an optimal sequence for this problem will have jobs of the same type ordered by release date. Jobs are added to the dynamic program in job-type Earliest Release Date (ERD) order. State variables v_i for $i = 1, \ldots, X$ are used to record the number of type i jobs sequenced. Basic definitions are found in Section 3.1.2. Let $f(v_1, v_2, \ldots, v_X, l)$ be the minimum makespan for a schedule with v_i type i jobs in place, and l the last job type.

Initialization

$$\begin{aligned} f(1,0,\ldots,0,1) &= \hat{r}_{1,1} \\ f(0,1,\ldots,0,2) &= \hat{r}_{1,2} \\ &\vdots \\ f(0,0,\ldots,1,X) &= \hat{r}_{1,X}. \end{aligned}$$

Recursion For $\quad v_i = 0,1,\ldots,N_i \quad i = 1,2,\ldots,X \quad l = 1,2,\ldots,X$

$$f(v_1, v_2, \ldots, v_X, l) = \min_{l'=1,2,\ldots,X} \{\max\{f(v'_1, v'_2, \ldots, v'_X, l') + S_{l',l}, \hat{r}_{l,v_l}\}\}$$

where

$$v'_i = \begin{cases} v_i - 1 & \text{if } i = l \\ v_i & \text{if } i \neq l. \end{cases}$$

Optimum

$$\min_{l=1.2,\ldots,X} f(N_1, N_2, \ldots, N_X, l).$$

3.3.2 Dynamic program for special case of $1|\overline{d}_j, \hat{r}_j, S_{i,j}|C_{MAX}$

Lemma If $\hat{r}_i \leq \hat{r}_j \Rightarrow \overline{d}_i \leq \overline{d}_j \forall i, j$ then an optimal sequence for makespan C_{MAX} and sum of tardiness $\sum T_j$ when $d_i = \hat{r}_i$ has jobs of the same type sequenced by Earliest Release Date (ERD).

Proof Denote $C(\alpha)$ as the objective value (either $\sum T_j$ or C_{MAX}) of a sequence α. Switching the two jobs i and j of the same type in a sequence $C(\sigma i \alpha j \sigma')$ cannot produce a better sequence (see Section 3.1.2). If $\sigma i \alpha j \sigma'$ is not feasible because job i fails to meet its deadline $\overline{d}_i$ then $\sigma j \alpha i \sigma'$ is not feasible either since job i will be scheduled later in this sequence. If $\sigma i \alpha j \sigma'$ is not feasible because job j does not meet its deadline $\overline{d}_j$ then the sequence $\sigma j \alpha i \sigma'$ will not be feasible either, since $\overline{d}_i \leq \overline{d}_j$. Therefore, the ERD ordering within job class holds.

Using this result the recursion equations from the dynamic program in Section 3.3.1 are altered to:

$$f(v_1, v_2, \ldots, v_X, l) = \min_{l'=1,2,\ldots,X} \begin{cases} (\max\{f(v'_1, v'_2, \ldots, v'_X, l') + S_{l',l}, \hat{r}_{l,v_l}\}) & \text{if } \leq \overline{d}_{l,v_l} \\ \infty & \text{otherwise.} \end{cases}$$

3.3.3 Dynamic program for $1|d_j = \hat{r}_j, S_{i,j}|\sum T_j$

The lemma in Section 3.1.2 shows that an optimal sequence for this problem has jobs of the same type ordered by release date, so jobs of the same type are added to the dynamic program in release date order. To calculate the tardiness of the last job added to a partial sequence, the makespan of the sequence at that point must be known. Makespan is calculated in the dynamic program using a state variable (a, b) to record the release date of the last job b of type a that came after an idle time period, and preceded a sequence of $y_{i,j}$ number of type j jobs following type i. Basic definitions for the dynamic program are found in Section 3.1.2. Denote $f(v_1, \ldots, v_X, y_{1,1}, \ldots, y_{X,X}, (a, b), l)$ as the minimum sum of tardiness for a schedule with v_i type i jobs sequenced, the last job in the sequence of type l. Assuming that $\hat{r}_{i,j} = d_{i,j}$ and that jobs are ordered by release date in the job type sets we can formulate a dynamic program as follows.

Initialization For $i = 1, 2, \ldots, X$, $j = 1, 2, \ldots, X$

$$v_j = \begin{cases} 1 & \text{if } j = i \\ 0 & \forall j \neq i \end{cases}$$

$$f(v_1, v_2, \ldots, v_X, 0, 0, \ldots, 0, (i, 1), i) \;=\; 0.$$

Recursion

For $v_i = 0, \ldots, V_i \quad i = 1, 2, \ldots, X \quad j = 1, 2, \ldots, X \quad (a, b) = 1, \ldots, n \quad l = 1, \ldots, X$
$y_{i,j} = 0, \ldots, X$

If $\sum_i \sum_j y_{i,j} = 0$ then:

$$f(v_1, \ldots, v_X, y_{1,1}, \ldots, y_{X,X}, (a, b), l) =$$
$$\min_{l', (a', b'), y'_{i,j}} \begin{cases} f(v'_1, \ldots, l') & (a, b) = (l, v_l) \text{ and } \hat{r}_{a',b'} + \sum_j \sum_k y'_{j,k} S_{j,k} \leq \hat{r}_{l,v_l} \\ \infty & \text{otherwise} \end{cases}$$

where $l' = 1, \ldots, X$, $(a', b') = 1, \ldots, n$, $y'_{i,j} = 0, \ldots, X$ and

$$v'_i \;=\; \begin{cases} v_i - 1 & \text{if } i = l \\ v_i & \text{if } i \neq l \end{cases}$$

Else:

$$f(v_1, \ldots, v_X, y_{1,1}, \ldots, y_{X,X}, (a, b), l) =$$
$$\min_{l', (a', b')} \begin{cases} f(v'_1, \ldots, l') + \max\left\{\hat{r}_{a,b} + \sum_i \sum_j y_{i,j} S_{i,j} - d_{l,v_l}, 0\right\} & \text{(a)} \\ \infty & \text{(b)} \end{cases}$$

Condition (a):

$$\hat{r}_{a,b} + \sum_{i=1}^{X} \sum_{j=1}^{X} y_{i,j} S_{i,j} \;\geq\; \hat{r}_{l,v_l}$$

Condition (b):

$$\hat{r}_{a,b} + \sum_{i=1}^{X} \sum_{j=1}^{X} y_{i,j} S_{i,j} \;<\; \hat{r}_{l,v_l}$$

where $l' = 1, \ldots, X$, $(a', b') = 1, \ldots, n$ and

$$v'_i = \begin{cases} v_i - 1 & \text{if } i = l \\ v_i & \text{if } i \neq l \end{cases}$$

$$y'_{i,j} = \begin{cases} y_{i,j} - 1 & \text{if } i = l' \text{ and } j = l \\ y_{i,j} & \text{otherwise.} \end{cases}$$

Optimum

$$\min_{y_{i,j},(a,b),l} f(V_1, V_2, \ldots, V_X, y_{1,1}, \ldots, y_{X,X}, (a,b), l)$$

where for $i = 1, \ldots, X$, $j = 1, \ldots, X$; $y_{i,j} = 1, \ldots, X$, $(a,b) = 1, \ldots, n$ and $l = 1, \ldots, X$.

3.3.4 Dynamic program for special case of $1|d_j = \hat{r}_j, \overline{d}_j, S_{i,j}|\sum T_j$

If $\hat{r}_i \leq \hat{r}_j \Rightarrow \overline{d}_i \leq \overline{d}_j \; \forall i, j$ then in an optimal sequence jobs of the same type are ordered ERD by the lemma in Section 3.3.2. We alter the recursion equation as follows:

$$f(v_1, \ldots, v_X, y_{1,1}, \ldots, y_{X,X}, (a,b), l) =$$

$$\min_{l',(a',b')} \begin{cases} f(v'_1, \ldots, l') + \max\left(\hat{r}_{a,b} + \sum_i \sum_j y_{i,j} S_{i,j} - d_{l,v_l}, 0\right) & \text{(a)} \\ \infty & \text{otherwise} \end{cases}$$

Condition (a):

$$\hat{r}_{l,v_l} \leq \hat{r}_{a,b} + \sum_{i=1}^{X} \sum_{j=1}^{X} y_{i,j} S_{i.j} \leq \overline{d}_{l,v_l}$$

where $l' = 1, \ldots, X$ and $(a', b') = 1, \ldots, X$.

3.3.5 Computational complexity

Assume that the number of machines m, and the number of job types X are fixed. The dynamic programs for $1|\hat{r}_j = d_j, S_{i,j}|C_{MAX}$ and the special case of $1|\hat{r}_j = d_j, \overline{d}_j, S_{i,j}|C_{MAX}$ have n stages. Each stage is generated by at most n^{X-1} combinations of state variables. The computational complexity is therefore at most $O(n^X)$. The dynamic programs for $1|\hat{r}_j = d_j, S_{i,j}|\sum T_j$ and the special case of $1|\hat{r}_j = d_j, \overline{d}_i, S_{i,j}|\sum T_j$ have X^2 number of $y_{i,j}$ variables, X v_i variables and a single variable (a,b), so complexity is $O(n^{X+X^2+1})$, and is thus polynomial in the number of jobs.

3.4 Dynamic programs for the approach stream approach

The following dynamic programs are based on priority constraints into IAF points. Feasible sequences must have jobs in precedence list order, so jobs are added to the dynamic programs in their list order. By enumerating all the possibilities the dynamic programs are able find optimal sequences.

3.4.1 Dynamic program for $1|prec, \hat{r}_j, S_{i,j}|C_{MAX}$

Jobs are ordered by precedence constraints. The same DP as for $1|\hat{r}_j, S_{i,j}|C_{MAX}$ will work for this problem if we define the v_i in the formulation in Section 3.3.1 to have the same meaning as the h_i from the formulation in Section 3.2.1.

3.4.2 Dynamic program for $1|prec, \overline{d}_j, \hat{r}_j, S_{i,j}|C_{MAX}$

The dynamic program for special case of $1|\overline{d}_j, \hat{r}_j, S_{i,j}|C_{MAX}$ in Section 3.3.2 will apply to this problem, if the v_i variables in the formulation are defined to have the same meaning as the h_i from the formulation in Section 3.2.1, and a suitable $g(i, j)$ function to get the type of job j in list i is used. The DP will work for any set of deadlines since jobs are added by precedence, not by the ERD rule.

3.4.3 Dynamic program for $1|prec, \hat{r}_j, S_{i,j}|\sum w_j T_j$

Basic definitions are found in Section 3.1.3. Let $f(h_1, \ldots, h_K, y_{1,1}, \ldots, y_{X,X}, (a, b), l))$ denote the minimum weighted tardiness for a sequence with h_i jobs from list i sequenced, the last job in the sequence from list l. Let $g(i, j)$ denote the job type of job j in job list i, where jobs are ordered by their release date. The ideas from Section 3.3.3 are used in this dynamic program; the sequence length may be found from (a, b), the last job b from precedence list a that precedes a sequence of $y_{j,k}$ number of type k jobs following type j.

Initialization For $i = 1, 2, \ldots, K$, $j = 1, \ldots, K$

$$h_j = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases}$$

$$f(h_1, h_2, \ldots, h_K, 0, 0, \ldots, 0, (i, 1), i) = 0.$$

Recursion For $h_i = 0, \ldots, N_i$ $i = 1, \ldots, K$ $j = 1, \ldots, X$ $k = 1, \ldots, X$ $(a, b) = 1, \ldots, n$ $l = 1, \ldots, K$ $y_{j,k} = 0, \ldots, X$

If $\sum_j \sum_k y_{j,k} = 0$ then:

$$f(h_1, \ldots, h_K, y_{1,1}, \ldots, y_{X,X}, (a,b), l) =$$
$$\min_{l',(a',b'),y'_{i,j}} \begin{cases} f(h'_1, \ldots, l') & (a,b) = (l, h_l) \text{ and } \hat{r}_{a',b'} + \sum_j \sum_k y'_{j,k} S_{j,k} \leq \hat{r}_{l,h_l} \\ \infty & \text{otherwise} \end{cases}$$

where $l' = 1, \ldots, X$, $(a', b') = 1, \ldots, n$, $y'_{j,k} = 0, \ldots, X$ and

$$h'_i = \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l \end{cases}$$

Else:

$$f(h_1, \ldots, h_K, y_{1,1}, \ldots, y_{X,X}, (a,b), l) =$$
$$\min_{l',(a',b')} \begin{cases} f(h'_1, \ldots, l') + \max\left\{w_{l,h_l}\left(\hat{r}_{a,b} + \sum_j \sum_k y_{j,k} S_{j,k} - d_{l,h_l}\right), 0\right\} & \text{(a)} \\ \infty & \text{(b)} \end{cases}$$

Condition (a):

$$\hat{r}_{a,b} + \sum_{j=1}^{X} \sum_{k=1}^{X} y_{j,k} S_{j,k} \geq \hat{r}_{l,h_l}$$

Condition (b):

$$\hat{r}_{a,b} + \sum_{j=1}^{X} \sum_{k=1}^{X} y_{j,k} S_{j,k} < \hat{r}_{l,h_l}$$

where $l' = 1, \ldots, X$, $(a', b') = 1, \ldots, n$ and

$$h'_i = \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l \end{cases}$$
$$y'_{j,k} = \begin{cases} y'_{j,k} - 1 & \text{if } j = g(l', h'_{l'}) \text{ and } k = g(l, h_l) \\ y_{j,k} & \text{otherwise.} \end{cases}$$

Optimum

$$\min_{y_{j,k},(a,b),l} f(N_1, N_2, \ldots, N_K, y_{1,1}, \ldots, y_{X,X}, (a,b), l)$$

where for $j = 1, \ldots, X$, $k = 1, \ldots, X$; $y_{j,k} = 1, \ldots, X$, $(a,b) = 1, \ldots, n$ and $l = 1, \ldots, K$.

3.4.4 Dynamic program for $1|prec, \hat{r}_j, S_{i,j}|\sum g_j(C_j)$

The above DP can be generalized to find an optimal sequence for any objective function $\sum g_j(C_j)$, where g_j is an arbitrary non-decreasing function of the completion time C_j of job j. The first recursion condition remains unchanged. The second recursion condition is set up:

$$f(h_1, \ldots, h_K, y_{1,1}, \ldots, y_{X,X}, (a,b), l) =$$
$$\min_{l',(a',b')} \begin{cases} f(h'_1, \ldots, l') + g_{l,h_l}\left(\hat{r}_{a,b} + \sum_j \sum_k y_{j,k} S_{j,k}\right) & \hat{r}_{a,b} + \sum_{j=1}^{X} \sum_{k=1}^{X} y_{j,k} S_{j,k} \geq \hat{r}_{l,h_l} \\ \infty & \text{otherwise} \end{cases}$$

where $l' = 1, \ldots, X$, $(a', b') = 1, \ldots, n$.

3.4.5 Dynamic program for $1|prec, \overline{d}_j, \hat{r}_j, S_{i,j}|\sum g_j(C_j)$

The recursion equation above is altered to include an extra condition that sets to infinity states that are generated with last job completion time after its deadline. That is, if $\hat{r}_{l,h_l} + \sum_j \sum_k y_{j,k} S_{j,k} > \overline{d}_{l,h_l}$ for state $(h_1, \ldots, h_K, y_{1,1}, \ldots, y_{X,X}, (a,b), l)$ then:

$$f(h_1, \ldots, h_K, y_{1,1}, \ldots, y_{X,X}, (a,b), l) \quad = \quad \infty.$$

3.4.6 Dynamic program for $m|prec, \hat{r}_j, S_{i,j}|C_{MAX}$

The following dynamic programs for multiple machine problems make use of ideas introduced in Section 3.3.3 and Section 3.2.6. Basic definitions for the dynamic program are found in Section 3.1.3. Let $f(h_1, \ldots, h_K, y_{1,1,1}, \ldots, y_{m,X,X}, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, l)$ be the minimum makespan for a partial sequence with h_i jobs scheduled from precedence list i, l the last list sequenced from, u the last machine added to, (a_z, b_z) the last job b_z from precedence list a_z to precede a sequence of $y_{z,j,k}$ type k jobs following type j jobs on machine z without a gap in the minimum separations, and v_z be the last job type processed on machine z. Let $g(i,j)$ denote the job type of job j in job list i, where jobs are ordered by their release date. Let indices $(a_z, b_z) = (0,0)$ and $v_z = 0$ denote no job scheduled on machine z and define $\hat{r}_{0,0} = 0$. Set $v_u = g(l, h_l)$. Then a dynamic program may be formulated as follows.

Initialization For $i = 1, \ldots, K$ and $j = 1, \ldots, K$

$$h_j \quad = \quad \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

and for $u = 1, \ldots, m$ and $z = 1, \ldots, m$

$$v_z = \begin{cases} g(i,1) & \text{if } u = z \\ 0 & \text{otherwise} \end{cases}$$

$$(a_z, b_z) = \begin{cases} (i,1) & \text{if } u = z \\ (0,0) & \text{otherwise} \end{cases}$$

$$f(h_1, h_2, \ldots, h_K, 0, 0, \ldots, 0, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, i) = \hat{r}_{i,1}.$$

Recursion For $h_i = 0, \ldots, N_i \quad i = 1, \ldots, K \quad j = 1, \ldots, X$
$k = 1, \ldots, X \quad z = 1, \ldots, m \quad (a_u, b_u) = 0, \ldots, n \quad u = 1, \ldots, m$
$l = 1, \ldots, K \quad y_{z,j,k} = 0, \ldots, X$
If $\sum_j \sum_k y_{u,j,k} = 0$ then:

$$f(h_1, \ldots, h_K, y_{1,1,1}, \ldots, y_{m,X,X}, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, l) =$$
$$\min_{l', y'_{u,j,k}, v'_u, u', (a'_u, b'_u)} \begin{cases} \max\{f(h'_1, \ldots, l'), \hat{r}_{a_u, b_u}\} & (a_u, b_u) = (l, h_l) \text{ and} \\ & \hat{r}_{a'_u, b'_u} + \sum_j \sum_k y'_{u,j,k} S_{j,k} \leq \hat{r}_{l,h_l} \\ \infty & \text{otherwise} \end{cases}$$

where $l' = 1, \ldots, K$, $v'_u = 0, \ldots, X$, $u' = 1, \ldots, m$, $(a'_u, b'_u) = 0, \ldots, n$ and

$$h'_i = \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l \end{cases}$$

$$y'_{z,j,k} = \begin{cases} 0, \ldots, X & \text{if } z = u \\ y_{z,j,k} & \text{otherwise} \end{cases}$$

$$v'_z = v_z \quad \text{if } z \neq u$$

$$(a'_z, b'_z) = (a_z, b_z) \quad \text{if } z \neq u$$

Else:

$$f(h_1, \ldots, h_K, y_{1,1,1}, \ldots, y_{m,X,X}, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, l) =$$
$$\min_{l', v'_u, u', (a'_u, b'_u)} \begin{cases} \max\left\{f(h'_1, \ldots, l'), \hat{r}_{a_u, b_u} + \sum_j \sum_k y_{u,j,k} S_{j,k}\right\} & \text{(a)} \\ \infty & \text{(b)} \end{cases}$$

Condition (a):

$$\hat{r}_{a_u, b_u} + \sum_{j=1}^{X} \sum_{k=1}^{X} y_{u,j,k} S_{j,k} \geq \hat{r}_{l,h_l}$$

Condition (b):

$$\hat{r}_{a_u,b_u} + \sum_{j=1}^{X}\sum_{k=1}^{X} y_{u,j,k} S_{j,k} < \hat{r}_{l,h_l}$$

where $l' = 1,\ldots,K$, $v'_u = 0,\ldots,X$, $u' = 1,\ldots,m$, $(a'_u, b'_u) = 0,\ldots,n$ and

$$\begin{aligned}
h'_i &= \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l \end{cases} \\
y'_{z,j,k} &= \begin{cases} y_{z,j,k} - 1 & \text{if } v'_u \neq 0 \text{ and } \quad z = u \text{ and } j = v'_u \text{ and } k = v_u \\ y_{z,j,k} & \text{otherwise} \end{cases} \\
v'_z &= v_z \quad \text{if } z \neq u \\
(a'_z, b'_z) &= (a_z, b_z) \quad \text{if } z \neq u \ .
\end{aligned}$$

Optimum

$$\min_{y_{z,j,k},(a_z,b_z),v_z,u,l} f(N_1,\ldots,N_K, y_{1,1,1},\ldots,(a_1,b_1),\ldots,v_1,\ldots,u,l)$$

where for $z = 1,\ldots,m$, $j = 1,\ldots,X$, $k = 1,\ldots,X$; $y_{z,j,k} = 1,\ldots,X$, $(a_z, b_z) = 1,\ldots,n$, $v_z = 1,\ldots,X$, $u = 1,\ldots,m$ and $l = 1,\ldots,K$.

3.4.7 Dynamic program for $m|prec, \hat{r}_j, S_{i,j}|\sum w_j T_j$

Basic definitions are found in Section 3.1.3. Let $f(h_1,\ldots,h_K, y_{1,1,1},\ldots,y_{m,X,X},(a_1,b_1),\ldots,(a_m,b_m),v_1,\ldots,v_m,u,l)$ be the minimum weighted tardiness for a partial sequence with definitions of state variables h_i, u, (a_z, b_z), $y_{z,j,k}$ and v_z as the previous DP in Section 3.4.6. Then a dynamic program may be formulated as follows.

Initialization For $i = 1,\ldots,K$ and $j = 1,\ldots,K$

$$h_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

and for $u = 1,\ldots,m$ and $z = 1,\ldots,m$

$$\begin{aligned}
v_z &= \begin{cases} g(i,1) & \text{if } u = z \\ 0 & \text{otherwise} \end{cases} \\
(a_z, b_z) &= \begin{cases} (i,1) & \text{if } u = z \\ (0,0) & \text{otherwise} \end{cases}
\end{aligned}$$

$$f(h_1, h_2, \ldots, h_K, 0, 0, \ldots, 0, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, i) \quad = \quad 0.$$

Recursion For $h_i = 0, \ldots, N_i \quad i = 1, \ldots, K \quad j = 1, \ldots, X$
$k = 1, \ldots, X \quad z = 1, \ldots, m \quad (a_u, b_u) = 0, \ldots, n \quad u = 1, \ldots, m$
$l = 1, \ldots, K \quad y_{z,j,k} = 0, \ldots, X$
If $\sum_j \sum_k y_{u,j,k} = 0$ then:

$$f(h_1, \ldots, h_K, y_{1,1,1}, \ldots, y_{m,X,X}, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, l) =$$
$$\min_{l', y'_{u,j,k}, v'_u, u', (a'_u, b'_u)} \begin{cases} f(h'_1, \ldots, l') & (a_u, b_u) = (l, h_l) \text{ and } \hat{r}_{a'_u, b'_u} + \sum_j \sum_k y'_{u,j,k} S_{j,k} \leq \hat{r}_{l,h_l} \\ \infty & \text{otherwise} \end{cases}$$

where $l' = 1, \ldots, K$, $v'_u = 0, \ldots, X$, $u' = 1, \ldots, m$, $(a'_u, b'_u) = 0, \ldots, n$ and

$$\begin{aligned} h'_i &= \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l \end{cases} \\ y'_{z,j,k} &= \begin{cases} 0, \ldots, X & \text{if } z = u \\ y_{z,j,k} & \text{otherwise} \end{cases} \\ v'_z &= v_z \quad \text{if } z \neq u \\ (a'_z, b'_z) &= (a_z, b_z) \quad \text{if } z \neq u \end{aligned}$$

Else:

$$f(h_1, \ldots, h_K, y_{1,1,1}, \ldots, y_{m,X,X}, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, l) =$$
$$\min_{l', v'_u, u', (a'_u, b'_u)} \begin{cases} f(h'_1, \ldots, l') + & \\ \max\left\{ w_{l,h_l} \left(\hat{r}_{a_u, b_u} + \sum_j \sum_k y_{u,j,k} S_{j,k} - d_{l,h_l} \right), 0 \right\} & \text{(a)} \\ \infty & \text{(b)} \end{cases}$$

Condition (a):

$$\hat{r}_{a_u, b_u} + \sum_{j=1}^{X} \sum_{k=1}^{X} y_{u,j,k} S_{j,k} \geq \hat{r}_{l,h_l}$$

Condition (b):

$$\hat{r}_{a_u, b_u} + \sum_{j=1}^{X} \sum_{k=1}^{X} y_{u,j,k} S_{j,k} < \hat{r}_{l,h_l}$$

where $l' = 1, \ldots, K$, $v'_u = 0, \ldots, X$, $u' = 1, \ldots, m$, $(a'_u, b'_u) = 0, \ldots, n$ and

$$\begin{aligned} h'_i &= \begin{cases} h_i - 1 & \text{if } i = l \\ h_i & \text{if } i \neq l \end{cases} \\ y'_{z,j,k} &= \begin{cases} y_{z,j,k} - 1 & \text{if } v'_u \neq 0 \text{ and } z = u \text{ and } j = v'_u \text{ and } k = v_u \\ y_{z,j,k} & \text{otherwise} \end{cases} \\ v'_z &= v_z \quad \text{if } z \neq u \\ (a'_z, b'_z) &= (a_z, b_z) \quad \text{if } z \neq u \ . \end{aligned}$$

Optimum

$$\min_{y_{z,j,k},(a_z,b_z),v_z,u,l} f(N_1, \ldots, N_K, y_{1,1,1}, \ldots, (a_1, b_1), \ldots, v_1, \ldots, u, l)$$

where for $z = 1, \ldots, m$, $j = 1, \ldots, X$, $k = 1, \ldots, X$; $y_{z,j,k} = 1, \ldots, X$, $(a_z, b_z) = 1, \ldots, n$, $v_z = 1, \ldots, X$, $u = 1, \ldots, m$ and $l = 1, \ldots, K$.

3.4.8 Dynamic program for $m|prec, \hat{r}_j, S_{i,j}|\sum g_j(C_j)$

The previous dynamic program can be generalized for any objective function $\sum g_j(C_j)$, where g_j is an arbitrary non-decreasing function of the completion time C_j of job j. The first recursion condition remains unchanged. The second recursion condition is set up:

$$f(h_1, \ldots, h_K, y_{1,1,1}, \ldots, y_{m,X,X}, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, l) =$$
$$\min_{l', v'_u, u', (a'_u, b'_u)} \begin{cases} f(h'_1, \ldots, u', l') + & \hat{r}_{a_u,b_u} + \\ g_{l,h_l}\left(\hat{r}_{a_u,b_u} + \sum_j \sum_k y_{u,j,k} S_{j,k}\right) & \sum_{j=1}^{X} \sum_{k=1}^{X} y_{u,j,k} S_{j,k} \geq \hat{r}_{l,h_l} \\ \infty & \text{otherwise} \end{cases}$$

where $l' = 1, \ldots, K$, $v'_u = 0, \ldots, X$, $u' = 1, \ldots, m$, and $(a'_u, b'_u) = 0, \ldots, n$.

3.4.9 Dynamic program for $m|prec, \overline{d}_j, \hat{r}_j, S_{i,j}|\sum g_j(C_j)$

General deadline constraints may be incorporated with the introduction of a further condition in the recursion equation. That is, if job l is the last job added to machine u, then if $\hat{r}_{a_u,b_u} + \sum_{j=1}^{X} \sum_{k=1}^{X} y_{u,j,k} S_{j,k} > \overline{d}_{l,h_l}$:

$$f(h_1, \ldots, h_K, y_{1,1,1}, \ldots, y_{m,X,X}, (a_1, b_1), \ldots, (a_m, b_m), v_1, \ldots, v_m, u, l) = \infty.$$

3.4.10 Computational complexity

Assume that the number of machines m, the number of job types X and the number of ordered sets K are fixed. The $1|prec, \hat{r}_j, S_{i,j}|C_{MAX}$ and $1|prec, \overline{d}_j, \hat{r}_j, S_{i,j}|C_{MAX}$ dynamic programs have the same form as the dynamic program in Section 3.3.1. Their complexity is thus $O(n^K)$. The dynamic programs for the $1|prec, \hat{r}_j, S_{i,j}|\sum g_j(C_j)$ and $1|prec, \overline{d}_j, \hat{r}_j, S_{i,j}|\sum g_j(C_j)$ problems have the same basic form as the dynamic program in Section 3.3.2. Their complexity is $O(n^{K+X^2+1})$. Finally the dynamic programs for the $m|prec, \overline{d}_j, \hat{r}_j, S_{i,j}|\sum g_j(C_j)$ problem (including objectives C_{MAX} and $\sum w_j T_j$) have K h_i variables, mX^2 $y_{i,j,k}$ variables and m (a_i, b_i) variables. Therefore complexity is $O(n^{K+m(X^2+1)})$.

3.5 Adding CPS constraints to dynamic programs

The CPS constraints of Psaraftis (1980) may be incorporated into any dynamic program presented in this chapter. If these constraints are used, constraint 7 in the machine job scheduling problem definition of Section 2.3.2 becomes active. Let the FCFS position of job i be I, and maximum position shift of this job be M_i. Then CPS constraints result in a set $C_i = \{I - M_i, \ldots, I, \ldots, I + M_i\}$ of feasible sequence positions for each job i. Let job i be added to the sequence at stage k. The stage number k is also the sequence position of job i. If an extra condition is added to the relevant recursion equation setting state $f()$ to infinity when the stage k is not contained in C_i, sequence positions outside the CPS range will be ruled out. That is, if $k > I + M_i$ or $k < I - M_i$ then $f() = \infty$. This condition does not affect the Big O computational complexity of the dynamic programs.

Chapter 4

The simulation model

This chapter describes the computer simulation model used to investigate a part of the Arrival Management problem. By the end of it, the reader should appreciate why the simulation has been built, what it can do and how it has been implemented. Section 4.1 specifies the simulation model in general terms. The case is made for a simulation model of airspace, the conceptual model is described and the assumptions underlying it are justified. Experimentation on a simulation model is conceived with the idea to uncover input-output relationships. Inputs and outputs that have been implemented in the simulation model are listed in Section 4.2. A significant amount of time has been spent coding up the simulation in Visual Basic. The core ideas behind this implementation are found in Section 4.3. The chapter finishes with a summary of the simulation model.

4.1 Specification of model

4.1.1 Purpose of model

The development of the simulation model has followed the pattern recommended by Law & Kelton (2000), Chapter 5. At the start of the process the goal and specific issues to be addressed by the model were discussed with Mr. Alan Drew of the Eurocontrol Experimental Centre (EEC). These are reproduced here.

Goal of this work The overall goal of the project is to develop an analysis tool to investigate scheduling and delay-sharing strategies when landing aircraft at airports.

Specific Issues to be addressed The analysis is to be undertaken on data from Stockholm airport and airspace as well as other generic airport and airspace

set-ups to be defined. Following the definition of the airspace the issues of interest will be how the system reacts to changes in the sequencing algorithms, the optimisation criteria within these algorithms and changes to the delay sharing strategies.

In order to assess the effects of scheduling and delay-sharing strategies performance indicators need to be estimable from the model. Those chosen were:

Landing rate: Number of aircraft that land in a time period.

Delay: Various delays may be of interest such as average delay to all planes, delay to all heavy or light aircraft, maximum delay of all planes, distribution of delay and delay over routes and sectors.

Air Traffic Control system risk indicators: Time in and around holding points, or time spent in approach sectors.

Air Traffic controller workload risk: The stability of advice generated.

4.1.2 Choice of model

An airspace computer simulation model was chosen to investigate the goal and specific issues to be addressed, through analysis on the performance indicators. Previous, related work has made use of analytic models and computer simulation models. Two types of simulation model have been developed. The first has modelled the dynamic updating process of aircraft sequencing to assess computational performance of different algorithms (Beasley et al. 2004), and to compare new sequencing methods with current (Venkatakrishnan et al. 1993). The second has used a model of real airspace to assess the performance of the algorithms on system behaviour (Carr et al. 2000). The second type of simulation model was chosen for this work for a variety of reasons, made with consideration to the overall modelling goal.

Effects on real airspace Interest is in what happens on a particular airspace, so a generic model of the dynamic sequencing algorithm process is insufficient.

Generic model The model needs to provide insight into effects of AMAN technology on real airspace, yet not be limited to a single set-up. This is not a problem for simulation modelling because of the flexibility that arises from the modular nature of the methodology. Different sequencer or delay-share modules may be developed and incorporated into the simulation. Once a model has been built it may be setup for different airports and airspace.

Figure 4.1: Information flows in the simulation model

Comparison of AMAN setups The model needs to be used to compare a wide variety of Arrival Manager setups, ranging from simple rules to complicated optimization routines, limiting use of analytic models.

Area of interest is pre-TMA The scope of the investigation here is wider than all published work; previously the effect of different sequences on the TMA has been examined. This work is concerned with how the system reacts to changes in the delay sharing strategies and so effects of sequence change further back from the TMA are of interest. As the scope is larger so the size of the problem for analytical methods, such as queuing theory, is greater. Simulation models of the algorithm updating process fail to model effects of delay share strategies. A simulation model of real airspace permits both the effects of landing sequences and delay-share strategies to be estimated.

Performance indicators All the performance indicators are estimable if a suitably designed and validated simulation model of airspace is used.

Experimentation Experiments on the simulation model may help to work towards the goal and specific issues to be addressed. A further benefit of simulation is the opportunity to ask "what-if?" questions that may not have been specific issues to address at the start.

4.1.3 Simulation sub-systems

The simulation of airspace used is split into 3 subsystems; the sequencer, the delay-share procedure and the airspace simulation. These interact with each other as in Figure 4.1. The arrows in the diagram indicate flows of information. The sequencer is fed using real-time information from the airspace simulation model. A landing sequence is calculated and using this, and the airspace information, the delay sharing procedure is run. Its output is fed back to the airspace simulation model and the process repeats. The following sections define in detail each subsystem.

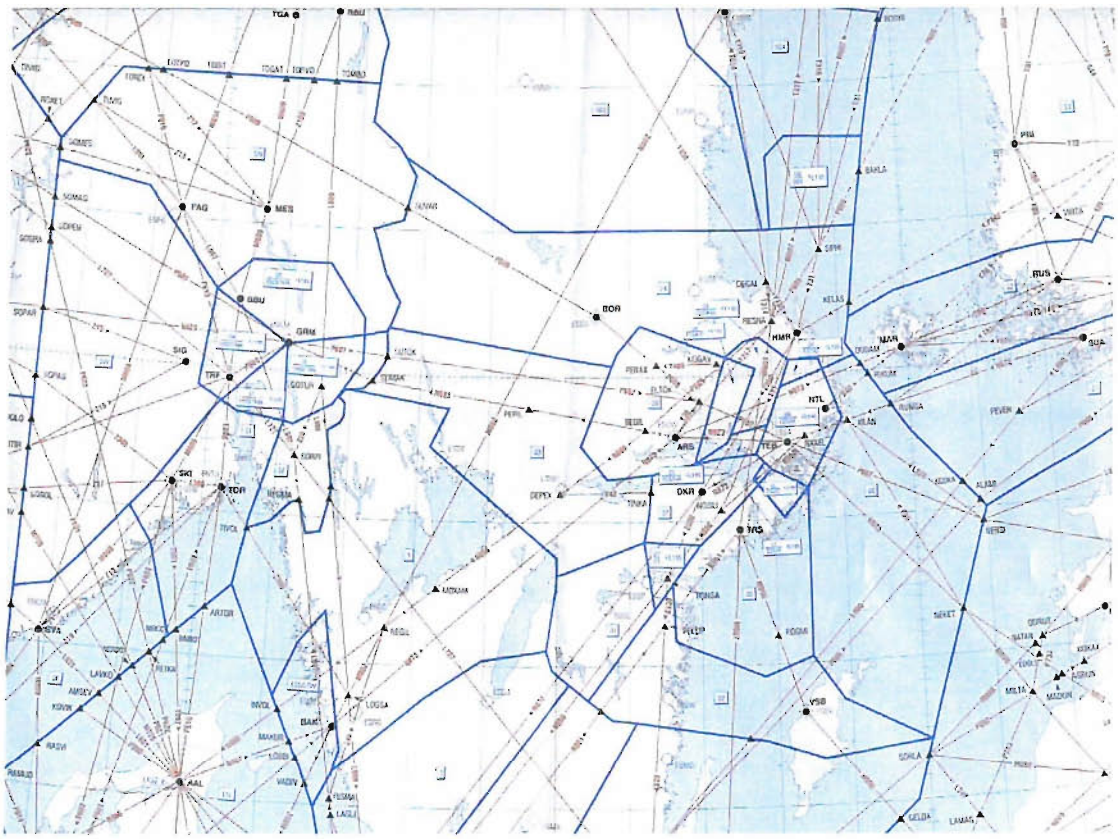


Figure 4.2: Air Traffic Control map of airspace around Stockholm

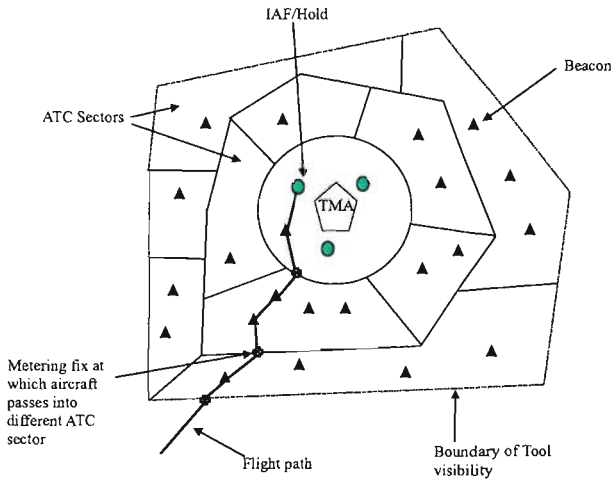


Figure 4.3: Schematic of the arrivals airspace

Model of airspace

Airspace is made up of a series of airways, navigating beacons and control sectors. Figure 4.2 shows a map of upper airspace over and around Sweden. Every aircraft flying through the sky has a flight plan that informs ATC on the expected time aircraft will fly over each navigating beacon on its route, the altitude at these points and the time the aircraft crosses a sector boundary. The simulation model of airspace is based on a 2-dimensional view of ATC flight plans. Altitude is ignored and aircraft fly across a map of longitude - latitude coordinates. The layout of this model airspace is shown in Figure 4.3.

The model does not map the entire world. Aircraft enter the model at a boundary and likewise leave at another boundary. The areas aircraft fly through in the model corresponds to the ATC sectors of interest in assessing the issues to be addressed. Aircraft leave the model the moment they are scheduled to leave the Intermediate Approach Fix (IAF) and begin approach and descent to the runway (note that the model therefore does not model the TMA). This model of pre-TMA airspace is generally quite valid. It describes the situation at Stockholm Arlanda where there are four IAF points, and London Heathrow where there are four holding points. It will also describe other airspace environments surrounding an airport where there is a concept of k points that aircraft fly through before they begin descent onto the runway.

Aircraft fly through the model based on their flight plans. The flight plans for a simulation run may either be a historical sample, or a statistical sample of historical samples. Traffic samples are generated using three pieces of information; overall arrival rates at the exit points (IAF's), probability that aircraft fly through each arrival point and the wake-vortex category probability of aircraft at different exit points. For example, a day's traffic might be generated for Stockholm Arlanda with different arrival rates for each hour of the day, probability that aircraft fly through four IAF points and the probability that the aircraft wake vortex classifications through each IAF point is heavy, medium or light. Generating a traffic sample from this information is a five-part process.

1. A non-stationary Poisson process generates an arrival sequence at the IAFs.
2. Each of these times is assigned a IAF based on their probability (a multinomial model).
3. Each of the times and their IAF is assigned a plane type based on this probability (a multinomial model).

4. The database of flight plans is randomly sampled to get the right number of planes from each category.

5. The flight plan for each of the sampled aircraft is changed so that the exit time is now the same as an exit time generated.

The way that a sample of aircraft fly through the model of airspace is described following the path of one aircraft through the system. The same events occur for each aircraft.

1. Aircraft arrives in the tool visible area at a time given by either a prepared traffic sample, or according to a statistical distribution. At any moment from this point on the flight may receive advice from the delay-share strategy on new target times it ought to pass over navigating beacons. If it receives no advice from the delay-sharing procedure the target times are as the flight plan data.

2. Aircraft arrives at each navigating beacon and sector boundary at the target time.

3. Plane arrives at the planned IAF at the planned time.

4. Aircraft exits IAF at a time based on aircraft type and IAF position to land in its sequencing position with the required separation to follow the previous landing.

Sequencer

Any sequencing algorithm may be used to sequence landings from those surveyed in Section 2.6 to the novel dynamic programs in Chapter 3. The sequence algorithm takes any information it may need such as earliest land time, preferred land time or latest land time from the airspace simulation. Using this information it forms a landing sequence. The sequencer may be run at a number of points in the model. For instance landing sequences may be (re)calculated each time a new aircraft enters an IAF.

Delay-share strategy

The delay-share strategy turns the landing sequence into a schedule. That is, it assigns landing times to the aircraft based on their sequence. Further, it assigns times over navigating beacons, IAF's and sector boundaries based on its strategy.

4.1.4 Model assumptions

The simulation model makes a number of assumptions. Descriptions of these with justifications follow.

1. Airport and landing procedure not modelled.

   Once the plane has left the hold it leaves the model. The model does not consider the system beyond the hold or indeed after landing. This would involve modeling the TMA of each airport the model is set-up for. The TMA is a very airport-specific and complicated area of airspace. To go to this level of detail would require significant effort every time a new airspace were set-up. For example, the simulation model in Andreussi et al. (1981) may be validated for Rome TMA, but the landing procedures at another airport may be so different that a completely new model is needed. This is undesirable. The main justification for this assumption comes from the specific issues of the problem to be addressed. These are focused on the ATC system, not the TMA surrounding an airport. The performance indicators from airspace pre-TMA should be sufficient for this purpose.

2. Planned landing rate can and will be achieved by approach controllers.

   Validation of the simulation model should test whether confidence in the landing rate planned by the model is comparable to that of Air Traffic controllers. This assumption is also justifiable if the model landing rate does not validate against real life, provided no inference is made on differences between current data operations and new algorithms. That is, only differences between new algorithms or delay-share strategies are compared. In this case (assuming that there would be no difference in controller separation behaviour across the combinations) all results are based on the same assumption of landing rate, so the comparison is fair. If interest is in assessing real landing rate with model landing rates and the validation process results in rejection of this assumption, then a statistical model of landing times, similar to that used by Venkatakrishnan et al. (1993), could be used in the simulation model.

3. Interactions between aircraft ignored.

   Some delay-sharing strategies may cause interactions between aircraft that generate extra delay to the system. These interactions will not be modelled. For instance, there is nothing in the model to stop two aircraft flying over the same point in airspace at the same time. No conflict detection or resolution measures are taken. This assumption is made for three reasons. First, if

interactions are an issue then indicators such as maximum number of aircraft in a sector per unit time should highlight that there could be a potential interaction problem that may need further investigation. Second, validation of the model ought also to help justify this. If there is confidence in the model behaving like reality then there may not be the need to go to such the level of detail of picking up interactions and generating appropriate control actions. Third, the focus of this work is not on avoiding conflicts and so another sub-system within the model that works to do this may be a waste of resources. Despite these reasons this assumption is a significant simplification of the ATC system. Safety is paramount in ATC and so indicators for interaction ought to be examined carefully to double check that unreasonable control actions are not commonplace.

4. Other flights in airspace (planes flying to alternative destinations) are not considered.

   Since interactions between aircraft are ignored for landing aircraft, there is no need to include other flights in airspace. The main effect they may have on flights directly affected by sequencing would be through interactions. Removing these aircraft will reduce the complexity of the model.

5. Aircraft taking off from the airport are not included.

   Sequencing may have a direct effect on these aircraft. However, the focus of this work is on the effects of AMAN systems on arrival aircraft. If aircraft take-off and land on separate runways, then the main effect that may be noticeable for the arrival aircraft is through interactions. For reasons previously discussed these are not considered. Since arrivals are of primary interest and aircraft departures are not within the scope of this work, they are removed. However, the flexibility of simulation means that future work could be done to incorporate them into the model without great expense.

6. All aircraft have the same priority.

   It is assumed that no bias exists in the current ATC system and so this is repeated in the model.

7. Sequencer and delay-sharing strategy are automatically followed by controllers.

   This is unlikely to be the case with an operational AMAN where the controller will issue all the instructions to aircraft. It is assumed to be true in the model because interest is in determining how the different systems perform relative to one another, if the system were actually put in place and

used. The interest is in efficiency indicators, not questions about operational viability.

8. Instructions to aircraft are not subject to error.

   Controllers are assumed to automatically follow the algorithm instructions, and so are aircraft. This may be justified for similar reasons to the controller issue. In ATC aircraft generally follow the advice given them, and interest in the model is not in the times when they do not.

9. Human factors, in particular communication issues are not modelled.

   Time for controllers to communicate changes to the plans, or guide aircraft is not considered. It is assumed that once a plan has been made the information is available and comprehended by aircraft and controllers. It is not considered necessary to model the delay in information processing in order to assess sequencing strategies because the interest in this model is efficiency of sequencing, not questions about operational viability and human factors. The design of the computer model here is inappropriate for such concerns.

10. Only landing airport modelled.

    Although an AMAN may have some influence on aircraft pre-AMAN boundary this is likely through aircraft interactions. Since these are ignored and the focus of the model is on the direct effect that an AMAN may have on the ATC system, aircraft are not modelled outside the area in which an AMAN may function. Most departure airports will be outside this range. Some will be included inside the range (i.e. short-range flights). However, the takeoff procedure will follow no special routine. Aircraft will appear at a certain point in the AMAN and pass over other beacons towards landing, just as other aircraft who did not takeoff in the AMAN area.

11. No explicit calculation of environmental constraints.

    Noise levels and emissions are not actively included in the model. It should be possible to estimate these quantities using the time spent in the air by each aircraft and the number of aircraft in the air at certain positions over a period of time. Environmental constraints are not a specific issue to be addressed by the model so more precise indicators are not included.

12. Aircraft cannot speed up.

    Some algorithms developed by authors such as Beasley et al. (2001) include the concept of aircraft speeding up over their planned flight plan. Although there is scope to increase the speed of aircraft during their flight this should

be a very unusual case. Indeed, a technical group advising at Eurocontrol has felt that an operational AMAN should not issue speed up advisories. The simulation model takes this viewpoint and the information on earliest land time it sends to its sequencer is the time that the aircraft would land if it were to fly its flight plan.

13. Wake vortex of an aircraft landing will only affect the aircraft landing after it.

    This assumption is abandoned by some authors such as Fahle & Wong (2003). However, for arrival traffic, consultation with Air Traffic controllers at Eurocontrol has assured the author that this assumption is valid.

4.2 Input and output

4.2.1 Input

Several types of input may be given to the model, ranging from simple choice of algorithm to providing a database of flight plan data. Descriptions follow.

Setting up model

To set up the model, information is needed on aircraft's time over navigating beacons and sector boundaries. All this information is available in Flight Plan data. Eurocontrol have a source of this data for flights in Europe stored in the Eurocontol Central Flow Management Unit (CFMU) database. Flight plans are calculated using 4D-trajectory algorithms, so the expected speed over points, altitude levels and other important variables are used to calculate the times over points. Consequently, no knowledge of where the beacons are or the speed of aircraft is needed for the model to calculate positions of aircraft since this information is implicitly contained within the CFMU data. To set up a model of new airspace it is necessary to use CFMU data for that airspace. The CFMU data is modified so that aircraft do not arrive before the model boundary, and leave after the IAF exit points. The model can then be run using either statistical samples of the data or an actual day's CFMU data.

Algorithms

Any sequencing algorithm could be incorporated into the simulation model. Six algorithms have been implemented. These are:

1. First-Come First-Serve at IAF points.
2. The $1|\overline{d}_j, \hat{r}_j, S_{i,j}|C_{MAX}$ dynamic program described in Section 3.3.2.

3. First-Come First-Serve at runway.
4. A heuristic algorithm minimizing total delay described below.
5. The $1|prec, \overline{d}_j, \hat{r}_i, S_{i,j}|C_{MAX}$ dynamic program of Section 3.4.2.
6. The $1|prec, S_{i,j}|\sum w_j T_j$ dynamic program from Section 3.2.3.

It is not possible to compare all possible sequencing algorithms, so the algorithms implemented have been carefully chosen to permit investigation of a range of techniques, constraints and optimization criteria. The broad range of algorithms increases the probability that important experimental effects are identified. The only algorithm not previously defined is algorithm 4. This is a simple heuristic chosen as a result of discussions at the Eurocontrol Experimental Centre. It represents a potential algorithm for use in an operational AMAN system.

Heuristic algorithm 4 This algorithm runs every time a new aircraft i arrives. If the FCFS position of i in the current recommend sequence is F_i, and i has a Maximum Position Shift (MPS) M_i it may make from its FCFS position, then it may be placed in any sequence position from $F_i - M_i$ to $F_i + M_i$. The algorithm chooses the position with the minimum total sequence delay that ensure i is not sequenced before its the earliest possible time. In the event of a tie the order of preference is FCFS position, increasing shift backwards and increasing shift forwards.

Delay share strategies

Four delay-sharing strategies are incorporated in the model. These are:

1. All delay at IAF points in holding patterns.
2. Delay apportioned evenly through route.
3. Delay in flight path segments as late as possible.
4. Delay in flight path segments as early as possible.

Strategies 2 - 4 delay aircraft before the IAF point if possible. The maximum amount of delay an aircraft may lose before IAF holding points is determined by a parameter of maximum proportion of delay they may lose along each flight path segment. If the subsequent maximum pre-IAF amount of delay is exceeded then the remainder goes into holding patterns at IAF points.

Re-sequence strategies

Re-sequencing strategy may have an effect on the results from an experiment, and the computational performance of the simulation. The events that may be used to trigger the sequencing algorithms are:

1. A new aircraft enters an IAF.
2. A new aircraft enters the system.
3. A batch of a aircraft has entered the system since the last batch of a aircraft.
4. A batch of a_t aircraft enters the system since the last batch of a_t aircraft, where a_t depends on the current time t.
5. New aircraft enters the system and time since the last re-sequence $\geq t$.

The results of re-sequencing aircraft do not affect the current aircraft flight segment, unless the aircraft is in a hold. For example, if the delay-share strategy is as late as possible, and an aircraft has been given a delay as a result of another aircraft triggering the re-sequence, then the delay advised only begins from the next navigating beacon in its route, if the aircraft is not in a hold.

When running model

Experiments on simulation models are set-up to examine input-output relationships. The sequencing algorithms have been coded for an airport with 4 IAF points, 3 different wake vortex categories and 1 runway. Inputs to the simulation model are listed in Table 4.1 and the number of parameters listed. The inputs listed can be split into three categories based on how they could be used in experimentation.

Variables Traffic levels and control options are variables. That is, they may be altered to different values or settings when experimenting with the model. In real life ATC would have complete control over the control option inputs of sequencing algorithm and delay-share strategy and so these are real-life variables as well as simulation variables. At a tactical level ATC does not have control over traffic levels. However, traffic level is a variable in the simulation because it may take different values over time. Traffic conditions on a Monday in January will be quite different from a Sunday in July.

Assumptions that make model close to reality The time it takes an aircraft to move from the IAF to the runway, landing separations and landing speed are

Table 4.1: Summary of simulation inputs

Input	Description	Number
Traffic Description	Arrival Rate by hour	24
	WV Category at IAF probability	12
	IAF probability	4
	Random seed	1
Algorithms	Implemented	6
Delay-Share Strategy	Implemented	4
Re-sequence Strategy	Implemented	5
Landing speed	Parameter converts distance to time	1
Time from IAF to runway	Dependent on aircraft and IAF	12
Separation matrix	Deterministic	9
TOTAL		78

assumptions that need to take a certain form in order to make the model valid. These inputs may be altered in the model, but it is likely that their validated range is small due to their nature.

Parameters These are inputs that the real system has no control over. However, the simulation model has control over them and the model may remain valid if they are altered. These parameters in the model include the separation matrix used in the sequencing algorithms and random number seeds.

4.2.2 Output

There are several types of output generated by the simulation:

Point to point segments and ATC sectors: Total, maximum, minimum, mean, standard deviation for number of aircraft, and aircraft flight time.

Sector delays: The aircraft ID and delay it had to lose in a sector at time t.

Aircraft delay: The delay each aircraft had to lose over each point in its flight plan.

Land times: Land time and aircraft ID.

Advised delays: The maximum, minimum, mean and standard deviation of delay advice given to each aircraft ID.

The level of detail in output is high, and so output can be manipulated to produce all the performance indicators in Section 4.1.1 and many others.

4.3 Implementation

4.3.1 Airspace simulation

The computer simulation model was built in Microsoft Visual Basic 6.0. It has a form-based interface designed to be used at Eurocontrol. The user may define new airspace by importing traffic samples from CFMU data, generate new traffic samples, set up experimental AMAN systems and run experiments. The simulation code is defined by three features; it is a discrete-event simulation, programmed in an objected-oriented fashion and input and output is controlled through databases. Descriptions of why these approaches were taken follow.

Discrete event simulation

Discrete Event Simulation (DES) is a modelling approach where a system is modelled over a countable number of points in time. At each point an event occurs that may change the state of the system (Banks 1998, Law & Kelton 2000). Discrete Event Simulation is particularly appropriate to the airspace simulation model because times aircraft fly over points naturally form the basis of events. The main property of a DES simulation is its ordered event list. In the airspace simulation this stores the times each aircraft in the traffic sample will next fly over a point in airspace. The simulation performs the actions that are associated with the first event in this list and then updates, first removing the current event. If the aircraft has left the system nothing further is done to the event list. If the aircraft will pass over another point in the future the next point in the flight path event is added to the list. This process continues until all aircraft have left the system.

Object-Oriented approach

The model has been programmed in Visual Basic using an Object-Oriented approach. This is a convenient approach to the airspace simulation because information and methods of accessing or manipulating the information are grouped together. This style of programming helped when debugging the model, and means that additional features may be more easily added to the simulation. Classes are used to represent aircraft, flight segments and output statistics. The aircraft class stores all information about the aircraft needed in the simulation - from inputs such as its planned route to outputs such as the delay over points in the route, in addition to methods that, for example, output the information. Similarly the route class has various methods and properties relating to the time aircraft pass over points. Objects in the output class for point to point segments and ATC sectors are updated as the simulation progresses, until the simulation

end when all output is saved in a database before being destroyed from memory. A special object in the code is the doubly-linked Event List class. A binary search method is used to add items to this.

DAO link to Access database

A quick and easy way to deal with large sets of input and output data is to use a database. A major benefit of such an approach is that efficient algorithms for searching and manipulating the data are part of the database. The DAO protocol was used in the simulation model to input and output data from two Microsoft Access databases:

AMAN Input This is used when producing new traffic samples, and running the simulation. It contains a table with the flight plan data, populated with CFMU data from the airspace under consideration. A number of queries are run on the table to sample the aircraft for the simulation. The database stores the traffic samples generated.

AMAN Output Output from each run is stored in this database. Each simulation run outputs a large amount of data into this database. Queries may then be made to obtain required performance indicators. A schematic of the relationships in this database is found in the Appendix.

4.3.2 Algorithms

The algorithms were also coded in Visual Basic. The dynamic programs chosen are not too computationally expensive, so the speed of Visual Basic in relation to other languages is not as important as it might have been.

4.4 Summary

A discrete-event simulation model of airspace surrounding Stockholm Arlanda airport has been built in Visual Basic. The design of the simulation model involves sequencing, delay-sharing and airspace components. The model is generic enough to permit further work to be undertaken on new airspace environments. The simulation model has been built as an analysis tool to investigate scheduling and delay-sharing strategies when landing aircraft at airports. There are differences between this model and previous work. First, the area of airspace considered is two sectors back from the TMA, whereas previous work has focused on the effects in the TMA area. Second, the effect of sequencing algorithm and delay-sharing strategies on a real airspace may be extracted. Previous work has not considered

the delay-sharing problem. Lastly, credibility of the model is greater than some previous work on sequencing algorithms because a real system is simulated. The simulation is built to enough level of detail to enable some performance indicators on delay, landing rate, efficiency and controller workload to be extracted for investigation through experimentation.

Chapter 5

Validation of simulation model

"Two things are identical if one can be substituted for the other without affecting the truth"

Wilhelm Gottfried Leibniz
(Loemker 1969)

This chapter reports some quantitative and qualitative methods used to validate the computer simulation model. The methods are not used to test that the model and reality are identical - they are not, rather they are used to gain confidence that similar conclusions are drawn from the model as would be found in the real world. A validated AMAN simulation model may then be used for insight into the effects on the real ATC system of different sequencing techniques and delay-share strategies.

The comparison between model and real world is based on data recorded by Eurocontrol on aircraft flying into Stockholm Arlanda in Autumn 2003. Specific reference to the traffic samples used, dates of traffic samples or IAF point names are not made for reasons of confidentiality. Section 5.1 describes how this data has been extracted to be of use in validation. Section 5.2 reports validation of the sampling procedure used to generate traffic. Sargent (2001) describes a variety of methods that may be used to validate the model output. Three are used in Section 5.3: confidence ranges for difference in output, sensitivity analysis and investigation of dynamic behaviour. Section 5.4 reports the significance of the tests for overall model validation, the strengths and weaknesses of the techniques used and summarizes the findings.

5.1 Validation data

The data used in the validation of the model was recorded on the Eurocontrol PROVE platform over a number of days. The times of the traffic samples are shown in Table 5.1. Three data sources were recorded: Radar Track (Track), Estimate (EST) and Flight Plan (FPL) data. This section describes how these sources were linked for use in validation.

Linking data sources Information from International Civil Aviation Organization (ICAO) that describes EST and FPL messages is found in the Appendix. FPL data provides information known prior to take-off about a flight, such as the aircraft wake-vortex category. The EST messages estimate the time flights will pass over navigating beacons. They are sent automatically on the AFTN network and are calculated by the ATC system that covers the corresponding area. The Track data file contains a large number of data fields (listed in the Appendix) such as the 4-D position of an aircraft. However, not all the data fields were present in the Track data recorded. In particular the operational flight plan ID and callsign were missing. These would be useful fields to identify aircraft and link them to FPL and EST data. The way to distinguish aircraft was based on the SSR code field. This 4 octal digits code is sent from the transponder of every aircraft and detected by radar. The SSR code is present in Track messages and in EST messages. Callsign is present in FPL and EST messages. The FPL and Track data can therefore be correlated via the EST message. However, linking on SSR code is not foolproof and a number of issues were dealt with to produce sensible validation data.

SSR code is not unique It was possible to find, for example, an instance when 4 aircraft with different coordinate positions had the same SSR code at the same moment in time. This was not necessarily a database error because it is common to reuse the SSR codes during the day. If linking SSR code to FPL data produced a number of flight plans for a particular Track data flight, the choice of link was made by choosing the FPL with EST land time closest to estimated Track land time.

Aircraft change SSR code mid-flight In this case there is no way of linking the Track data aircraft with its old SSR code in FPL and EST data. As a result an incorrect FPL may be linked to a flight in the Track data. One sure symptom of this is an aircraft with an enormous delay. Therefore, these errors were guarded against by removing links where aircraft have delay in excess of 30 minutes.

Time over beacons not explicit The actual time an aircraft passes over a beacon (such as an IAF) was chosen as the point with closest Euclidean distance to the beacons longitude-latitude position.

When has an aircraft landed? There is an altitude field in the Track data. If land times were based on when this is zero problems would occur because there are some flights where this field is zero throughout. Land time may be estimated by looking at the last time the aircraft SSR code is found in the Track data. The SSR code will stop being received by radar once the pilot has turned off the transponder. This approach may not be accurate for the last minutes in the Track data sample so a combination of the altitude level and last received track point was used: if altitude was less that 50 feet and the last radar track time recorded was not equal to the last time in the sample, then the aircraft was deemed to have landed.

Aircraft delay at a point in its flight plan was based on the difference between estimated radar track time at the point, and the first recorded estimate time of arrival in the EST data. In particular, the time an aircraft passed over its IAF point was calculated as the time the aircraft was closest to the IAF point it was due to pass in its EST message. Landing rate performance indicators were based on the landing time, as calculated above.

FPL Data Summary The FPL data describes the underlying expected situation, and it is available for the entire day radar track data was recorded. Analysis of the FPL data is presented graphically in Figures 5.1 to 5.3. Figure 5.1 shows how the number of aircraft landing in the FPL data at Arlanda varies with each sample. Each line corresponds to a different day in which a sample of Track data exists. Some radar track samples were taken on the same day, so there are fewer days than radar track samples. Each line in Figures 5.2 and 5.3 similarly corresponds to a day with a Track data sample. These graphs show the proportion of aircraft flying to different IAF points (or holds). Each sample is from Monday to Friday, within a month apart. The charts suggest that the different traffic samples belong to a homogeneous set.

Table 5.1: Recorded Track data time periods

Sample	Time Start	Time End
1	16:10:33	18:09:10
2	09:08:35	10:19:48
3	13:49:15	15:12:08
4	09:37:29	11:00:23
5	09:19:22	12:57:52
6	13:20:51	16:17:12
7	08:25:36	11:12:46
8	06:10:22	08:42:21
9	08:06:24	09:45:15
10	13:31:03	15:51:33
11	08:59:26	12:11:10
12	14:07:08	15:29:48
13	06:54:34	08:09:02
14	09:53:52	11:28:27
15	08:13:34	09:16:53
16	09:45:45	10:49:31

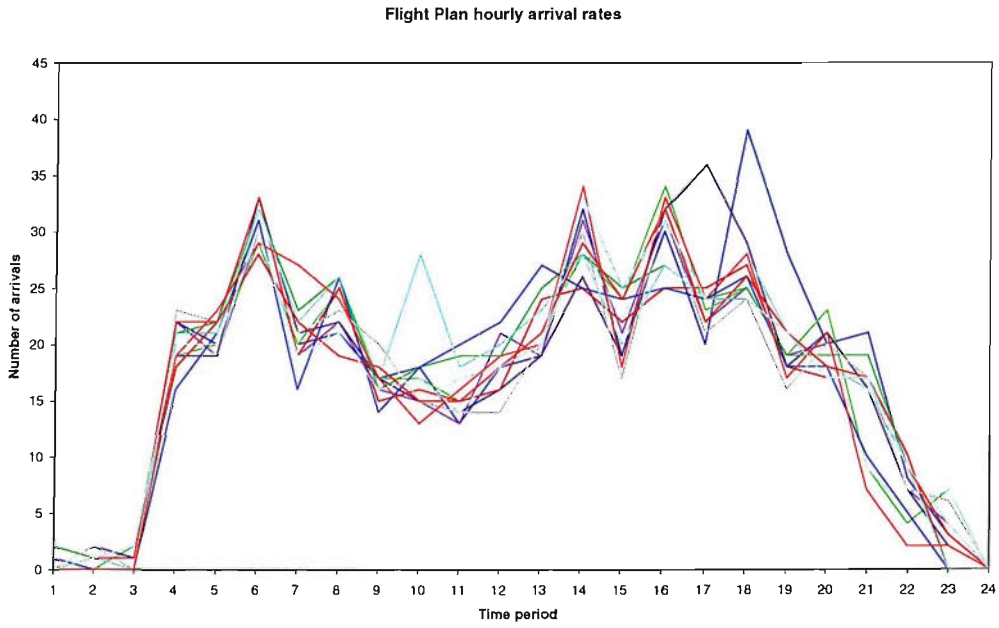


Figure 5.1: FPL Landing rates for each day Track data recorded

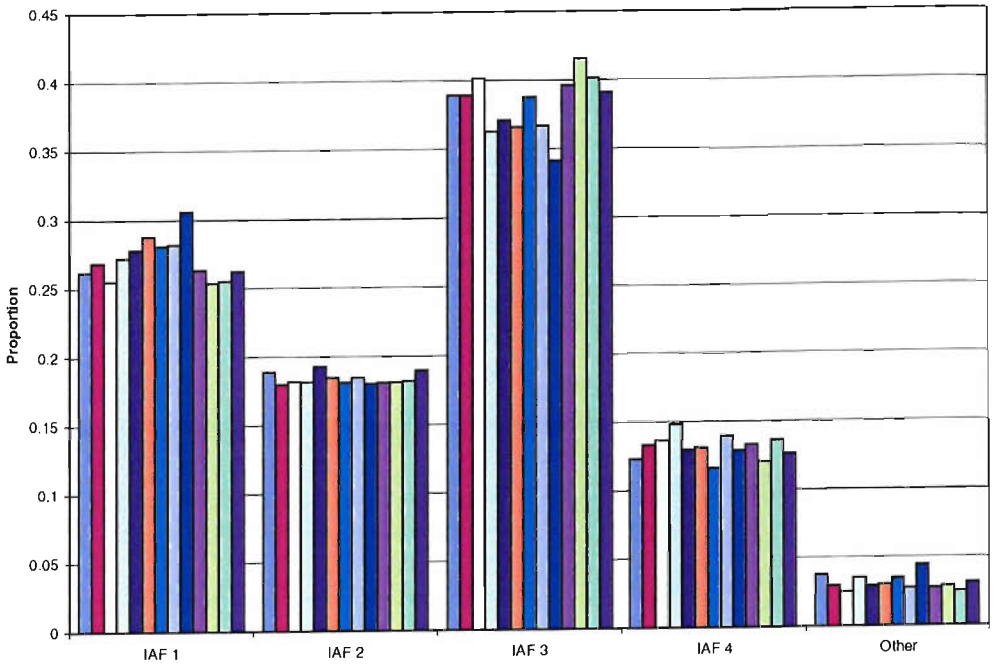


Figure 5.2: Daily proportion of aircraft flying to each IAF for Track data recorded

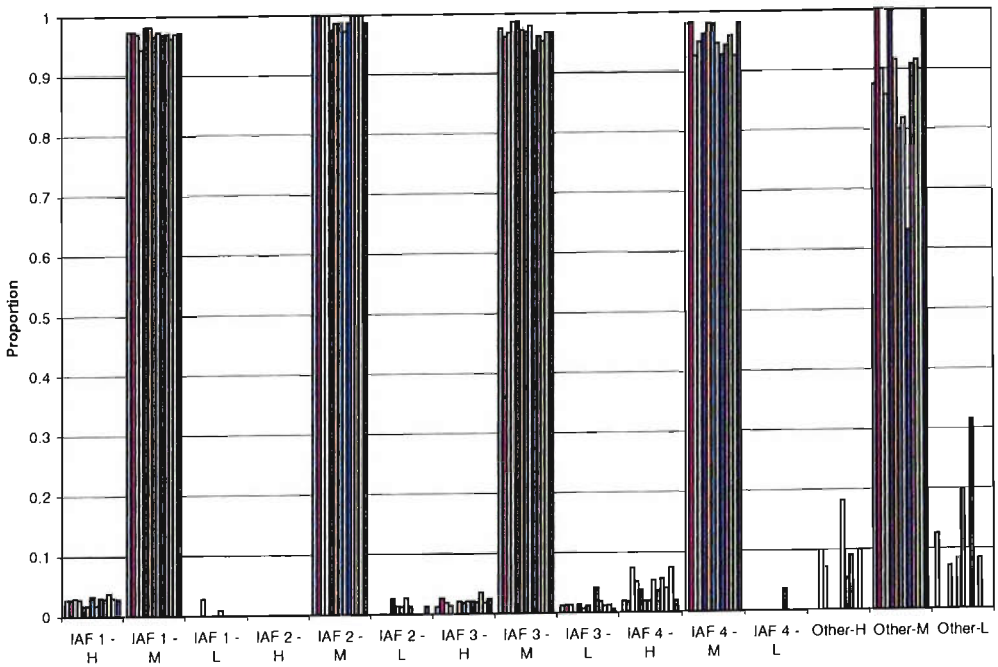


Figure 5.3: Daily aircraft type proportions by IAF for Track data recorded

5.2 Model inputs

Question Given the data sources described in Section 5.1, is there enough evidence to reject the sampling procedure as a sufficiently accurate model of reality?

The process to generate traffic sample described in Section 4.1.3 follows five steps. This section will focus on validating the first three components of the process.

5.2.1 Arrival rate model

Question Does a non-stationary Poisson process with rates that may change hourly accurately represent the arrival process of aircraft to the four IAF points (combined) at Stockholm Arlanda?

Hypothesis I In any time period of 1 hour during any day at Stockholm Arlanda arrivals to IAF points follow a Poisson process.

Law & Kelton (2000) point out that if a process is Poisson between times [0,T] then the arrival times are distributed uniformly between [0,T]. So Hypothesis I reduces to:

Hypothesis II In any time period of 1 hour during any day at Stockholm Arlanda the arrival times X are distributed Uniformly [0, 1 hour], i.e. $X \sim U[0, 1]$.

Test 1 If there are K samples from the Track data of arrivals at the four IAF points, then testing simultaneously that all K samples are $U[0, 1]$ may be done using K-Sample Empirical Distribution Function (EDF) Goodness-of-fit (GOF) statistics. Alternatively, since observations in the samples are independent we could pool them together into a single sample, and test if this sample is distributed $U[0, 1]$. Again, GOF statistics may be used.

Result Table 5.2 shows the test scores for the K-Sample Cramer Von Mises W_k^2 (Kiefer 1959), K-Sample Anderson Darling A_k^2 (Scholz & Stephens 1987) test statistics, and for the pooled data the Anderson Darling (A^2), Kolmogorov-Smirnov (K) and Chi-Square (χ^2) tests (see D'Agostino & Stephens (1986)). They show that there is not enough evidence to reject the null hypothesis.

Table 5.2: Results for Hypothesis III, Test 1

Statistic	Value	p-value
W_k^2	2.461	0.716*
A_k^2	15.099	0.723*
A^2 (Pooled)	2.196	0.076*
χ^2 (Pooled)	17.757	(df=22) 0.603
K (Pooled)	0.057	0.111

*2000 Bootstraps

5.2.2 Arrival route model

Question Does the probability that aircraft fly to different IAFs during a day follow a multinomial distribution?

Hypothesis III The number of aircraft (X_1, X_2, X_3, X_4) arriving at each IAF is distributed with a multinomial distribution with parameters n, p_1, p_2, p_3, p_4.

Test 1 Estimate p_i from FPL data from half of the data set using the maximum likelihood estimates. Test with the remaining half to see if it fits the estimated multinomial model using χ^2 test statistic.

Result Using the Chi-square goodness-of-fit finds $\chi^2 = 21.854$ on 30 degrees of freedom. A reference distribution formed from 2000 bootstraps results in a p-value of 0.901. Thus, there is not enough evidence to reject the null hypothesis.

5.2.3 Wake-vortex category model

Question Does aircraft wake-vortex category probability at each IAF follow a multinomial distribution throughout a day?

Hypothesis IV The number of aircraft (X_1, X_2, X_3) split by wake-vortex category to a particular IAF is distributed with a multinomial distribution with parameters n, p_1, p_2, p_3.

Test 1 Estimate p_i from FPL data from half the data set provided by Eurocontrol using the maximum likelihood estimates. Test the remaining data set to see if it fits the estimated multinomial model using χ^2 test statistic.

Result Table 5.3 shows the test scores. There is not enough evidence to reject any of the null hypotheses individually.

Table 5.3: Results for Hypothesis IV, Test 1

IAF	Statistic	Value	df	p-value
IAF 1	χ^2	2.808	10	0.985*
IAF 2	χ^2	13.056	10	0.169*
IAF 3	χ^2	22.101	20	0.325*
IAF 4	χ^2	7.728	10	0.654*

*2000 Bootstraps

5.3 Model outputs

The main outputs from the model are landing time and delay of aircraft. This section focuses on validating that the delay and landing rate outputs from the model are an accurate representation of reality. Three types of analysis are used; confidence, sensitivity analysis and dynamic behaviour. Since the traffic samples from the Track data may be seen as all having the same underlying type of traffic sample, the scope for sensitivity analysis is reduced. The range of confidence in the model is also limited to the single set of parameters used in the validation. The validation methods that follow are based on comparisons with output from 500 runs of the simulation, with the model inputs setup as Table 5.4. The parameters for generating traffic samples were set based on maximum likelihood estimates using half of the data set provided by Eurocontrol.

5.3.1 Delay

Question Is there a difference between delay output from the model and delay output in real life?

Confidence

Hypothesis V There is no difference between delay in the model set with maximum likelihood estimates of inputs, and landing delay from Track data (landing delay = difference between first EST land time and actual Track data land time).

Test 1 In order to get a reference distribution function of delay from the model, the delay from a single simulation run must be aggregated, and then all the aggregates pooled together. Mean delay is used as the aggregate function. It is a useful measure since the real-life sample lengths are variable (see Table 5.1), and it scales for length. However, since the Track data is taken from different times of day, comparisons may not be fair unless it is possible to reject the hypothesis that

Table 5.4: Summary of inputs to simulation model validation setup

Parameter	Value(s)
Arrival Rate by hour	0.5, 1, 0.75, 21, 21.75, 28.75, 22.25, 21.75, 17.75, 15, 14.75, 17.25, 20.25, 31.5, 21, 31.75, 22.75, 25.5, 18.75, 19, 14.5, 6.75, 3.75, 0
Heavy at IAFs probability	0.0310, 0, 0.0244, 0.0475
Medium at IAFs probability	0.969, 0.997, 0.963, 0.956
Light at IAFs probability	0, 0.003, 0.012, 0
IAF probability	0.266, 0.190, 0.408, 0.136
Random seed	1-500
Sequencing Algorithm	FCFS at Hold
Delay-Share Strategy	All delay at IAF in Hold
Runway	1
Separation matrix	Eurocontrol data
(Distance in Nautical miles)	H: 4, 5, 6 / M: 3, 3, 5 / L: 3, 3, 3 (H, M, L)
Landing speed	296.32 km/h
Minutes from IAF to runway	Eurocontrol data

	H	M	L
IAF 1	13.82	13.82	15.05
IAF 2	18.42	18.42	22.07
IAF 3	18.16	18.16	22.69
IAF 4	16.50	16.50	17.19

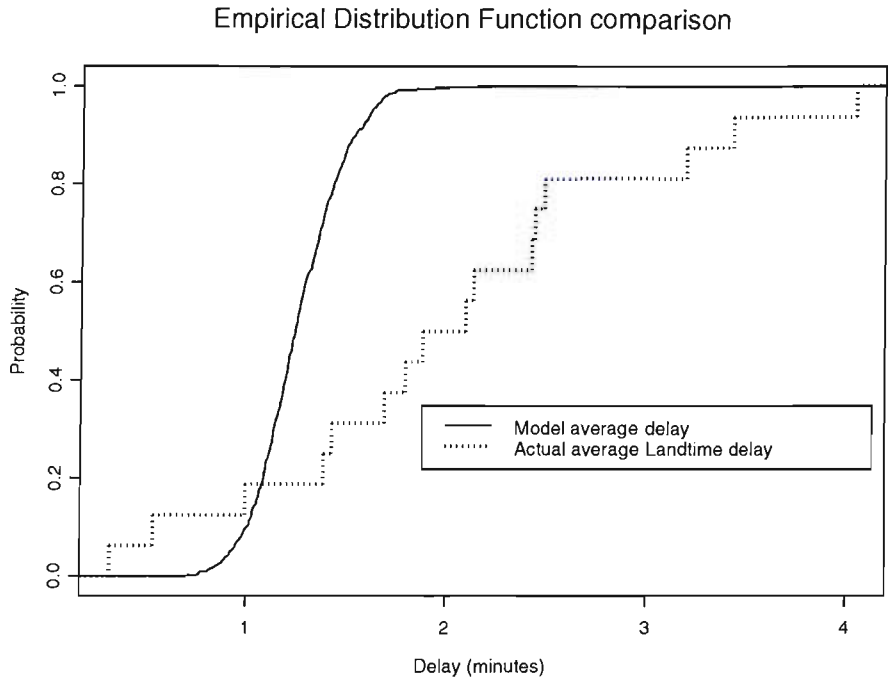


Figure 5.4: Empirical Distribution Function plots of model mean delay and actual mean landing delay

delay is dependent on time of day. The hypothesis is rejected for the data set in a dynamic behaviour test later in this section. The first test for a difference in delay is a graphical comparison of the Empirical Distribution Functions (EDFs) of mean delay distribution from the model, with the distribution of mean delay from the traffic samples.

Result Figure 5.4 shows the comparison. Visual inspection rejects the hypothesis that the two distributions are the same, so there is no need for more formal statistical tests.

Delay at IAF is different to delay on runway. Running a paired-t test (Law & Kelton 2000) on the mean IAF delay and runway delay on all traffic samples (Shapiro-Wilks test for normality (D'Agostino & Stephens 1986) of both data series non-significant at 10% level) produces a 95% confidence interval on the difference of [-2.611,-0.753], i.e. mean IAF delay is greater than mean runway delay. Perhaps the pilots do not all turn the transponder off at the same moment they land, affecting the delay indicator. Or maybe the control actions that take place in reality are not as deterministic as those assumed in the simulation. Whatever the reason, the simulation effectively ends at the IAF point and no modelling is done on the

airspace between the IAF and runway. The model assumes that aircraft take a fixed time (dependent on their type and IAF) to fly from IAF to touchdown. Some analysis showed quite a variation in the time between IAF and runway, as might be expected given the difference in IAF and runway delay values. If land time delay were the purpose of the simulation this result might suggest further work in the model on the time between IAF and runway. However, it is not necessarily appropriate to compare the delay outputs from the model to the land time delay. The simulation model has been created to investigate scheduling and delay-sharing strategies when landing aircraft at airports. Interest is in how the system reacts to change in sequencing algorithms, the optimization criteria within the algorithms and change to delay sharing strategy. By system it is meant the area contained inside the model boundaries, i.e. up to and including the IAF points. A more appropriate comparison between model and reality would be to compare delays at IAFs from the Track data, with delay from the model. This leads to the next hypothesis.

Hypothesis VI There is no difference between delay in the model set with maximum likelihood estimates of inputs, and IAF delay from Track data.

Test 1 Compare EDFs.

Result An EDF showing this comparison is presented in Figure 5.5. Hypothesis VI is rejected based on visual inspection.

Figure 5.5 suggests why there might be a difference. The shapes of EDFs are similar except the model delay seems shifted to the right: the simulation model is overestimating delay. Difference in EDFs seems arise because Track IAF delay may be negative, but the delay given to aircraft in the model is by definition always positive. When the model is experimented on the interest is in how the system reacts to change and what effect the change has on delay. Since delay in the model is always positive, perhaps the wrong question was originally asked.

Question revisited Is there a difference between positive delay output from the model, and positive delay output in real life?

Hypothesis VII There is no difference between mean positive delay in the model set with maximum likelihood estimates of inputs, and mean positive IAF delay from Track data.

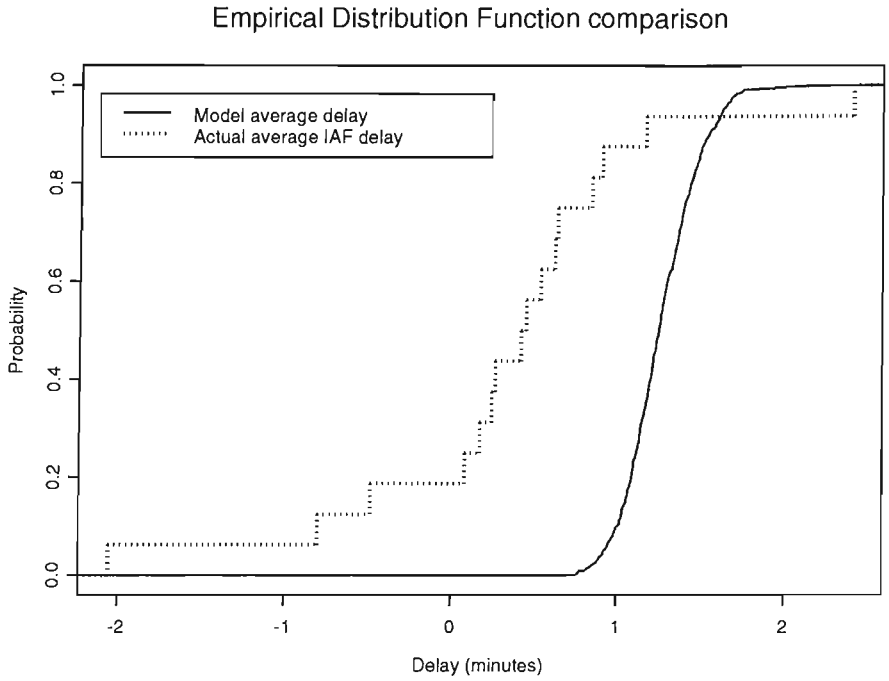


Figure 5.5: Empirical Distribution Function plots of model mean delay and actual mean IAF delay

Test 1 Compare the EDF plots.

Results Comparison of EDF functions for mean IAF delay greater than zero from the model, and Track data is shown in Figure 5.6. It is difficult to know whether to accept or reject the hypothesis that they are the same based on visual inspection.

Test 2 Test for differences between the distributions with 2-sample EDF test statistics Cramer Von Mises W_2^2 (Kiefer 1959) and Anderson Darling A_2^2 (Scholz & Stephens 1987). The W_2^2 statistic may be further broken down into components testing for difference in mean C_1 , variance C_2, skewness C_3 and a remainder C_R (Cheng & Jones 2004).

Results Table 5.5 shows the scores and bootstrapped p-values for these statistics. None may be rejected at the 95% level. The decomposition of W_2^2 into tests for mean C_1, variance C_2 and skewness C_3 suggests that there is no significant difference in these distribution summaries. Difference between means is further examined in the next hypothesis.

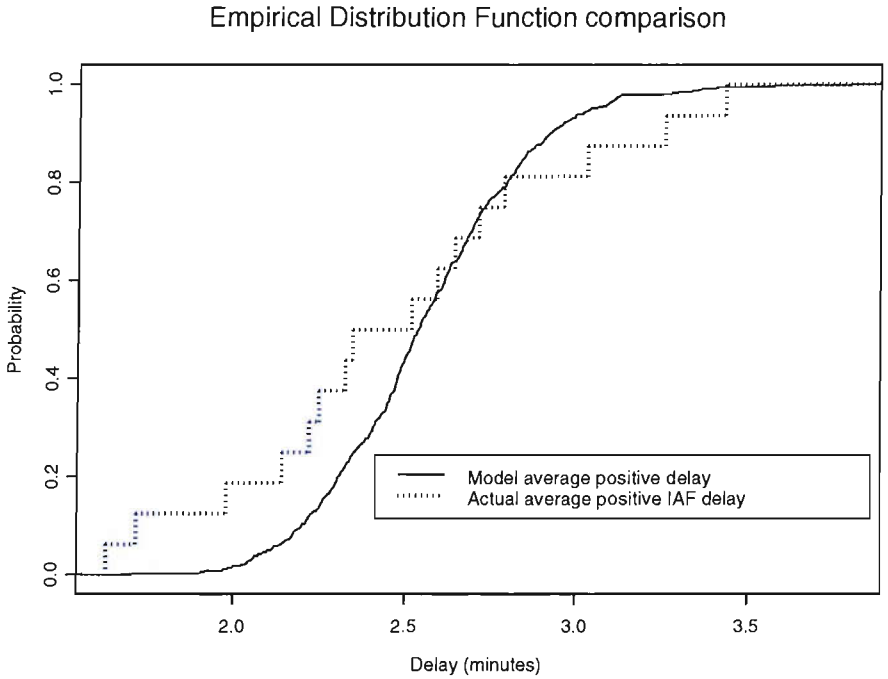


Figure 5.6: Empirical Distribution Function plots of model and actual mean positive delay at IAFs

Table 5.5: Results for Hypothesis VII, Test 2

Statistic	Value	p-value
W_2^2	0.0687	0.7665*
C_1	0.0315	0.595*
C_2	0.0201	0.364*
C_3	0.000593	0.806*
C_R	0.00064	0.903*
A_2^2	0.421	0.8175*

*2000 Bootstraps

Hypothesis VIII There is no difference between mean positive delay from the Track data, and the simulation model set with maximum likelihood estimates of inputs.

Test 1 The hypothesis that model mean delay follows a normal distribution is rejected at the 95% level by the Shapiro-Wilks test for normality. The test used to test for difference in mean delay therefore needs to not be based on this assumption. The non-parametric Mann-Whitney test statistic Z (Rice 1995) is suited to the purpose.

Results A Mann-Whitney test statistic $Z = 0.900$ results with p-value of 0.368. The null hypothesis cannot be rejected.

Test 2 Bootstrap difference in distribution means to obtain a confidence interval (Efron & Tibshirani 1998).

Result The difference in mean delay is 0.0806 and [2.5, 5, 95, 97.5]% empirical percentiles found from 5000 bootstraps are [-0.159, -0.124, 0.282, 0.319]. Both the 90% and 95% intervals cover 0. So, Hypothesis VIII cannot be rejected with 95% confidence that the true difference in mean positive delay lies between [-0.159, 0.319].

Dynamic behaviour

The tests above made use of pooling comparisons based on delay obtained at different times of the day. This section justifies the procedure.

Question Is time of day significant in determining mean delay?

Hypothesis IX Time of day is not statistically significant in determining mean positive delay at IAF points in the time periods for which Track data is available.

Test 1 Aggregate the Track delay at IAF samples by hour. Test that the true location parameter for mean delay is the same in each of the time periods against the alternative that it is different in at least one of the groups. A parametric test for this is one-way Analysis of Variance (ANOVA). A non-parametric equivalent is the Kruskal-Wallis test (Rice 1995).

Result ANOVA is invalid as a Q-Q plot of the model residuals showed they were not normally distributed. The two-sided Kruskal-Wallis $\chi^2 = 7.895$ with 11 degrees of freedom yields a p-value = 0.723. This suggests that the hypothesis that the true location parameter for mean delay is the same, for all groupings of delay by hour, cannot be rejected. In other words, there is not enough evidence to suggest time of day is statistically significant in determining mean delay for the days Track data was recorded.

5.3.2 Landing rate

Question Is there a difference between landing rate output from the model and landing rate in real life?

Confidence

Analysis of the FPL data in Figure 5.1 showed that landing rate will be a function of time. This makes checking if the distribution of landing rate from the model matches Track data more difficult since, in effect, there is a different distribution of landing rate at each moment in time.

Hypothesis X There is no difference in landing rate distribution between the model and Track data samples.

Test 1 Define landing rate $l(t)$ at time in minutes t to be the number of aircraft that landed since $l(t-60)$. Then compare a plot of landing rates from Track data with a 95% empirical range of landing rates and the mean landing rate from 500 runs of the simulation model.

Result Visual inspection of Figure 5.7 suggests that the model configured to FPL data follows the behaviour of the Track data landing rate quite well.

Test 2 At each time point in Figure 5.7 the model forms a reference distribution of landing rate from its independent runs. If the distribution of landing rate was not time dependent then independent landing rates could be pooled together and goodness-of-fit tests carried out to compare this distribution with the model distribution. In this case let Y represent the set of observed landing rates and X the set of model landing rates. One way to test if the two distributions are the same would be to test if the set $\{\Pr(y \leq X) : \forall y \in Y\}$ is distributed uniformly on [0,1]. The same idea may be applied to test whether there is no difference between the time dependent model distributions and the actual situation. Change the

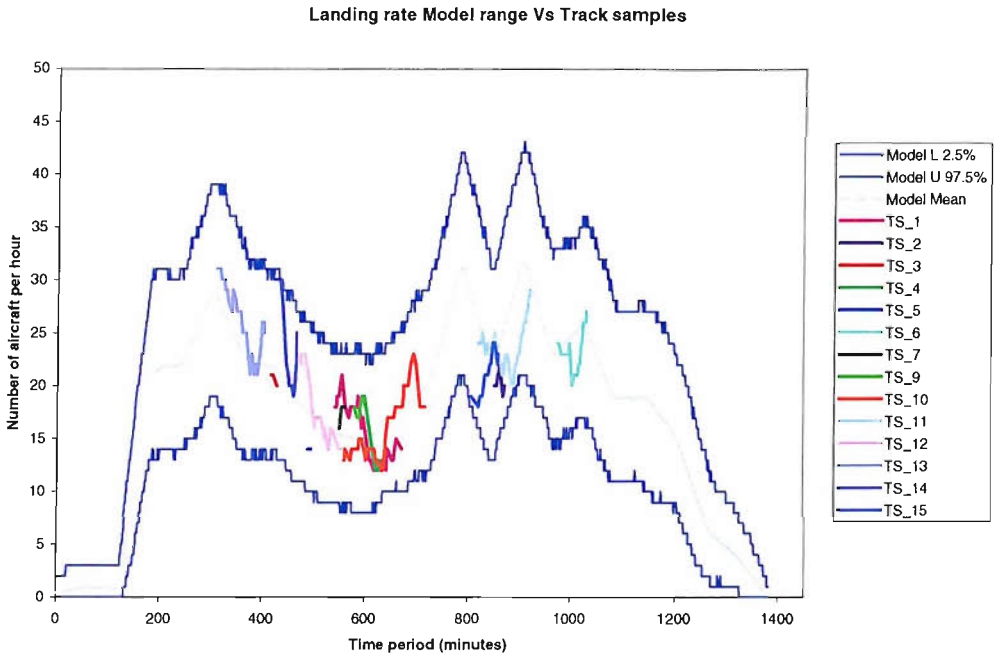


Figure 5.7: Simulation empirical 95% percentile range and mean land rate comparison with moving average Track data landing rates

Table 5.6: Results for Hypothesis X, Test 2

Statistic	Value	p-value
K^2	0.129	0.815
A^2	0.256	0.967*
χ^2	3.455	(df=6) 0.750

*2000 Bootstraps

distribution of X based on the time period t at which the observation $y \in Y$ was made and test if $\{\Pr(y_t \leq X_t) : \forall y_t \in Y_t, t \in T\} \sim U[0,1]$. Testing $U[0,1]$ may be done using goodness-of-fit tests such as Anderson-Darling A^2, Cramer-Von Mises W^2 or χ^2 statistics.

Result It was possible to form 22 independent samples of landing rates, each of length 1 hour from the Track data. These were compared to the 22 different reference distributions at the corresponding time points. The test statistics of the test $\{\Pr(y_t \leq X_t) : \forall y_t \in Y_t, t \in T\} \sim U[0,1]$ are reproduced in Table 5.6. Based on these, we do not reject the hypothesis that there is no difference in landing rate distribution between the model and landing rate contained in the Track data samples, at the 95% level.

Table 5.7: Linear model fit for Hypothesis XI, Test 1

Coefficient	Value	Std. Error	t value	$\Pr(>\|t\|)$
u	4.406	3.874	1.137	0.269
v_i	0.765	0.188	4.068	0.001

Sensitivity analysis

Question Does change in input arrival rate have the same effect on landing rate in model as in real life?

Hypothesis XI There is a positive correlation between model and actual landing rates in different time periods.

Test 1 Fit a regression line to points (v_i, w_i) paired by time where v_i is mean landing rate in the model and w_i landing rate sample i from Track data (Kleijnen 1995).

Result The points used were the 22 independent sample landing rates paired with mean landing rate from model. A fit of the one-way linear model $\mathrm{E}[w_i] = u + v_i$ where u is a constant yields the results in Table 5.7. This model fits returns residual standard error is 4.017 on 20 degrees of freedom, a multiple R^2 of 0.453 and overall F-statistic test equal to 16.55 on 1 and 20 degrees of freedom with p-value 0.001. Diagnostics on the fitted model are satisfactory: a Shapiro-Wilks normality test of residuals yields W = 0.972 with a p-value = 0.784.

The results show that the model mean and actual landing rate samples are positively correlated. The t-test rejects the hypothesis that model mean has no effect on the least squares estimated linear model at the 99% level, i.e. Hypothesis XI cannot be rejected.

Dynamic behaviour

Question Is the time series of landing rate in the model the same as in real life?

Test 1 Compare the time series plot of landing rate of a single run from the model with a single sample.

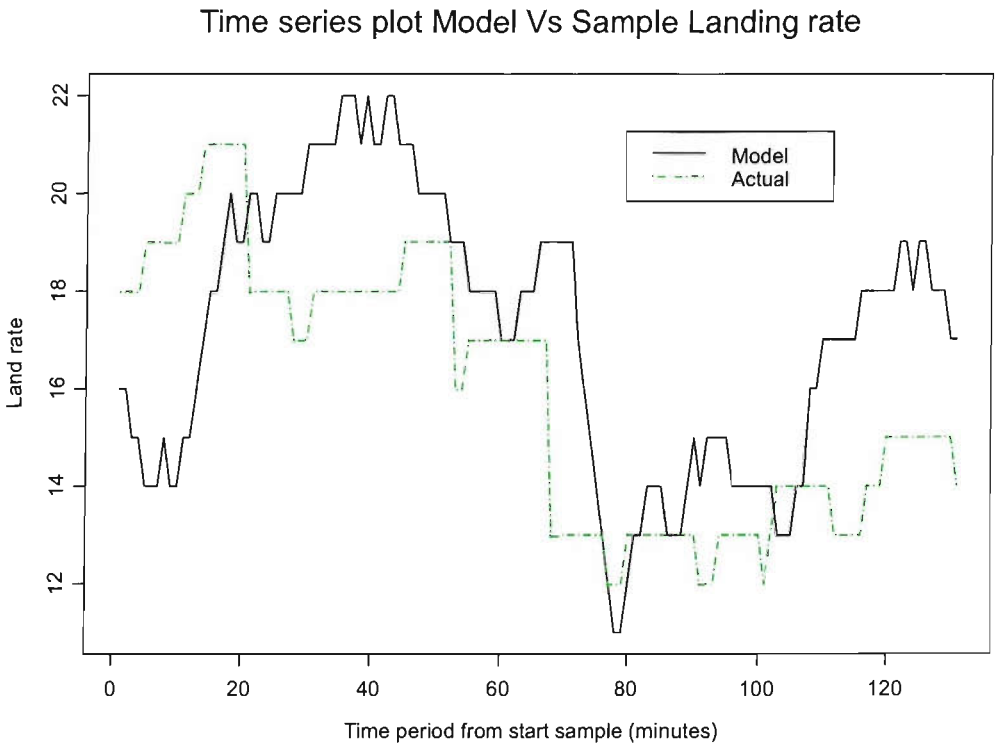


Figure 5.8: Landing rate time series plots of model run 1 and Track data sample 1

Result A traffic sample with a number of landing rate observations (i.e. one of the longer samples) is compared to a series at the same time of day from the simulation run with random seed set to 1 in Figure 5.8. Visually the two samples seem similar.

Test 2 The autocorrelation function is a device often employed in time series analysis to summarize how reliant observations are on previous observations in the series (Chatfield 1980). If the model time series output is the same as actual time series we would expect this summary to be similar for both time series.

Result Figures 5.9 and 5.10 show the two autocorrelation functions. It is difficult to see clear evidence here that the two series have different characteristics.

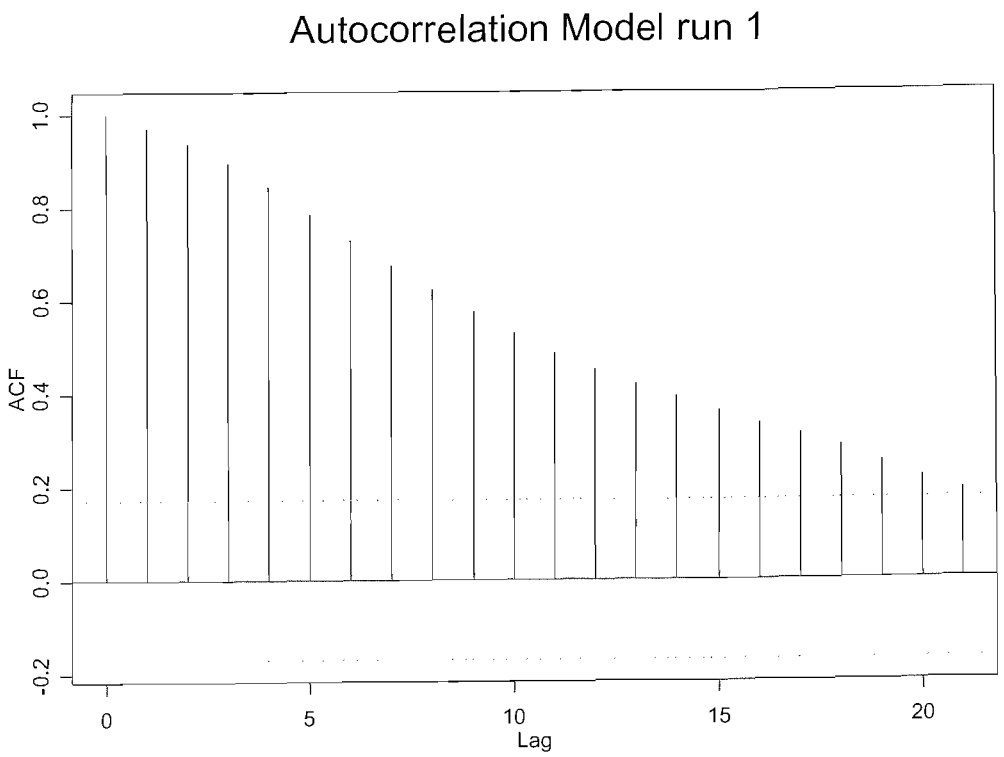


Figure 5.9: Autocorrelation plot landing rate; model

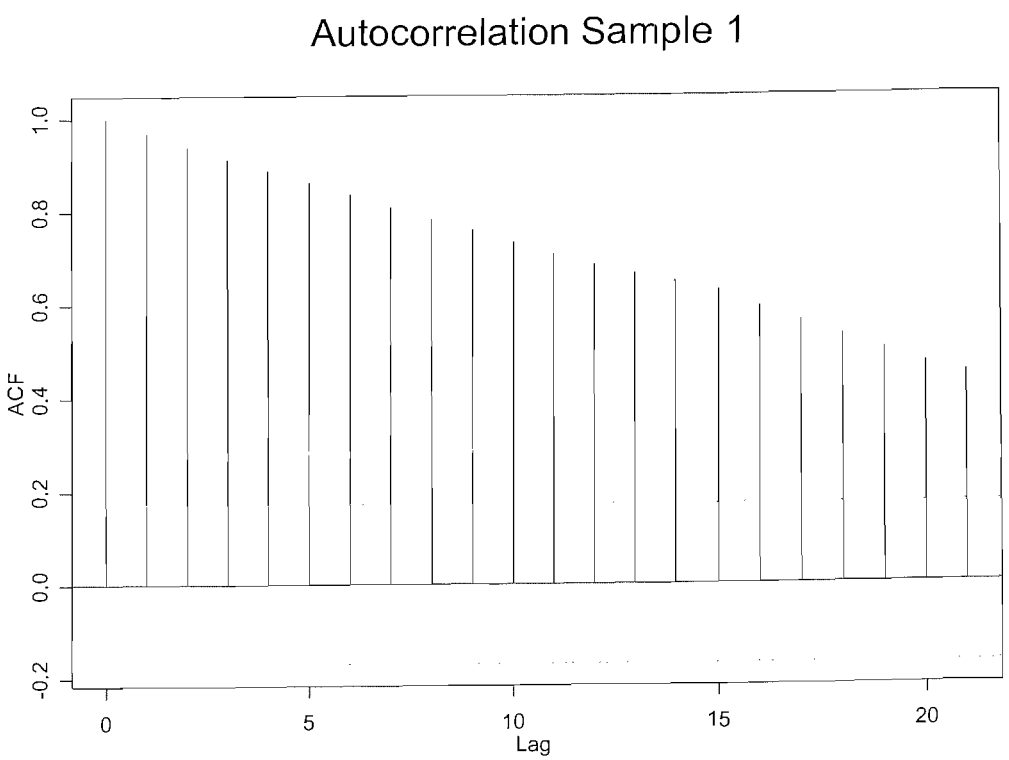


Figure 5.10: Autocorrelation plot landing rate; sample

5.4 Model evaluation

5.4.1 Confidence in input and output

A number of hypotheses for confidence in different input/output components of the model have been tested individually. However, the validation procedure is really concerned with confidence in the model as a whole.

Hypothesis XII There is no difference between model and reality.

Test 1 Hypothesis tests have been carried out to validate different components of the model against radar track data sources. Each hypothesis has a test with a p-value, or probability that the amount of variation between the simulation model and track data would be expected if the hypothesis was correct. In total there were 9 distinct tests made of null hypotheses $H_{o1}, \ldots, H_{o9}$. If H_o is the composite hypothesis that all H_{ok} are true (if one is not true H_o should be rejected) then Fishers test (D'Agostino & Stephens 1986) may be used. When H_o is true we should find that the p-values p_i of the 9 hypotheses are independent, identically distributed $U[0, 1]$ variables. This can be tested using any goodness-of-fit test statistic. Some of the hypotheses have different test statistics. In this case only one test score may be used else independence is lost. The Anderson-Darling test statistic is chosen whenever there is a choice. This is for consistency and this statistic is known to be more powerful than some others (D'Agostino & Stephens 1986).

Result An Anderson-Darling test statistic $A^2 = 1.657$. Running a Monte-Carlo simulation to find the distribution of A^2 under the null with 5,000 samples found a corresponding p-value of 0.135, so there is not enough evidence to reject the hypothesis that all H_{ok} are true.

5.4.2 Validation strengths and weaknesses

Strengths

Quantitative Because the validation is carried out on numerical comparisons between model and observed Track data it is possible to use objective methods to validate.

Statistical hypothesis tests The testing for differences follows a scientific process.

Confidence In some instances it was possible to give a confidence intervals on differences.

Weaknesses

Hypothesis tests: Data Vs Accuracy A flaw with hypothesis testing a model is that the less data that is available the more likely the hypothesis is not rejected, and the more data available the more likely to reject. This is because the hypothesis that there is no difference between model and reality should not be accepted in the long-run because by definition the model is not reality (Law & Kelton 2000).

Data Confidence in validation is limited by amount of data available. Dynamic tests and sensitivity analysis were limited by data.

Dynamic tests These were based on subjective evaluation of charts.

5.4.3 Conclusions

Sixteen Track data samples have been used to help validate a computer simulation model built to aid investigation of change to scheduling and delay-share strategy on airspace. The data was organised into databases and queries run on it to give indicators of delay and landing rate.

Validation on the sampling procedure lead to a number of conclusions.

- Tests for the hypothesis that a non-stationary Poisson process generates arrival times at IAF points all had probabilities greater than 0.05 that the variation in the test score would be seen if the model is correct.
- The probability that aircraft fly to IAF points with a multinomial distribution could not be rejected with a p-value of 0.90.
- The probability of aircraft wake vortex category at each IAF follows a multinomial model. This hypothesis was tested and all test scores had p-values greater than 0.15.

Validation on the outputs was carried out by fitting maximum likelihood estimates to sampling inputs from half of the data set provided by Eurocontrol. When comparing the results from these runs with the Track data the following conclusions were reached.

- There is not enough evidence to support the hypothesis of difference in positive mean delay from model with positive mean IAF delay from Track data. Test scores on the difference in distributions could not be rejected at the

95% level. Tests for difference of mean of mean positive delay distributions could not be rejected at the 95% level using the Mann-Whitney test, or a decomposition of the Cramer Von Mises test statistic. A 95% confidence interval of the true difference is [-0.159, 0.319].

- The hypothesis that there is no difference in landing rate could not be rejected at the 95% level.
- Landing rate in the model and the Track data follow the same trend in time. A fitted regression model for the relationship between model mean landing rate and independent track landing rates was significant at the 0.001 level with a least-squares point estimate of model mean coefficient at 0.765.
- Landing rates have similar time series characteristics. Confidence in this is based on qualitative assessment.

The overall evaluation of confidence in model output is based on a test that all the hypotheses made in this chapter are correct. This test could not be rejected at the 95% level. A summary of the conclusions in tabular form may be found in the Appendix.

Chapter 6

Simulation experiment methodology

This chapter reviews the statistical methodology used in the experiments that follow. The methodology is well known, but application to computer simulation models like the AMAN simulation requires care. Section 6.1 shows how traditional Design of Experiment methodology may be applied, and improved, in application to computer simulation models of this sort. Section 6.2 reports a variety of methods to investigate the distribution of simulation output when a single observation is taken per run. Many of the methods have been developed for smaller data sets and so their use on simulation experiment output allows certain changes to be made in their application. Taking only one observation per simulation run in terminating simulation experiments preserves independence in data observations, but some information on the underlying in-run distributions may be lost. Section 6.3 presents some methods to further investigate in-run distributions.

6.1 Setting up computer simulation experiments

Techniques employed by the classical Design of Experiment (DoE) literature have in the main been developed for physical experiments, such as agricultural or pharmaceutical trials. The experimenter has a question or hypothesis they wish to investigate by estimating how input variables affect output indicators. Efficient experimental designs are built to allow the important input effects to be estimated simultaneously. Three principles underpin design of these experiments (Box, Hunter & Hunter 1978):

Randomization Effects that are not interesting or not known may be present. Randomization guards against their potential bias by balancing them out.

Blocking Blocks of homogeneous experimental material receive identical input stimuli. The block effects may be estimated to allow interesting effects to be free of their bias.

Replication Experiment repeats increase the precision of output and the power (in the Neyman-Pearson sense (Rice 1995)) to detect significant effects.

Many simulation models may be experimented on in a similar manner to physical experiments. However, computer simulation experiments present two opportunities for gain over traditional DoE. Firstly, randomness is controllable through deterministic random-number streams. Variance reduction methods have been developed to make use of this property (Law & Kelton 2000). Secondly, the time of a simulation run may be short and so computer simulation experiments have the potential to replicate to numbers in great excess of traditional DoE. This aids precision, power to detect effects and opens doors to alternative types of analysis. The second property is of particular relevance to this work - the AMAN simulation run time is largely determined by the speed of the sequencing algorithm and may complete in as little as half a minute. This enables many repeats to be made per experimental design point.

Experimental designs are chosen with an underlying Analysis of Variance (ANOVA) model in mind (Wu & Hamada 2000):

$$\boldsymbol{Y} = \beta \boldsymbol{X} + \epsilon \quad \text{with} \quad \mathrm{E}[\boldsymbol{Y}] = \beta \boldsymbol{X} \tag{6.1}$$

where the (n x 1) response vector $\boldsymbol{Y}$ is dependent on (n x p) design matrix $\boldsymbol{X}$ of known form, the (p x 1) estimated coefficient vector β and a (n x 1) vector of errors ϵ. A common problem with ANOVA type models is over-parameterization: there are more parameters to estimate than there can be independent normal equations to solve. This means that there will be an infinite number of solutions unless extra restrictions are added (Searle 1971). Careful thought on the form of restriction is needed to make sure estimates are useful and understandable with respect to the objectives of experiment. Let β_{ij} represent level j of coefficient type i. The usual restrictions put on the model would be to set $\sum_j \beta_{ij} = 0$. This form of constraint is readily interpretable for balanced, complete designs. A general alternative to this is to set base settings of parameters to zero, e.g. $\beta_{i1} = 0 \; \forall i$. In this case each coefficient estimate of type i compares to its level 1. This makes good sense for interpretation of an experiment run to compare new treatment combinations with a base. This is especially true for simulation experiments where the interest is in comparing the validated base design point to different treatment combinations.

Given a set of experimental constraints a number of competing designs may be formulated. Choice of the design may be made on the relative importance of

coefficient estimates. For instance, some designs may not estimate all important coefficients, others may do so but have a poor underlying coefficient correlation structure rendering interpretation of results problematic. Designs are chosen to ensure that output modelling is made from as good a starting point as possible.

Once the experimental design has been run output analysis may begin. Methods used on simulation models range from observing animation, plotting dynamic behaviour to more rigorous statistical methods (Law & Kelton 2000). In each simulation run j at design point i for $k = 1 \ldots, n_{ij}$ there may be several observations z_{ijk} of interest, correlated in some way by k. Many statistical analysis methods assume independent data observations, so a common method to analyse z_{ijk} is to aggregate the observations by k into a single statistic y_{ij}. Many repeat runs per design point result in empirical distributions $Y_j = \{y_{1j}, \ldots, y_{nj}\}$ of the statistic for each design point j. Methods used to analyze these distributions are described in Section 6.2. A problem with using a single statistic y_{ij} to represent $z_{ijk} \forall k$ is loss of information: everything that happens in a run is reduced to just one statistic, usually without knowledge of its sufficiency (in the statistical sense (Davison 2003)). Statistical methods based on proportions may be used to further analyze the original observations z_{ijk} to gain more simulation information. Some relevant models of proportions are reviewed in Section 6.3.

6.2 Analysis of simulation run aggregate statistic distributions

This section considers analysis of a terminating simulation experiment. The data to be analyzed is a set of independent observations
$\boldsymbol{Y} = \{y_{ij}; i = 1, \ldots, q, j = 1, \ldots, n_i\}$ where j is the repeat made at design point i. The question to investigate is how the (M x q) experimental design matrix $\boldsymbol{X}$ of M input factors at q design points affects output data. Section 6.2.1 runs through some graphical techniques. Section 6.2.2 presents an overview of a relatively new methodology, EDFIT, that permits investigation of the significance of $\boldsymbol{X}$ on $\boldsymbol{Y}$ distributions. Alternative methods follow based on a summary of the distribution of $\boldsymbol{Y}$. Estimation routines, significance tests and bootstrap recipes used in this work are given. Sections 6.2.3 and 6.2.4 look to analysis on $\mathrm{E}[\boldsymbol{Y}]$, Section 6.2.5 on $\mathrm{Var}(\boldsymbol{Y})$ and Section 6.2.6 gives some general-purpose methods to investigate other distribution summaries such as skewness.

6.2.1 Graphical distribution analysis

The scatter diagram is a basic graphical technique often used to visualize experiment output. When many replicates are made per design point it is more useful to summarize information from scatter diagrams in design point Empirical Distribution Functions (EDFs). Analysis may then be made to see if one design point stochastically dominates another, or if differences exist in distribution summaries such as mean. An alternative to EDF inspection is to further compact the distributions using design point boxplots or histograms (Rice 1995). Histograms are more applicable to discrete data and boxplots to continuous. It is perhaps easier to see differences in location, variance and shape of distribution using boxplots and histograms than EDFs.

6.2.2 The EDFIT method of distribution analysis

Overview

The EDF may be used as a graphical technique, but is further put to use in the Empirical Distribution Integral Test (EDFIT) methodology developed by Cheng & Jones (2000) and Cheng & Jones (2004). This statistical method aims to find differences between design point EDFs in a similarly structured way to Analysis of Variance. If a difference is found, the methodology provides an indication as to whether the difference is in location, variance or shape. Let an independent set of n_i runs be made at each experimental design point $i = 1, \ldots, q$, then define standardized EDFs:

$$Y_i^*(u) = \sqrt{n_i}(S_i^*(u) - \bar{S}^*(u)) \quad i = 1, 2, \ldots, q \tag{6.2}$$

where

$$S_i^*(u) = \frac{\sum_{j=1}^{n_i} I_{[0,u]}(s_{ij})}{n_i}, \; 0 < u < 1 \tag{6.3}$$

and

$$I_{[0,u]}(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq u \\ 0 & \text{otherwise.} \end{cases}$$

Since EDFIT uses standardized EDFs it is distribution free. A linear model analogue for standardized EDFs is made in Cheng & Jones (2004). Let $F_i(y)$ be the discrete cumulative distribution function (CDF) of scaled ranks of observations at

the i^{th} design point and:

$$\boldsymbol{F}(u) = (F_1(u), F_2(u), \ldots, F_q(u)).$$

Then the EDF linear model is:

$$\boldsymbol{F}(u) = \boldsymbol{A}\boldsymbol{b}(u)$$

where

$$\boldsymbol{A} = [\mathbf{1_q}|\boldsymbol{A}_1]$$

$$\boldsymbol{b}(u) = \begin{pmatrix} b_0(u) \\ \boldsymbol{b_1}(u) \end{pmatrix}$$

where $\boldsymbol{b}(u)$ is a column vector of unknown functions. Cheng & Jones (2004) derive an estimator for $\boldsymbol{b}_1(u)$ and a test for the null hypothesis $H_0 : \boldsymbol{b}_1(u) = \mathbf{0}$. Cheng has also developed tests for significance of individual components $b_i(u)$ and uses a Fourier analysis to break this test down into tests correlated to mean, variance and skewness. Since this is not published at the time of writing his work is presented next for completeness.

Cheng decomposition of b

First, define matricies as Cheng & Jones (2004). Denote $r = n_i/n$, $J = \mathbf{1}_q$, $D = \sqrt{r}\boldsymbol{I}_q$ where $\boldsymbol{I}_q$ is the $q \times q$ identity matrix and $p = r\mathbf{1}_q$. Set $B = D(\boldsymbol{I}_q - \mathbf{1}_q p^T)A_1$ and $G = (B^T B)^{-1}$. The estimate of $\boldsymbol{b}_1(u)$, $\hat{\boldsymbol{b}}_1(u) = (1/\sqrt{n})GB^T Y^*(u)$. Denote $S(u) = B\hat{\boldsymbol{b}}_1(u)$. Then the EDFIT test statistic $T^2 = S^T(u)S(u)$ is used to test the hypothesis $\boldsymbol{b}_1(u) = \mathbf{0}$.

Assuming that $\boldsymbol{b}_1(u)$ is not null the individual components of $\boldsymbol{b}_1(u)$ can be considered. If the experiment is unbalanced or incomplete then $B^T B$ is not diagonal. In this case we cannot, strictly speaking, test for each component independently. Thus if we are testing if some particular $b_i(u) = 0$, this should be conditional on the others also being fitted. However we might consider the fitting of each $b_i(u)$ as if it had been fitted on its own using the expressions for $\hat{b}_i(u)$. All the $\hat{b}_i(u)$ are of the form

$$\hat{b}_i(u) = \sum_{j=1}^{q} b_{ij} Y_j^*(u)$$

(where $\sum_{j=1}^{q} b_{ij} = 0$). We can therefore replace each $Y_j^*(u)$ by its Fourier decomposition using $1/\sqrt{rn}Y_j^*(k/n) = (S_j^*(k/n) - k/n) = \sum_{l=1}^{n} \hat{c}_{lj} \sin(l\pi k/n)$, say. We have that (Cheng & Jones 2004)

$$\hat{c}_{lj} = \frac{2}{n} \sum_{k=1}^{n} \frac{1}{\sqrt{rn}} Y_j^*(k/n) \sin(l\pi k/n).$$

Thus

$$\begin{aligned}
\hat{b}_i(\frac{k}{n}) &= \sum_{j=1}^{q} b_{ij} Y_j^*(k/n) \\
&= \sum_{j=1}^{q} b_{ij} [\sqrt{rn} \sum_{l=1}^{n} \hat{c}_{lj} \sin(l\pi k/n)] \\
&= \sum_{l=1}^{n} \sqrt{rn} \sum_{j=1}^{18} b_{ij} \hat{c}_{lj} \sin(l\pi k/n) \\
&= \sum_{l=1}^{n} C_{li} \sin(l\pi k/n)
\end{aligned}$$

where

$$C_{li} = \sqrt{rn} \sum_{j=1}^{q} \hat{c}_{lj} b_{ij}$$

Suppose the function $f(k/n)$ has Fourier expansion: $f(k/n) = \sum_{j=1}^{n} \hat{\beta}_j \sin(j\pi k/n)$. (Note that we must have $\hat{\beta}_n = 0$.) From Parseval's Theorem

$$\frac{1}{n} \sum_{k=1}^{n} f^2(\frac{k}{n}) = \frac{1}{2} \sum_{l=1}^{n} \hat{\beta}_l^2.$$

In our case

$$\frac{1}{n} \sum_{k=1}^{n} \hat{b}_i^2(\frac{k}{n}) = \frac{1}{2} \sum_{l=1}^{n} C_{li}^2$$

We can therefore examine the distribution of each of the C_{li}^2 $l = 1, 2, 3$ which effectively and respectively measure the mean, variance and skewness of $\hat{b}_i(k/n)$, and look at

$$R_i^2 = \frac{1}{n} \sum_{k=1}^{n} \hat{b}_i^2(\frac{k}{n}) - C_{1i}^2 - C_{2i}^2 - C_{3i}^2$$

for the remainder of the function, $\hat{b}_i(k/n)$.

EDFIT Tables

The Cheng decomposition of $\boldsymbol{b}$ permits similar summary tables to be made to those produced by statistical packages for linear model results. Individual coefficients are tested for overall significance in distribution with T^2 statistics, and for difference in mean, variance and skewness with Fourier component test scores C_i.

A first step in linear model fitting is often to test coefficient types with an ANOVA table. Similar EDFIT one-way and two-way classification tests are shown in both Cheng & Jones (2000) and Cheng & Jones (2004). A more general method of building EDFIT tables when the experimental design may be unbalanced or incomplete is described here. In this situation the covariance matrix of coefficients $\boldsymbol{b}$ is non-orthogonal. It is therefore not possible to decompose ANOVA sums of squares, or EDFIT Test statistics into separate components for each factor. Differencing is used in ANOVA tables for this problem (Searle 1971). The EDFIT test statistic, T^2 is non-decreasing as the number of coefficients in the EDF linear model increases. It is therefore reasonable to apply the same idea.

Let β_{ij} denote coefficient j of variable type i. Denote the increase in test statistic T^2 due to type i coefficients be $R[\beta_i|\beta_{i-1},\ldots,\beta_1]$ and the test statistic when all coefficient types to β_i are included be $R[\beta_i,\beta_{i-1},\ldots,\beta_1]$. Then:

$$R[\beta_i|\beta_{i-1},\ldots,\beta_1] \;=\; R[\beta_i,\beta_{i-1},\ldots,\beta_1] - R[\beta_{i-1},\ldots,\beta_1]$$

is a test statistic for fitting β_i type coefficients conditional on $\beta_{i-1},\ldots,\beta_1$ being fitted. As noted in Searle (1971), the order of testing significance of β_i may affect the significance of effect detection.

To obtain a reduced EDFIT table with test scores broken down into components requires a method to aggregate the individual component scores. Now, the individual components of $\boldsymbol{S}(u)$, $S_j(u)$ say, have the same form as the $\hat{b}_i(u)$ above. That is, all the $S_j(u)$ are of the form

$$S_j(u) = \sum_{k=1}^{q} s_{jk} Y_k^*(u)$$

where $\sum_{k=1}^{q} s_{jk} = 0$. Following along the lines of the Cheng decomposition of $\hat{b}_i(u)$:

$$\frac{1}{n}\sum_{l=1}^{n} S_j^2(\frac{l}{n}) = \frac{1}{2}\sum_{m=1}^{n} C_{mj}^2.$$

Table 6.1: Generic EDFIT table layout example

Coefficient	T^2	C_1	C_2	C_3	C_R
β_1	$\mathbf{e_{11}(p_{11})}$	$\mathbf{e_{12}(p_{12})}$	$e_{13}(p_{13})$	$e_{14}(p_{14})$	$e_{15}(p_{15})$
$\vdots$					
β_M	$e_{M1}(p_{M1})$	$e_{M2}(p_{M2})$	$e_{M3}(p_{M3})$	$e_{M4}(p_{M4})$	$e_{M5}(p_{M5})$

The distribution of various C^2_{mj} may therefore test difference components correlated to mean, variance and skewness of each $S^2_j(u)$. The overall test $T^2 = \sum_{j=1}^{q} S^2_j(u)$ and so a test statistic for each Fourier decomposition term of T^2 is $C^2_m = \sum_{j=i}^{q}(1/2)C^2_{mj}$. This may test for difference in mean, variance and skewness of T^2. The C^2_m statistics are also non-decreasing as number of coefficients increases. They may be used in an EDFIT table in the same manner as T^2 statistics.

Presentation of EDFIT tables

Two types of EDFIT table may be formed: a reduced EDFIT table testing significance of coefficient types, and a full EDFIT table testing significance of individual coefficients. Both types are presented in this work with the basic layout of Table 6.1. In a reduced EDFIT table the order of significance tests is the descending order of the table. So in Table 6.1 coefficient type β_1 is added first. In a full EDFIT Table the order is not important. In both tables column labels refer to the EDFIT statistics for overall test statistic T^2, the Fourier decompositions C_1 - C_3 and the remainder C_R. Actual statistics are given as e_{ij} and corresponding p-values p_{ij} in brackets. If a term is significant at the 0.02 level it is highlighted in bold type. For example, if Table 6.1 is a reduced EDFIT table then coefficient type β_1 is significant in overall distribution and a difference is found in component 1, correlated to mean of the distribution.

Calculation of EDFIT significance levels

The EDFIT procedure is distribution-free so test statistic distributions under the null may be calculated using Monte-Carlo simulation. For any particular EDF linear model (i.e. $\boldsymbol{A}$ matrix) the distribution of T^2 and individual β_{ij} terms may be calculated to arbitrary accuracy with a total of B, say, simulations. More simulation work is required for a reduced EDFIT table because reference distributions for each coefficient type i need to be estimated. Each $R[\beta_i|\beta_{i-1}, \ldots, \beta_1]$ test statistic is compared to a Monte-Carlo distribution formed using an $\boldsymbol{A}$ matrix with only the β_{ij} $(j = 1, \ldots, k_i)$ columns active. If c is the maximum number of coefficient types

to be included in an EDFIT table and B Monte-Carlo replicates are made for each coefficient type then the total number of Monte-Carlo simulations run is cB.

6.2.3 Metamodelling mean from a continuous distribution

If difference in distribution is found using graphical techniques or EDFIT, it may be interesting and informative to fit models to distributional summaries such as the mean. Such a model is used in regression analysis, where the linear model is of the form of Equation (6.1). There follow some methods of estimation and significance testing when some of the following three assumptions are made.

1. $\mathrm{E}(\epsilon) = 0$. This is implicit in model (6.1).
2. $\mathrm{Var}(\epsilon) = \boldsymbol{I}\sigma^2$, or independent errors with constant variance.
3. $\epsilon_i \sim N(0, \sigma_i^2)$.

Ordinary Least Squares estimation

The estimator that arises from minimizing error sum of squares $\epsilon'\epsilon$ is $\hat{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$. $\hat{\beta}$ has the following properties when the above assumptions hold (Draper & Smith 1998).

1 only $\hat{\beta}$ is an unbiased estimator of β.

1 and 2 $\hat{\beta}$ is the minimum variance unbiased estimator of β.

1, 2 and 3 $\hat{\beta}$ is the maximum likelihood estimator. Decisions on significance of regression and individual coefficients may be made based on the standard asymptotic theory.

Generalized and Weighted Least Squares estimation

Suppose that $\mathrm{Var}(\epsilon) = \boldsymbol{V}$. Then the generalized least square estimator $\tilde{\beta} = (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{Y}$. Assuming $\mathrm{Var}(\epsilon) = \boldsymbol{V}$ is known then $\tilde{\beta}$ is the minimum variance linear unbiased estimator of β (Draper & Smith 1998). If $\boldsymbol{V}$ is a diagonal matrix, observations are independent with different variance, and $\tilde{\beta}$ is a Weighted Least Squares (WLS) estimate. With computer simulation experiments the form of $\boldsymbol{V}$ may be non-diagonal. For example, if variance-reduction techniques such as common random number streams between runs at different design points are used then independence between observations is compromised (Banks 1998). However, it may be reasonable to ignore these correlations if model errors appear to be random. Variance might be expected to vary by design point and so the Weighted Least Squares estimation may be more appropriate than OLS.

Generalized least square procedures for WLS problems are well developed. Models for the mean are fitted with the weights in $\boldsymbol{V}$ based on a model of variance with estimated parameters $\boldsymbol{\theta}$. If the variance model is dependent on the model for mean (e.g. variance proportional to a power θ of the mean response) then the algorithm becomes recursive with estimates of $\boldsymbol{\beta}$ used to update those of $\boldsymbol{\theta}$ and visa versa, until some kind of convergence criteria is satisfied (Carroll & Ruppert 1988).

Carroll & Ruppert (1988) do not advocate estimating a diagonal $\boldsymbol{V}$ using the inverse of design point sample variance as weights. This is because in the traditional design of experiment world few repeats are made at individual design points, so sample variance's have very small degrees of freedom, making them unstable. However, in simulation experiments when large numbers of repeats are made at each design point it is often safe to estimate $\boldsymbol{V}$ using the inverse of design point sample variance. This has the added benefit of removing concerns about the form of variance model.

Transformations

Transformations of the response aim to rescale the original data so that assumptions 1 to 3 may hold and standard results used. A class of transformation often used is the Box-Cox family (Draper & Smith 1998).

Resampling routines to test significance

Resampling routines may be used to find confidence intervals on parameters $\boldsymbol{\beta}$ (or test for significance) when assumption 3 does not hold. One method in use for ANOVA-type models is to resample the residuals ϵ to form new values of $\boldsymbol{Y}^*$ (Davison & Hinkley 1997). This process makes assumptions 1 and 2. An alternative approach applicable to experiments with a large number of repeats at each design point is to resample observations by their design point EDF. Let Y_{ij} denote observation j at design point i, where $j = 1, \ldots, n_i$ and $i = 1, \ldots, q$. Since OLS only requires assumption 1 to be unbiased the following bootstrap algorithm could be used whatever the form of $\boldsymbol{V}$.

Resampling OLS For $r = 1, \ldots, B$

1. For $i = 1, \ldots, q$ sample $b^*_{i1}, \ldots, b^*_{in_i}$ randomly with replacement from $\{1, 2, \ldots, n_i\}$.
2. For $i = 1, \ldots, q$ and $j = 1, \ldots, n_i$ set $Y^*_{ij} = Y_{ib^*_{ij}}$ then

3. Fit OLS regression coefficients $\hat{\beta}_r^* = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}^*$.

This bootstrap also has the advantage that original observations are resampled, rather than residuals. If residuals are independent but do not have constant variance then an equivalent WLS bootstrap can be used. This will produce tighter confidence intervals of β than the previous bootstrap. Let $\boldsymbol{S}^2(\boldsymbol{Y})$ denote the (n x n) matrix with diagonal entries equal to the inverse design point sample standard deviation of the corresponding design points.

Resampling WLS For $r = 1, \ldots, B$

1. For $i = 1, \ldots, q$ sample $b_{i1}^*, \ldots, b_{in_i}^*$ randomly with replacement from $\{1, 2, \ldots, n_i\}$.
2. For $i = 1, \ldots, q$ and $j = 1, \ldots, n_i$ set $Y_{ij}^* = Y_{ib_{ij}^*}$ and $\tilde{\boldsymbol{V}}^* = \boldsymbol{S}^2(\boldsymbol{Y}^*)$ then
3. Fit WLS regression coefficients $\tilde{\beta}^{*r} = (\boldsymbol{X}'\tilde{\boldsymbol{V}}^{*-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\tilde{\boldsymbol{V}}^{*-1}\boldsymbol{Y}^*$.

6.2.4 Metamodelling mean for count data

Models and fitting routines in Section 6.2.3 may be used with count data if the assumption is made that the data is reasonably approximated by a continuous response. However, it is preferable to use a model that suits the count data. One such model is a log-linear model.

Log-linear models

Log-linear models where $\ln(\mathrm{E}(Y_i)) = \beta \boldsymbol{X_i}$ with $\mathrm{Var}(Y_i) = \sigma^2 \mathrm{E}(Y_i)$ for $i = 1, \ldots, n$ may be built and estimated using Generalized Linear model routines (Davison 2003). The model assumes a constant coefficient of variation, i.e. variance proportional to mean. Also assumed is that known maximum count values do not exist - they are not specifically included in the model. If it is further assumed that $\sigma^2 = 1$ then the log-linear model may be thought of as a Poisson model for counts with mean $\mu_i = \exp(\beta \boldsymbol{X_i})$. This assumption may be checked using analysis of deviance: if $\sigma^2 = 1$ the residual deviance should be approximately equal to the degrees of freedom in the model (Myers, Montgomery & Vining 2002). Otherwise, the more general model still holds and an estimate of σ^2 is used in the β covariance matrix to test significance.

6.2.5 Metamodelling variance

A metamodel of variance details how regressors $\boldsymbol{X}$ affect variance of output $\mathrm{Var}\,(Y_i)$. Generalized least squares routines may be used to estimate variance models of this form, fitting to residuals from the mean model (Carroll & Ruppert 1988). Alternatively, models could be fitted to $(Y_{ij} - \overline{Y}_i)$, where $\overline{Y}_i$ represents sample mean at design point i (Goos, Tack & Vandebroek 2001). This approach allows the variance model to be estimated independent to a model for mean. Many forms of variance model exist. In this work the Box & Meyer (1986) form is used:

$$\mathrm{Var}\,(Y_i) \;=\; \sigma^2 \exp(\boldsymbol{\theta} \boldsymbol{X_i}) \tag{6.4}$$

for $i = 1, \ldots, n$, where σ^2 is a scale parameter, $\boldsymbol{\theta}$ an unknown (n x 1) parameter vector and $\boldsymbol{X_i}$ the i^{th} column of $\boldsymbol{X}$, a (p x n) design matrix. This form of model is chosen because it guarantees that $\mathrm{Var}\,(Y_i)$ is a positive quantity, while explaining the variation of output in terms of the regressors $\boldsymbol{X}$. Carroll & Ruppert (1988) present methods to estimate the parameters $\boldsymbol{\theta}$ in this and other forms of variance model. A number of competing techniques exist. For instance, a log transformation on both sides yields:

$$\ln(\mathrm{Var}\,(Y_i)) \;=\; \boldsymbol{\theta} \boldsymbol{X_i} + \ln(\sigma^2) \tag{6.5}$$

and a bias-corrected WLS estimiation procedure may be used. An alternative is to assume normality of mean model residuals, or $(Y_{ij} - \overline{Y}_i)$, and then apply maximum likelihood directly to model (6.4). These estimates may be calculated using a non-linear least squares routine. It has been shown that maximum likelihood estimates are asymptotically equivalent to bias-corrected WLS estimates, and other pseudolikelihood and Restricted Maximum Likelihood (REML) estimates, even when the normality assumption does not hold (Davidian & Carroll 1987). If the model for variance does not depend on the mean, as in model (6.4), then the estimates $\hat{\boldsymbol{\theta}}$ are asymptotically normal with mean $\boldsymbol{\theta}$. Their covariance matrix depends on whether residuals from a model for mean or $(Y_{ij} - \overline{Y}_i)$ are used. Denote $\boldsymbol{v}$ as a vector of length q, where $\boldsymbol{v} = \boldsymbol{\theta X} + \ln(\sigma^2)$, $\boldsymbol{v}_\theta = \delta \boldsymbol{v} / \delta \boldsymbol{\theta}$ and $\boldsymbol{\xi}(\boldsymbol{\theta})$ as the covariance matrix of $\boldsymbol{v}_\theta$. Then, assuming the model for mean is correct and squared residuals from the model (with kurtosis κ) are used to fit, $\hat{\boldsymbol{\theta}}$ has covariance $(2 + \kappa)\{4n\boldsymbol{\xi}(\boldsymbol{\theta})\}^{-1}$. If the model is fitted to $(Y_{ij} - \overline{Y}_i)$ (with kurtosis κ) and m repeats are made per design point then $\hat{\boldsymbol{\theta}}$ has covariance $\{(2 + \kappa) + 2/(m-1)\}\{4n\boldsymbol{\xi}(\boldsymbol{\theta})\}^{-1}$ (Carroll & Ruppert 1988).

Note from the two forms of covariance that using the residuals from a model for the mean is more efficient than using sample means. However, as $m \to \infty$ the relative loss in efficiency $\to 0$. So, if m is large enough there is more concern about the selected model for mean than there is about efficiency. In a simulation context when m is large, fitting to $(Y_{ij} - \overline{Y}_i)$ is thus preferable, unlike traditional use of such models.

6.2.6 Metamodelling of further distribution summaries

Covariates may have no effect on mean or variance of a distribution yet still account for difference in other distribution summaries such as skewness or kurtosis. The analysis of distribution methods in Section 6.2.2 may point to such systematic differences. If so, a model may be built to explain the distribution summaries in terms of $\boldsymbol{X}$.

Consider skewness γ. An attempt to make use of all the data might fit a linear model to $Z_{ij} = ((Y_{ij} - \overline{Y}_i)^3)/(S_i^2)^{3/2}$ where S_i^2 is the sample variance at design point $i = 1, \ldots, q$, and $j = 1, \ldots, n_i$. This model would have $\mathrm{E}[Z_i] = \gamma_i = \boldsymbol{\beta X}_i$. However, if $Y_{ij} \sim N(\mu_i, \sigma_i^2)$ then Z_{ij} is not normal and so standard theory revolving around this assumption fails. Also, it is difficult to know what form $\boldsymbol{V}$ might take for data transformed in such a way. The OLS bootstrap algorithm could, in principle, be used anyway since the OLS estimates are unbiased. However, the estimates returned are likely to be quite variable, reducing power to detect. A linear model fit to sample skewness at each design point overcomes these problems at the expense of collapsing the number of observations from n to q. The Central Limit Theorem shows that the distribution of sample skewness is approximately normal, so standard theory may well be applicable. If the normality assumption fails, resample routines may be used to test for significance. The following use skewness but other summaries such as kurtosis could similarly be used.

Resampling skewness γ with OLS For $r = 1, \ldots, B$

1. For $i = 1, \ldots, q$ sample $b_{i1}^*, \ldots, b_{in_i}^*$ randomly with replacement from $\{1, 2, \ldots, n_i\}$.
2. For $i = 1, \ldots, q$ set $\overline{Y}_i^* = 1/n_i \sum_{j=1}^{n_i} Y_{ib_{ij}^*}$ and $S_i^{*2} = 1/(n_i - 1) \sum_{j=1}^{n_i} (Y_{ib_{ij}^*} - \overline{Y}_i^*)^2$ and $\gamma_i^* = (1/(n_i - 1) \sum_{j=1}^{n_i} (Y_{ib_{ij}^*} - \overline{Y}_i^*)^3)/(S_i^{*2})^{3/2}$ then
3. Fit OLS regression coefficients $\hat{\boldsymbol{\beta}}_r^* = (\boldsymbol{X'X})^{-1}\boldsymbol{X'}\gamma^*$.

This routine may run into problems if correlation exists between individual observations at different design points. Variance and mean models may be quite robust to ignoring such pairs, but higher moments such as skewness or kurtosis are heavily dependent on the extreme values of a distribution. If the observed extreme values are correlated through random number stream then confidence intervals based on the routine above will not be symmetric about the original point estimate, affecting power. Assuming that the same number of observations n_i^d are run per paired design point, and all observations in the paired design points are paired, then the following bootstrap may be used.

Resampling skewness γ with OLS, accounting for correlation between design points
For $j = 1, \ldots, d$ let D_j define the j^{th} set of design points with correlated observations and $f(i)$ give the set index j of D that design point i belongs to. Let n_j^d observations belong to each correlated group. Then for $r = 1, \ldots, B$

1. For $j = 1, \ldots, d$ sample $b_{j1}^*, \ldots, b_{jn_j^d}^*$ randomly with replacement from $\{1, 2, \ldots, n_j^d\}$.
2. For $i = 1, \ldots, q$ set $\overline{Y}_i^* = 1/n_i \sum_{j=1}^{n_i} Y_{ib_{f(i)j}^*}$ and
   $S_i^{*2} = 1/(n_i - 1) \sum_{j=1}^{n_i^d} (Y_{ib_{f(i)j}^*} - \overline{Y}_i^*)^2$ and
   $\gamma_i^* = 1/(n_i - 1) \sum_{j=1}^{n_i} (Y_{ib_{f(i)j}^*} - \overline{Y}_i^*)^3/(S_i^{*2})^{3/2}$ then
3. Fit OLS regression coefficients $\hat{\boldsymbol{\beta}}_r^* = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\gamma}^*$.

6.3 Some methods to examine in-run distributions

6.3.1 Examining tails by setting thresholds

The methods above use a single statistic y_{ij} to represent the potentially correlated in-run output $z_{ijk} \forall k$. This is done to make sure the data analysed is independent. However, if in-run outputs are only aggregated by a single statistic, such as an average, some information may be lost. For example, underlying output distributions Z_1 and Z_2 for two treatments may not be different in means $\overline{z}_1$ and $\overline{z}_2$, yet Z_1 still have a longer tail than Z_2. This may be important. One way to look at the tail behaviour of the Z_i regardless of their distribution and still take a single observation per design point is to set a threshold T and record the proportion of data points D that exceed T in each run. Failure C might then be defined as a binary variable where

$$C = \begin{cases} 1 & D > 0 \\ 0 & D = 0. \end{cases}$$

Tail behaviour may be analyzed by estimating how covariates X affect $Pr(D > 0)$ by building logit models. The rational behind logit models and some problems associated with them are next discussed.

Logit models

Let Y_{ij} follow a Bernoulli distribution taking values 1 or 0 with probability π_i and $1 - \pi_i$ at design point i for $j = 1, \ldots, n_i$. Then Y_i is binomially distributed with index n_i and parameter π_i. The usual linear model assumptions for $Y_i = \boldsymbol{\beta X_i} + \epsilon_i$ do not hold for this response distribution (McCulloch & Searle 2001). However, a link function may transform the non-linear Y_i to a linear scale. One such link function is the logistic transformation $\ln[\pi_i/(1 - \pi_i)]$, or equivalently $\pi_i = 1/(1 + \exp(-\boldsymbol{\beta X_i}))$. This is interpreted as the log of the odds of success.

Logit models fall into the class of Generalized Linear models (McCullagh & Nelder 1983). A Weighted Least Squares algorithm may be used for these models to obtain maximum likelihood estimates $\hat{\boldsymbol{\beta}}$. It is known that asymptotically $\hat{\boldsymbol{\beta}}$ is normally distributed (as $n \to \infty$ each element $n_i \to \infty$ in constant proportion, number of distinct binomial observations N and number of parameters p is fixed). Walds test for significance of $\hat{\boldsymbol{\beta}}$ is made assuming its asymptotic form. Asymptotic confidence intervals of $\hat{\boldsymbol{\beta}}$ may similarly be constructed. Alternative confidence intervals may be formed by the likelihood ratio statistic or score statistic (Azzalinin 1996). These are more complicated to compute (Cox & Hinkley 1974). Some problems are known to occur with logit models, in particular results based on the asymptotic theory may be misleading for a number of cases. For example, if $\pi_i = 1$ or 0 whenever coefficient $\hat{\beta}_j$ is active then the estimated $|\hat{\beta}_j| \to \infty$ so $\hat{\beta}_j$ cannot be reliably estimated and maximum likelihood estimates do not exist (Santner & Duffy 1989). Other problems are discussed in Hauck & Donner (1977), Azzalinin (1996) and Myers et al. (2002). A simulation experiment with many repeat runs per design point presents an opportunity to use resampling to check the asymptotic results, and perhaps better understand them.

Resampling logit regression model For $r = 1, \ldots, B$

1. For $i = 1, \ldots, q$ sample $b^*_{i1}, \ldots, b^*_{in_i}$ randomly with replacement from $\{1, 2, \ldots, n_i\}$.
2. For $i = 1, \ldots, q$ and $j = 1, \ldots, n_i$ set $Y^*_{ij} = Y_{ib_{ij}}$ then
3. Fit logit regression coefficients $\hat{\boldsymbol{\beta}}^*_r$ using a weighted least squares routine.

6.3.2 Thresholds with sensitivity limits

Other definitions of success and failure may also be formed. Here the idea of a sensitivity S in the proportion of data points allowed to exceed the threshold is introduced. If the threshold T is set and the proportion of data points D that exceed T recorded, then failure C is redefined as:

$$C = \begin{cases} 1 & D > S \\ 0 & D \leq S. \end{cases}$$

6.3.3 Targets

The sensitivity idea may be taken one step further if T viewed as a target that one seeks to make most of the time. It is expected that T will be exceeded and interest is in how covariates $\boldsymbol{X_i}$ affect the distribution θ_i of the proportion of data that exceed T at design point i. A parametric model is developed to analyze this. The proportion of data that exceeds T at design point i is modeled as a Bernouilli variable with probability $1 - \pi_i$, as above. A beta distribution with shape parameters s_{i1} and s_{i2} is used to model θ_i, the proportion of data that exceed T at design point i. Three potential methods to investigate how covariates $\boldsymbol{X_i}$ affect the distribution θ_i follow.

Approach 1 Generalized linear models. A link function makes the response linear, and permits prediction of the beta expected value $\mathrm{E}[\theta_i]$. A disadvantage of this approach is that it is limited to detecting difference in expected value of the beta distribution.

Approach 2 Maximum likelihood estimates $\hat{s}_{i1}$ are $\hat{s}_{i2}$ are asymptotically normally distributed. Weighted least squares with variance estimated from the standard error of maximum likelihood estimates may be used to regress design matrix $\boldsymbol{X}$ against $\hat{s}_1$ and $\hat{s}_2$. That is,

$$\begin{aligned} \hat{s}_1 &= \beta_1 \boldsymbol{X} + \epsilon_1 \\ \hat{s}_2 &= \beta_2 \boldsymbol{X} + \epsilon_2. \end{aligned}$$

The parameter maximum likelihood estimates of the beta distribution are sufficient statistics that explain everything about the distribution. A way to test for change in distribution is therefore the t-test for significant covariates. This test does not explain what part of the distribution they change. But mean, variance and skewness regression models may also be estimated using $\hat{s}_1$ and $\hat{s}_2$, and similar t-tests be made.

Approach 3 A final alternative is to somehow directly estimate $\hat{s}_1$ and $\hat{s}_2$ based on $\beta_1 X$ and $\beta_2 X$.

Use of these methods to investigate θ results in a clearer understanding of the proportion of data above threshold T, when there are data above the threshold. This is useful to determine whether once a threshold is broken, it is broken by a large proportion of the data, or only by a small proportion.

6.3.4 Comment

No method based on threshold setting is completely satisfactory. Unless the the choice of threshold is already made, for instance as a result of target setting, results will be based on subjective choice of T or S. A disadvantage from the use of proportions is that information on size of difference between data and the threshold is lost.

Chapter 7

Experiments on model of arrivals to Stockholm Arlanda airport

Question What are the effects of change to landing sequence algorithm and delay-share strategy on the validated computer simulation model of aircraft arrivals into Stockholm Arlanda airport?

Brentnall, Cheng, Drew & Potts (2003) report a preliminary experiment made to investigate this question. This chapter continues the analysis in more detail. Section 7.1 explains the choice of five performance measures used and sets up the basic simulation input parameters. Sections 7.2 to 7.5 report a sequence of experiments made on the simulation model. In each case an experimental design is formulated to help answer a specific question, statistical analysis is made on output and the findings are summarized. Section 7.6 brings together the experimental work to provide an answer to the above question, and draws some general conclusions that may be applicable to aircraft arrivals into any airport.

7.1 Experiment setup

7.1.1 Performance indicators

Change to the system is examined through five performance indicators representing potential benefits and risks. Two benefit performance indicators chosen are delay and landing rate. These relate to the efficiency of Air Traffic Control because if more aircraft land, or aircraft have less delay then the Air Traffic Control system is thought of as more efficient. The precise definitions used are:

Delay Mean positive delay for all aircraft in simulation landed in time period [240,540) minutes.

Table 7.1: Summary of algorithm inputs to model

i	Algorithm	Deadline	CPS	Other
1	FCFS IAF	-	-	-
2	Global makespan	15min	3	If Deadline fails ignore constraint
3	FCFS R/W	-	-	-
4	MPS Heuristic	-	-	MPS=3
5	Approach Stream Makespan	15min	3	If Deadline fails ignore constraint
6	Tardiness IAF	15min	3	If Deadline fails ignore constraint

Landing rate Number of aircraft landing in an hour during the am arrival peak hour 6, i.e. [360,420] minutes.

The risks examined are from the change to control methods that may result from different landing sequences or delay-share strategies. The first risk indicator chosen is time spent in the approach sectors. Its change may have effects on system behaviour and controller workload. Another risk to controllers' behaviour is the stability of the delay-to-lose advice generated by an AMAN: if controllers do not receive stable sequence advice they may be tempted to ignore it. Finally, holding time is used to examine effects of moving delay away from the airport. The precise definitions of these performance indicators are:

Holding Mean time aircraft spend in all holding areas in time period [240,540) minutes.

Time in approach sectors Mean time aircraft spend in Stockholm Arlanda approach "ESOS" sectors in time period [240,540) minutes.

Stability Mean Standard deviation of the delay advised to aircraft by the AMAN in time period [240,540) minutes.

7.1.2 Input factors

There are a number of input factors that may be varied in experimentation. The following experiments concentrate on the major factors: sequencing algorithm, delay-share strategy, wake-vortex mix and arrival rate profile. Input parameters used to validate the simulation model found in Table 5.4 are used as the simulation base set-up. Settings for AMAN sequencing algorithms, delay-share and resequence strategies are shown in Tables 7.1 to 7.3. These are described in Section 4.2.1. Input parameters are set to these values in the following experiments unless otherwise specified.

Table 7.2: Summary of delay-share strategy inputs to model

j	Delay-share Strategy	Route delay proportion
1	All in Hold	-
2	Even	0.2
3	Late	0.2
4	Early	0.2

Table 7.3: Resequence strategies used

Delay-share	Algorithm i					
Strategy j	1	2	3	4	5	6
1	1				1	1
2	2	2	2	3	2	
3	2	2	2	3	2	
4	2	2	2	3	2	

Resequence strategy 1 Upon arrival to IAF
Resequence strategy 2 Every batch size 2 aircraft arrivals to system, or at IAF if not already sequenced
Resequence strategy 3 Every arrival to the system

7.1.3 Output analysis

The analysis techniques described in Section 6.2 are used to analyze the effects of input factors on performance indicators. That is, models are built to test for differences in distribution, and the distribution summaries mean, variance and skewness. Charts are used to display point estimates and confidence intervals for estimated coefficients in a number of cases. In these charts the x-axis Latin letters are used to represent Greek coefficients used in the text, where a = α, b = β, d = δ, f = κ and g = λ. The material presented is a selection from the total experimental analysis. It has been chosen in order to provide detailed statistical evidence for the conclusions drawn.

7.2 Experiment I

7.2.1 Question

The computer simulation has been validated with a set of input parameters against data from arrivals to Stockholm Arlanda in autumn 2003. What would be the effect of changing only algorithm and delay-share strategy?

7.2.2 Design

There are 18 possible algorithm and delay-share strategy combinations. As not all algorithm - delay-share combinations are possible, we have to use an incomplete two-way design. The design is shown in Table 7.3, where combinations that may be run in the simulation have a re-sequence strategy. This design is feasible since the time of a single simulation run is at most around two minutes on the computer used, a 2.4Ghz Pentium 4 with 512MB of DDR RAM. If 50 runs are made at each design point and performance indicator Y_{ijk} recorded, then for $i = 1, \ldots, 6$, $j = 1, \ldots, 4$ and $k = 1, \ldots, 50$ the experimental output linear model is:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk} \tag{7.1}$$

where μ is an overall mean, α_i the effect of algorithm i, β_j the effect of delay-share strategy j, $(\alpha\beta)_{ij}$ their interaction and ϵ_{ijk} a random error. Only 18 of the coefficients are estimable in total, but this includes all the main effects α_i and β_j. The analysis of this experiment is geared towards seeing if α_i and β_j effects are significant, and if so, how. This information feeds back to an idea of how change to sequencing algorithm and delay-share strategy affect the system as-is.

7.2.3 Analysis

Delay

Analysis of distribution Figure 7.1 shows boxplots of mean delay split by algorithm and delay-share strategy. This visualization shows a clear difference in distribution between algorithms operating with a delay-share strategy in hold, and those not in hold.

Analysis of mean Setting a factor $\rho_{ij} \in \{0, 1\}$ of two levels to represent the difference identified above, the reduced model:

$$Y_{ijk} = \mu + \rho_{ij} + \epsilon_{ijk} \tag{7.2}$$

may be fitted using OLS, since variance is approximately constant. Residuals from the fit are not normally distributed so the asymptotic theory for significance of estimates is compromised. However, Jacknife one-at-a-time deletion of observations (Davison & Hinkley 1997) may be used to find these. The result is an estimate of the second level of ρ_{ij} (with the first set to 0) at -1.492 and 95% confidence interval of [-1.493, -1.490]. This difference is backed up by an empirical 95% confidence interval of [-1.537,-1.451], from a bootstrap experiment with 2000 repeats, on the difference in mean between these two groups.

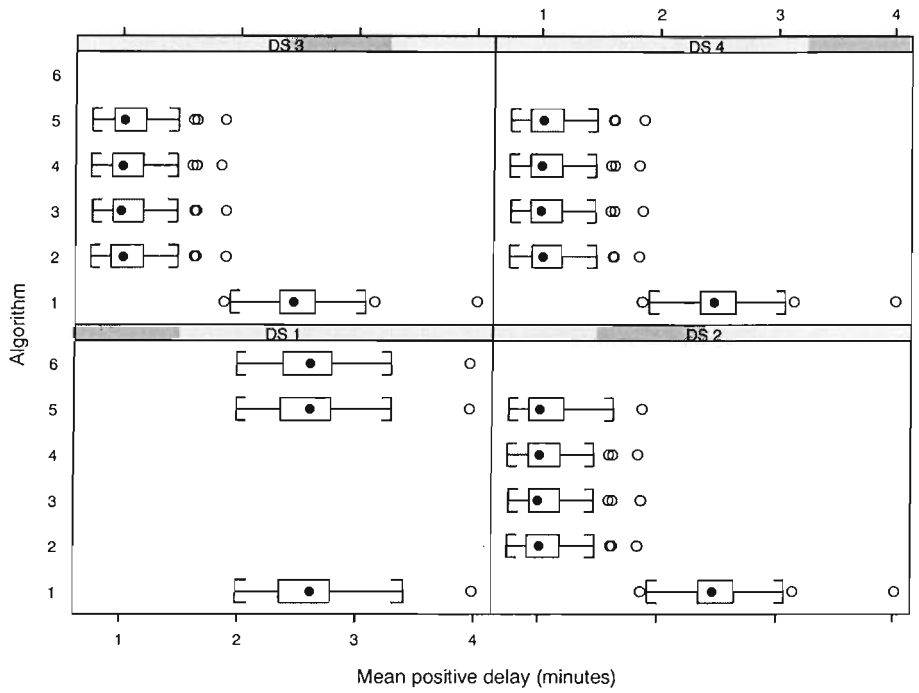


Figure 7.1: Experiment I: Mean positive delay boxplots

Landing rate

Analysis of distribution An EDFIT analysis of the distribution of landing rate at design points did not return any significant coefficients. This confirmed visual inspection of boxplots.

Analysis of mean Analysis of distribution using EDFIT or charts assumes that a traffic sample effect γ_k is bound up in error ϵ_{ijk}. This is a reasonable assumption to make if, regardless of correlation between traffic samples, the final distribution of the response at each design point is representative of the true distribution. However, use of common traffic samples is a simulation trick to reduce variance and find important coefficients. If algorithm and delay-share combination effect is denoted π_i and blocking effect of traffic sample γ_k then investigation of AMAN effects may be carried out using the statistical model:

$$Y_{ik} = \mu + \pi_i + \gamma_k + \epsilon_{ik} \tag{7.3}$$

for $i = 1, \ldots, 18$, $k = 1, \ldots, 50$. Friedmans nonparametric test (Rice 1995) is suitable to test the null hypothesis that $\pi_i = 0$ for $i = 1, \ldots, 18$. The statistic for no difference due to factor π is 45.991, with df = 17 and a p-value = 0.0002.

This suggests that there may be a difference in landing rate between the algorithm and delay-share strategy combinations.

Linear model (7.3) was estimated using OLS since variance appeared constant. The linear model assumes discrete data with range [15,36] may be approximated by a continuous distribution. Of the 18 estimates, only two significantly different groups were found. Group 1 contained algorithm 1 and other algorithms with delay-share strategy 1 (in hold), group 2 the remaining combinations. A simpler model to (7.3) was made, replacing π_i with a factor $\rho_i \in \{0, 1\}$ of two levels based on the split. Residuals were not normally distributed, and a transformation to normality was not found. Significance tests were carried out using Jacknife one-at-a-time deletion of observations. Setting group 1 as a base with value 0 the estimate coefficient for group 2 is -0.220, with a Jacknife 95% confidence interval of [-0.223,-0.216]. However, the multiple R^2 for $Y_{ik} = \mu + \rho_i + \gamma_k + \epsilon_{ik}$ of 0.9928 compares to a multiple R^2 of 0.9923 for $Y_{ik} = \mu + \gamma_k + \epsilon_{ik}$, bringing into question the amount of information contained in the confidence internal.

Holding

Analysis of distribution The boxplots in Figure 7.2 show little difference between time in hold for algorithms at delay-share strategy 1, i.e. in hold. However, the pre-hold strategies (2 - 4) reveal differences. Algorithms 1 and 5 seem to require more holding than 2 - 4. There is also some evidence of a difference between the three pre-hold delay-share strategies. All strategies use holding only as a last resort, but delaying aircraft as late as possible (strategy 3) perhaps requires more holding time than delaying aircraft earlier on. These apparent trends are analyzed below. An EDFIT analysis finds α_i, β_j and $(\alpha\beta)_{ij}$ all significant. However, in this case the significant individual EDFIT coefficient are hard interpret without an estimation procedure.

Analysis of mean Figure 7.2 shows that a fitted model to means would not satisfy a constant variance assumption, and no Box-Cox transformation of the response was found to achieve this. Coefficient WLS estimates and bootstrap (B=1000) confidence intervals are shown in Table 7.4. The main points from the boxplots come through in these estimates: a significant difference is found between delay-share strategy 1 and other strategies, and between algorithms 1 and other algorithms. However, the other effects looked for are not statistically significant. Although the β_j coefficient point estimates are different, their confidence intervals overlap so the possibility that they are all the same cannot be rejected.

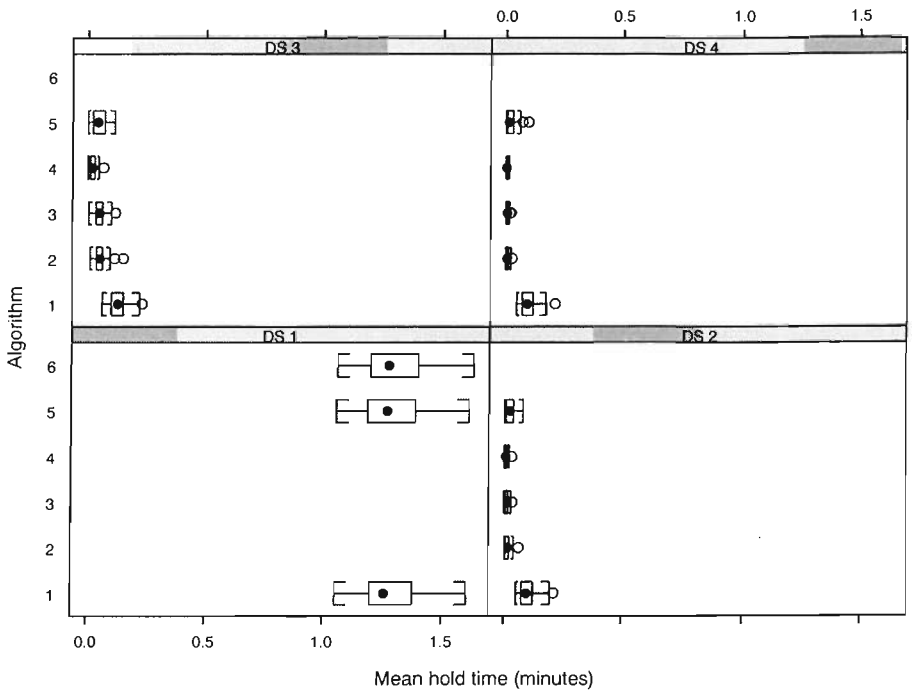


Figure 7.2: Experiment I: Mean holding time boxplots

Table 7.4: Experiment I: Holding time, WLS linear model coefficients

Coefficient	Estimate	95% C.I.	
		Lower	Upper
μ	1.29	1.26	1.33
α_2	-0.09	-0.10	-0.08
α_3	-0.09	-0.10	-0.08
α_4	-0.09	-0.10	-0.09
α_5	0.01	-0.05	0.06
α_6	0.01	-0.04	0.07
β_2	-1.19	-1.23	-1.15
β_3	-1.16	-1.20	-1.12
β_4	-1.20	-1.24	-1.16
$(\alpha\beta)_{22}$	0.00	-0.01	0.02
$(\alpha\beta)_{32}$	0.00	-0.01	0.02
$(\alpha\beta)_{42}$	0.00	-0.02	0.01
$(\alpha\beta)_{52}$	-0.08	-0.14	-0.03
$(\alpha\beta)_{23}$	0.01	0.00	0.03
$(\alpha\beta)_{33}$	0.01	-0.01	0.03
$(\alpha\beta)_{43}$	-0.02	-0.03	0.00
$(\alpha\beta)_{53}$	-0.09	-0.15	-0.04
$(\alpha\beta)_{54}$	-0.08	-0.14	-0.03

Table 7.5: Experiment I: Holding time, variance model coefficients

Coefficient	Point Estimate	95% C.I. Lower	Upper
σ^2	0.0199	-	-
Group 2	-1.36	-1.95	-0.77
Group 3	-1.90	-2.47	-1.33
Group 4	-2.89	-3.44	-2.35

Likewise, the confidence intervals on the $(\alpha\beta)_{5j}$ coefficients suggest there is not enough evidence to say that algorithm 5 produces a higher mean hold time with delay-share strategies 2 - 4 than do algorithms 2 - 4.

Variance analysis Figure 7.2 shows a difference in variance between the design points. Here the difference is quantified with a variance model. The fit of a model to all 18 design points returned 4 groupings. These groups may be seen by viewing Figure 7.2. The groups (descending by size of variance) are:

Group 1 All holding algorithms.

Group 2 Algorithm 1 not in hold and algorithm 5 with delay-share strategy as-late as possible (strategy 3).

Group 3 Algorithms 2-4 with delay-share strategy as late as possible, algorithm 5 with delay-share strategy as early as possible (strategy 4) and even-spread (strategy 2).

Group 4 Algorithms 2-4 with delay-share strategy as early as possible and even-spread.

A bootstrap experiment for multiple comparison confidence intervals between these groups was carried out by resampling 6,000 observations from the design point EDFs and recording difference in sample standard deviation between the 4 groups. Bonferroni 95% multiple confidence intervals (Rice 1995) are shown in Table 7.6. All groups are significantly different at the 0.05 level. A fit of a variance model where the effect of group 1 is set to zero produced similar predictions of variance at each design point to the full model. The maximum likelihood estimates and their approximate 95% confidence intervals are shown in Table 7.5. Correlation between model predictions and design point sample variance is 0.997. From the model and use of the Bonferroni inequality there is 90% confidence on difference in coefficients for groups 1 with all others, and group 2 with 4.

Table 7.6: Experiment I: Holding time, simultaneous 95% confidence intervals on difference between group standard deviation

Difference	Multiple 95% C.I.*	
	Lower	Upper
Group 1 - 2	0.075	0.112
Group 1 - 3	0.098	0.133
Group 1 - 4	0.114	0.149
Group 2 - 3	0.016	0.028
Group 2 - 4	0.033	0.044
Group 3 - 4	0.013	0.020

*6000 Bootstraps

Approach sectors

Analysis of distribution Figure 7.3 shows boxplots of the distributions. It appears that there is the same split as delay between holding algorithms with algorithm 1 and the rest. There is also a shift in location according to delay-share strategy. EDFIT analysis finds similar coefficients significant to those for holding time. However, it is again difficult to interpret their significance without fitting models.

Analysis of mean Fit of the full model using OLS appears to have constant variance of residuals. Differences were picked up between the delay-share strategies. Interactions were significant, seemingly adjusting for the difference between algorithm 5 at delay-share strategy 1 and other strategies. An alternative model was fitted representing algorithm 5 at delay-share strategy 1 as a different algorithm 7, and ignoring the interaction terms. That is:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk} \tag{7.4}$$

where $i = 1, \ldots, 7$, $j = 1, \ldots, 4$, and $k = 1, \ldots, 50$. Plots of residuals were not very different from those of the full model, so this alternative seems just as valid. It is also more interpretable. Residuals were clearly not normally distributed and so the bootstrap was used for significance tests. The estimated coefficients and confidence intervals of the alternative model (7.4) are presented in Table 7.7. Similar results to the model for mean holding time are found. They suggest that moving delay back from the IAF area shifts the mean delay out of approach sectors, even when delaying aircraft as late as possible before the IAFs. The algorithms have different point estimates for their effect on this, but confidence intervals overlap so there is not enough evidence to suggest that they have different effects.

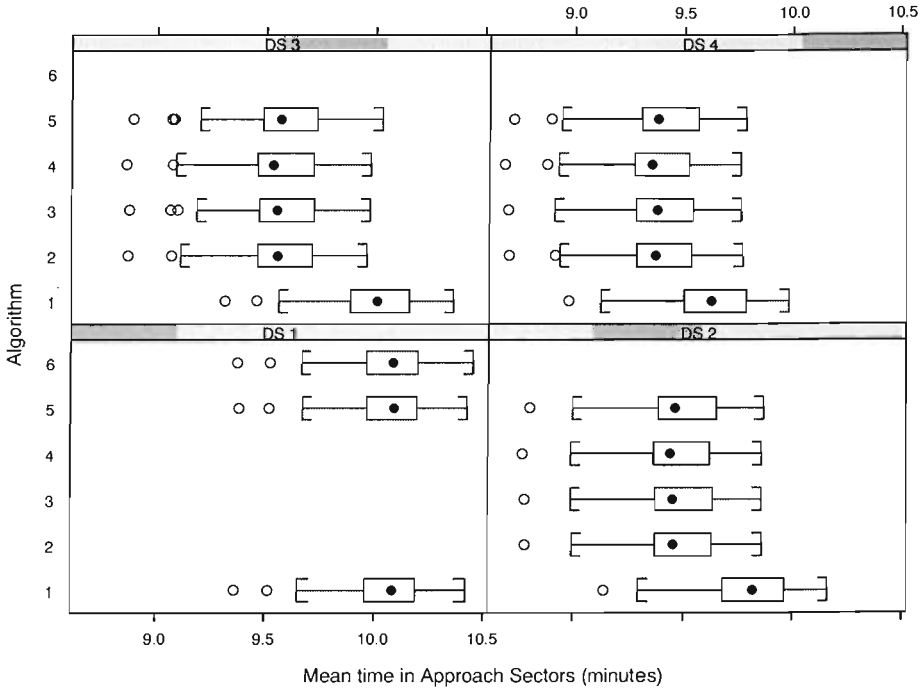


Figure 7.3: Experiment I: Mean time in approach sectors boxplots

Table 7.7: Experiment I: Time in approach sectors, linear model coefficients

Coefficient	Estimate	95% C.I.*	
		Lower	Upper
μ	10.064	10.002	10.124
α_2	-0.337	-0.383	-0.286
α_3	-0.336	-0.385	-0.286
α_4	-0.348	-0.395	-0.294
α_5	-0.324	-0.372	-0.274
α_6	0.007	-0.084	0.097
α_7	0.001	-0.090	0.084
β_2	-0.264	-0.339	-0.191
β_3	-0.147	-0.220	-0.073
β_4	-0.369	-0.444	-0.296

*1000 Bootstraps

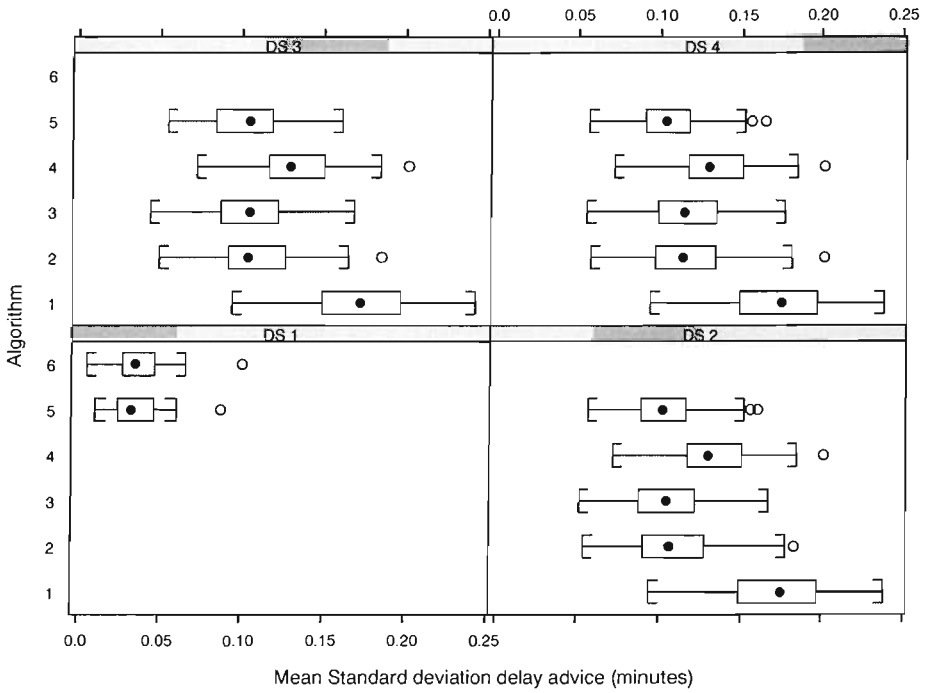


Figure 7.4: Experiment I: Mean standard deviation of delay advised boxplots

Stability of AMAN advice

The analysis of this indicator is slightly different to others. By definition the current set-up does not have a stability of advice concept, since nothing generates advice. This is reflected in the simulation indicator for the base combination: once an aircraft arrives at an IAF point it is placed at the end of the sequence and no change is made to the preceding aircraft sequence. As the indicator for the base combination is zero by definition it does not make sense to include it in the following analysis.

Analysis of distribution Figure 7.4 shows the distribution boxplots. There is a difference between algorithms across delay-share strategies 2-4 and a difference between delay-share strategy 1 and the other strategies. EDFIT analysis finds algorithm and delay-share coefficients significant in mean and variance.

Analysis of mean There is a difference in variance between delay-share strategy 1 and other strategies so WLS was used to fit a model with α_i and β_j terms only. The results are shown in Table 7.8. They show that algorithm 4 has a higher value coefficient than other algorithms (significant pair-wise at 90%). Also, shifting delay

Table 7.8: Experiment I: Stability, WLS linear model coefficients

Coefficient	Estimate	95% C.I.*	
		Lower	Upper
μ	0.1040	0.0966	0.1119
α_2	-0.0626	-0.0691	-0.0563
α_3	-0.0636	-0.0704	-0.0573
α_4	-0.0421	-0.0490	-0.0355
α_5	-0.0686	-0.0747	-0.0619
α_6	-0.0658	-0.0746	-0.0569
β_2	0.0680	0.0615	0.0748
β_3	0.0675	0.0608	0.0736
β_4	0.0719	0.0654	0.0784

*1000 Bootstraps

back from the hold is significant and increases the indicator. There is no significant difference between the delay-share coefficients 2 - 4.

7.2.4 Findings

Processes causing difference in output

Delay The reason for the significant difference in delay between algorithm and delay share strategies is the inefficiency caused by sequencing aircraft based on their order to IAF points. This happens in algorithm 1 and may occur in other holding algorithms. For instance, suppose there are 3 medium type aircraft A, B and C. Aircraft B enters IAF 1 and is assigned as next to land behind aircraft A who is already in the TMA. The time aircraft B lands is not the minimum separation of $A - B$ because B cannot make this time. A few minutes later another aircraft C arrives at a different IAF 2 but could have been placed with minimum separation behind A since IAF 2 is closer to the runway. Instead C is placed (with minimum separation) behind aircraft B. There is more delay overall than would be found from sequence $A - C - B$. This scenario is avoided if aircraft are placed in a sequence other than FCFS at IAF, or if a sequence is made further back than the IAF points. This inefficiency need not be the actual mechanism that occurs in real life. However, since the base combination has been validated against real traffic samples, there may be a real process that causes the difference. A point to note is that the difference is not very large, mean delays are reduced in a 95% confidence range of [1.451, 1.537] minutes.

Landing rate The 95% confidence interval of [0.216,0.223] aircraft an hour increase in landing rate by sequencing at holds seems odd, especially given the higher

delays from this type of sequencing. Looking at the individual landing rates by algorithm and traffic sample shows that there is a difference in landing rate recorded split by pre and at-IAF delay-share strategies. On some occasions pre-IAF lands an extra aircraft in the recorded hour than does at IAF, but the at-IAF strategy lands an extra aircraft more often. The amount of information this result contains is minimal in terms of predictive power and though significant, could be a result of the arbitrary choice of hour.

Risks Stability results for algorithm 3, FCFS at runway, highlight how delay advice will be unstable even when FCFS position in landing sequence is known. Algorithm 1, FCFS at IAF, increases the variability in delay advice over FCFS at runway as a result of the inefficiencies described for delay. Larger delays given to aircraft result in greater variability in delay-to-lose advice. The stability of algorithm 4, the heuristic, is worse than other algorithms for a different reason. It includes a maximum position shift based on aircrafts FCFS position in the landing sequence when it arrives in the AMAN. The shift is set to three, as in the other smart algorithms with CPS constraints. But the algorithm does not bind the maximum shift an aircraft makes from FCFS position to be three. For example, consider a system sequence $M - M$, and a new heavy type aircraft H arrives. Suppose maximum shift is set to 1. Now the H FCFS position is $M - H - M$ but $M - M - H$ minimizes total delay so is chosen by the heuristic. Another medium aircraft $\overline{M}$ arrives with FCFS position $M - M - \overline{M} - H$ which minimizes delay and so is chosen. Aircraft H has now been shifted two positions from its original FCFS position. This process causes algorithm 4 to be significantly more unstable than other algorithms where movement from FCFS is limited by CPS constraints.

Conclusions

The following conclusions are drawn based on analysis of this experiment. The results might be applicable to an airport with similar arrival traffic characteristics to the model of Stockholm Arlanda.

Delay and landing rate Current methods of sequencing are very efficient in comparison with systems that allow minimum separations to be achieved where possible. If the minimum separations were achieved where possible then a 95% confidence interval on the decrease in mean positive delay in the morning peak hour is [1.451, 1.537]. New sequencing methods and delay-share strategies are not found to increase in landing rate.

Time holding Three results were found when delay as moved back at most two sectors from the IAF points. First, the amount of delay lost around the IAF

points was reduced. Second, landing aircraft FCFS at IAF required a larger mean holding time that other algorithms. Third, the variation around the time aircraft had to hold was reduced. A final point, if route restrictions were removed and aircraft did not have to arrive at each IAF point FCFS then the variability of holding could also be reduced by operating with delay-share strategies as early as possible and even-spread through route.

Time in approach sectors Shifting delay back from the IAF points reduced mean time in approach sectors dependent on the strategy. Thus sectors further back from approach would be required to absorb a significant amount of aircraft delay, even if aircraft are delayed as close to IAF points as possible without holding.

Stability of AMAN advice Risk due to stability of advice is increased by definition. The least risk from the mean of the indicator occurred when delaying in holds. The largest risk came from marrying the FCFS at IAF algorithm with pre-IAF delay-share strategies. Algorithm 4 has a statistically significant greater risk than algorithms 2, 3 and 5. This arose because algorithm 4 does not incorporate an absolute CPS strategy, and allows greater change to sequence position.

7.3 Experiment II

7.3.1 Question

What would be the effect of increasing traffic level intensity?

7.3.2 Design

Three criteria are used to choose the design.

Estimation Let k_u be the number of arrival traffic intensity levels. Then the linear model under investigation for $i = 1, \ldots, 6$, $j = 1, 2, 3$, $k = 1, \ldots, k_u$ and $l = 1, \ldots, 50$ is:

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \delta_k + (\alpha\beta)_{ij} + (\alpha\delta)_{ik} + (\beta\delta)_{jk} + \epsilon_{ijkl} \tag{7.5}$$

where δ_k is the effect of traffic level k and $(\alpha\delta)_{ik}$ and $(\beta\delta)_{jk}$ are interaction effects of traffic level with algorithm and delay-share strategy. Other terms are defined as Experiment I. A good design will separate the interesting $(\alpha\delta)_{ik}$ and $(\beta\delta)_{jk}$ interactions as much as possible from other effects.

Reuse Observations from Experiment I are to be reused in this experiment. Observations from this experiment are planned to be used in further experiments analyzing wake-vortex category and arrival route factors.

Run time As traffic level increases there are more aircraft in the model and so the basic airspace simulation will take more time. The dynamic programs will also take longer to run since their complexity depends on the number of aircraft to sequence.

A full factorial design with 18 algorithm/delay-share strategy combinations and three additional traffic levels was chosen. The correlation structure of the output model is as good as it can be because the design is full-factorial. Three further traffic levels were chosen for reuse reasons: new fractional factorial runs are made in Experiments III and IV, on top of this experiment, and choosing to run an additional three traffic levels facilitated balancing new factors in these experiments. The three traffic intensities increase at regular intervals from the validated base level, to an intensity corresponding to approximately the maximum arrival capacity of a single runway if only medium type aircraft arrive. Only the morning peak is run on the model to reduce the total run time. The values chosen are shown in Figure 7.5.

7.3.3 Analysis

As the number of design points increases it becomes more difficult to spot trends through graphical analysis. The following analyses describe any observable chart trends, and look for further patterns by fitting statistical models. The analysis focuses on finding new effects for traffic level and the more important algorithm and delay-share strategy interactions with traffic level.

Delay

Variance is seen to increase with arrival rate in output boxplots. This observation is backed up by a fitted model of variance $\mathrm{Var}\,(Y_{ijk}) = \sigma^2 \exp(\delta_k)$, predicting design point sample variance with correlation coefficient 0.999. The coefficient estimates in Table 7.9 show that as arrival traffic intensity increased, variance in mean positive delay also increased. As a result, WLS was used to estimate a model for mean. Inference based on the bootstrap found δ_k, α_i and $(\alpha\beta)_{5j}$ coefficients significant, but other no other interactions. Correlation between predictions from $\mathrm{E}\,[Y_{ijk}] = \mu + \alpha_i + \beta_j + \delta_k + (\alpha\beta)_{ij}$ and design point sample mean is 0.999. The $(\alpha\delta)_{ik}$ and $(\beta\delta)_{jk}$ effects did not return significant in models for other

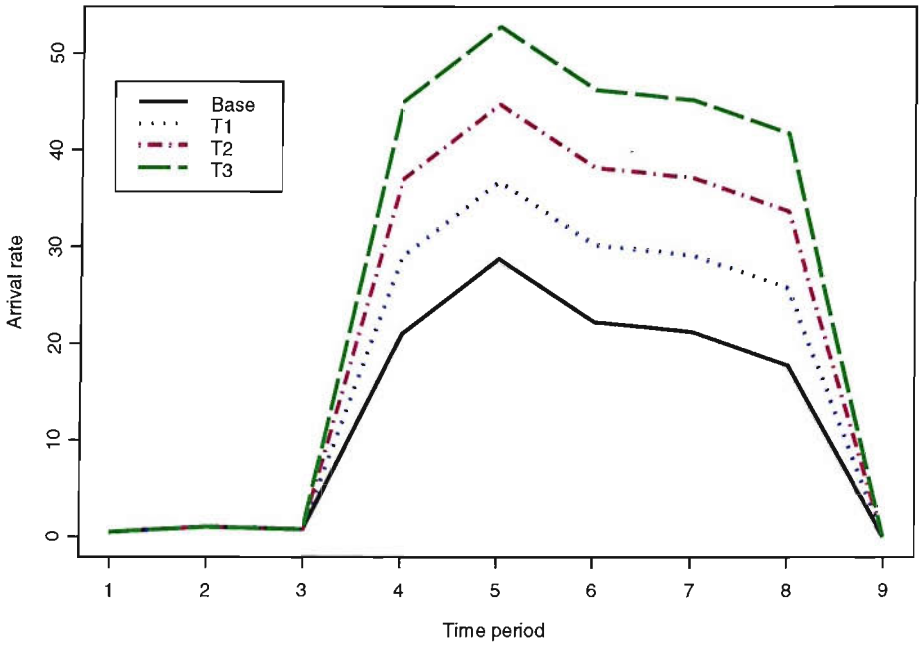


Figure 7.5: Experiment II: Arrival rate levels

distribution summaries. Modelling skewness only returned significant positive δ_k using the bootstrap. This model correlates with design point sample skewness at 0.911. The coefficients and confidence ranges from this model in Table 7.10 show how the skewness of mean positive delay increased as traffic level increased, suggesting that on some runs aircraft experienced greater delays than the majority of others, and this was only due to the quantity of arrival aircraft.

Landing rate

Landing rate boxplots show a clear shift in location as arrival rate increased, and other changes in variance and Shape. An EDFIT analysis suggested that the

Table 7.9: Experiment II: Delay, variance model coefficients

Coefficient	Point Estimate	95% C.I. Lower	Upper
σ^2	0.0751	-	-
δ_2	0.517	0.235	0.799
δ_3	1.342	1.060	1.624
δ_4	2.474	2.192	2.756

Table 7.10: Experiment II: Delay, skewness model coefficients

Coefficient	Point Estimate	95% C.I.* Lower	Upper
μ	1.30	0.87	1.37
δ_2	0.56	0.04	0.81
δ_3	0.48	0.24	0.86
δ_4	0.82	0.56	1.20

*1000 Bootstraps

difference in location, variance and skewness is only due to arrival rate. In building models of variance a fitted model $\mathrm{Var}\,(Y_{ijkl}) = \sigma^2 \exp(\delta_k)$ correlates with design point sample variance at 0.994. Table 7.12 shows the estimated coefficients and approximate confidence ranges. Variance increases up to traffic level 4, where there is not enough evidence to say that variance is not the same as low levels of arrival rate. Since variance differs with arrival rate level, WLS was used to fit linear models of the mean. Only traffic level δ_k returned significant using the bootstrap. Coefficients and confidence ranges for a model $\mathrm{E}\,[Y_{ijk}] = \delta_k$ are shown in Table 7.11. The model correlation coefficient with design point sample mean is 0.994.

The model $\mathrm{E}\,[Y_{ijk}] = \delta_k$ is based on an assumption that traffic sample γ_{kl} is bound up with ϵ_{ijkl}. However, the previous experiment (Section 7.2.3) found a difference between algorithms (ρ_i) when traffic sample effects were incorporated, but the effect did not add much information to the model. The additional runs made in this experiment allowed further tests on whether ρ_i is important. Inference was made using a log-linear model $\ln(\mathrm{E}\,[Y_{ijkl}]) = \rho_i + \gamma_k$, where ρ_i represents the split in sequencing technique as defined in Section 7.2.3. The ρ_i level 2 (when level 1 is set to 0) point estimate returned -0.00233 with SE 0.000665 and dispersion parameter 0.00762. That is, ρ_i is again found significant and landing rate decreases when sequencing pre-IAF. However, the size of effect is very small indeed, and though significant does not contribute much to the predictive power of the model. Correlation between predicted and observed values from the model $\ln(\mathrm{E}\,[Y_{ijkl}]) = \rho_i + \gamma_k$ is 0.9858772. Correlation between predicted and observed values from $\ln(\mathrm{E}\,[Y_{ijkl}]) = \gamma_k$ is 0.9858728. This suggests that ρ_i adds so little information as to be redundant.

Holding

Boxplots appeared to show a shift in location, variance and shape as traffic level increased. In particular the tail of holding time at traffic level 4 is pronounced.

Table 7.11: Experiment II: Landing rate, WLS linear model coefficients

Coefficient	Point Estimate	95% C.I.* Lower	Upper
μ	24.53	24.23	24.82
δ_2	4.62	4.11	5.12
δ_3	13.00	12.48	13.53
δ_4	21.76	21.32	22.23

*1000 Bootstraps

Table 7.12: Experiment II: Landing rate, variance model coefficients

Coefficient	Point Estimate	95% C.I. Lower	Upper
σ^2	22.752	-	-
δ_2	0.249	0.051	0.447
δ_3	0.339	0.140	0.537
δ_4	0.139	-0.059	0.337

This tail is statistically significant: a fit of a linear model to skewness at design points, shown in Table 7.14, found that the distribution of holding time becomes more positively skewed as traffic level increased. Linear model assumptions checked: analysis of residuals showed approximately constant variance and a Shapiro-Wilks test for normality of the residuals returned $W = 0.9876$ and a p-value of 0.7025. One may conclude that there were some occasions when more aircraft than usual needed to hold, regardless of delay-share strategy or algorithm.

Analysis of distribution through EDFIT is summarized in Table 7.13. This shows that all terms had significant effects on the output distributions. The Fourier components of the statistics suggest that the difference due to $(\alpha\delta)_{ik}$ is in variance rather than location. As variance is non-constant WLS was used to fit to $\mathrm{E}[Y_{ijk}] = \mu + \alpha_i + \beta_j + \delta_k + (\beta\delta)_{jk}$. This model predicts design point sample mean with correlation coefficient 0.996. The size and direction of the coefficient estimates are shown in Figure 7.6. In particular, the $(\beta\delta)_{ik}$ coefficient is seen to decrease as the k index increased. In other words, as traffic level intensity increased, delay-share strategy had a greater effect on reducing the mean of the distribution of mean holding time.

Fitting a model to variance found the $(\alpha\delta)_{ik}$ term significant, as suggested by EDFIT. The model $\mathrm{Var}(Y_{ijkl}) = \sigma^2 \exp(\alpha_i + \beta_j + \delta_k + (\alpha\delta)_{ik} + (\beta\delta)_{jk})$, with $\sigma^2 = 0.0302$, correlates to design point sample variance with coefficient 0.999.

Table 7.13: Experiment II: Holding time, full EDFIT table

Coefficient	T^2	C_1	C_2	C_3	C_R
α_2	**659.93 (.000)**	**252.59 (.000)**	**321.94 (.000)**	**73.4 (.000)**	**11.99 (.000)**
α_3	**764.92 (.000)**	**285.14 (.000)**	**374 (.000)**	**95.1 (.000)**	**10.67 (.000)**
α_4	**924.91 (.000)**	**290.09 (.000)**	**451.11 (.000)**	**167.11 (.000)**	**16.6 (.000)**
α_5	**83.16 (.000)**	15.4 (.116)	**41.54 (.000)**	**18.92 (.000)**	**7.3 (.000)**
α_6	21.1 (.337)	3.41 (.611)	11.55 (.044)	4.08 (.076)	2.06 (.888)
β_2	**1294.21 (.000)**	**1117.53 (.000)**	**32.73 (.000)**	**66.28 (.000)**	**77.67 (.000)**
β_3	**1227.7 (.000)**	**966.9 (.000)**	**76.87 (.000)**	**125.91 (.000)**	**58.02 (.000)**
β_4	**1239.9 (.000)**	**1062.44 (.000)**	**33.56 (.000)**	**50.92 (.000)**	**92.98 (.000)**
δ_2	**151.79 (.000)**	6.27 (.396)	**60.42 (.000)**	**6.52 (.009)**	**78.58 (.000)**
δ_3	**434.16 (.000)**	8.35 (.322)	**209.95 (.000)**	**49.26 (.000)**	**166.6 (.000)**
δ_4	**624.95 (.000)**	4.06 (.503)	**269.74 (.000)**	**108.68 (.000)**	**242.46 (.000)**
$(\alpha\beta)_{22}$	22.05 (.124)	9.17 (.246)	7.75 (.034)	1.7 (.158)	3.44 (0.08)
$(\alpha\beta)_{32}$	29.33 (.058)	12.45 (.184)	**11.04 (.012)**	2.38 (.089)	3.47 (0.08)
$(\alpha\beta)_{42}$	13.35 (.284)	4.08 (0.45)	3.83 (.135)	1.11 (.229)	4.33 (.028)
$(\alpha\beta)_{52}$	**81.36 (.000)**	**42.37 (.008)**	**33.66 (.000)**	0.83 (.324)	4.51 (.021)
$(\alpha\beta)_{23}$	**121.99 (.000)**	**51.04 (.006)**	**54.47 (.000)**	**11.57 (0)**	**4.91 (.011)**
$(\alpha\beta)_{33}$	**166.86 (.000)**	**67.18 (.003)**	**76.23 (.000)**	**18.08 (.000)**	**5.38 (.005)**
$(\alpha\beta)_{43}$	**78.45 (.001)**	13.35 (.175)	**31.58 (.000)**	**18.35 (.000)**	**15.18 (.000)**
$(\alpha\beta)_{53}$	**65.73 (.004)**	**44.09 (.011)**	**14.02 (.006)**	1.52 (.192)	**6.1 (.004)**
$(\alpha\beta)_{54}$	**109.13 (.000)**	**56.63 (.002)**	**47.54 (.000)**	3.07 (.052)	1.9 (.492)
$(\alpha\delta)_{22}$	**121.79 (0)**	0.16 (.901)	**28.79 (0)**	**67.15 (.000)**	**25.7 (.000)**
$(\alpha\delta)_{32}$	**120.65 (.000)**	1.89 (.635)	**21.07 (.004)**	**71.15 (.000)**	**26.53 (.000)**
$(\alpha\delta)_{42}$	**113.48 (0)**	10.17 (.308)	8.49 (.046)	**66.47 (.000)**	**28.35 (.000)**
$(\alpha\delta)_{52}$	**69.28 (.003)**	2.28 (.596)	**25.13 (.000)**	**30.16 (.000)**	**11.71 (.000)**
$(\alpha\delta)_{62}$	18.25 (.722)	0.58 (.883)	6.06 (.328)	8.19 (.077)	3.41 (.938)
$(\alpha\delta)_{23}$	**345.19 (.000)**	0.26 (.872)	**159.16 (.000)**	**153.78 (.000)**	**31.98 (.000)**
$(\alpha\delta)_{33}$	**350.87 (.000)**	3.03 (.562)	**148.44 (.000)**	**176.26 (.000)**	**23.14 (.000)**
$(\alpha\delta)_{43}$	**517.43 (.000)**	2.28 (0.63)	**196.77 (.000)**	**287.08 (.000)**	**31.3 (.000)**
$(\alpha\delta)_{53}$	**198.41 (.000)**	27.58 (0.06)	**106.26 (.000)**	**45.63 (.000)**	**18.94 (.000)**
$(\alpha\delta)_{63}$	50.79 (.246)	6.7 (.605)	26.85 (0.03)	11.76 (.044)	5.47 (.601)
$(\alpha\delta)_{24}$	**420.23 (.000)**	33.21 (.064)	**322.51 (.000)**	**49.08 (.000)**	**15.44 (.000)**
$(\alpha\delta)_{34}$	**464.56 (.000)**	32.99 (.066)	**358.3 (.000)**	**62.84 (.000)**	**10.43 (.000)**
$(\alpha\delta)_{44}$	**688.68 (.000)**	**66.32 (.009)**	**485.31 (.000)**	**132.71 (.000)**	4.33 (.075)
$(\alpha\delta)_{54}$	**231.23 (.000)**	**79.45 (.001)**	**110.1 (.000)**	**26.25 (.000)**	**15.43 (.000)**
$(\alpha\delta)_{64}$	58.69 (0.19)	19.15 (.375)	28.68 (.026)	6.27 (.123)	4.6 (.752)
$(\beta\delta)_{22}$	**248.1 (.000)**	28.44 (.111)	**123.7 (.000)**	**34.92 (.000)**	**61.04 (.000)**
$(\beta\delta)_{32}$	**304.38 (.000)**	48.6 (.037)	**107.35 (.000)**	**92.68 (.000)**	**55.75 (.000)**
$(\beta\delta)_{42}$	**208.5 (.000)**	11.7 (.314)	**92.16 (.000)**	**33.92 (.000)**	**70.72 (.000)**
$(\beta\delta)_{23}$	**956.64 (.000)**	**291.98 (.000)**	**303.2 (.000)**	**242.32 (.000)**	**119.14 (.000)**
$(\beta\delta)_{33}$	**1192.19 (.000)**	**386.55 (.000)**	**164.85 (.000)**	**570.23 (.000)**	**70.56 (.000)**
$(\beta\delta)_{43}$	**767.29 (.000)**	**164.93 (.000)**	**258.03 (.000)**	**183.87 (.000)**	**160.45 (.000)**
$(\beta\delta)_{24}$	**1547.32 (.000)**	**872.26 (.000)**	**86.79 (.000)**	**341.07 (.000)**	**247.2 (.000)**
$(\beta\delta)_{34}$	**1574.24 (.000)**	**795.63 (.000)**	0.47 (.686)	**557.46 (.000)**	**220.68 (.000)**
$(\beta\delta)_{44}$	**1377.6 (.000)**	**709.95 (.000)**	**154.75 (.000)**	**240.18 (.000)**	**272.72 (.000)**

Table 7.14: Experiment II: Holding time, skewness model coefficients

	Estimate	SE	t-value	p-val
μ	0.911	0.101	9.052	0.000
δ_2	1.936	0.142	13.595	0.000
δ_3	1.400	0.142	9.830	0.000
δ_4	1.877	0.142	13.185	0.000

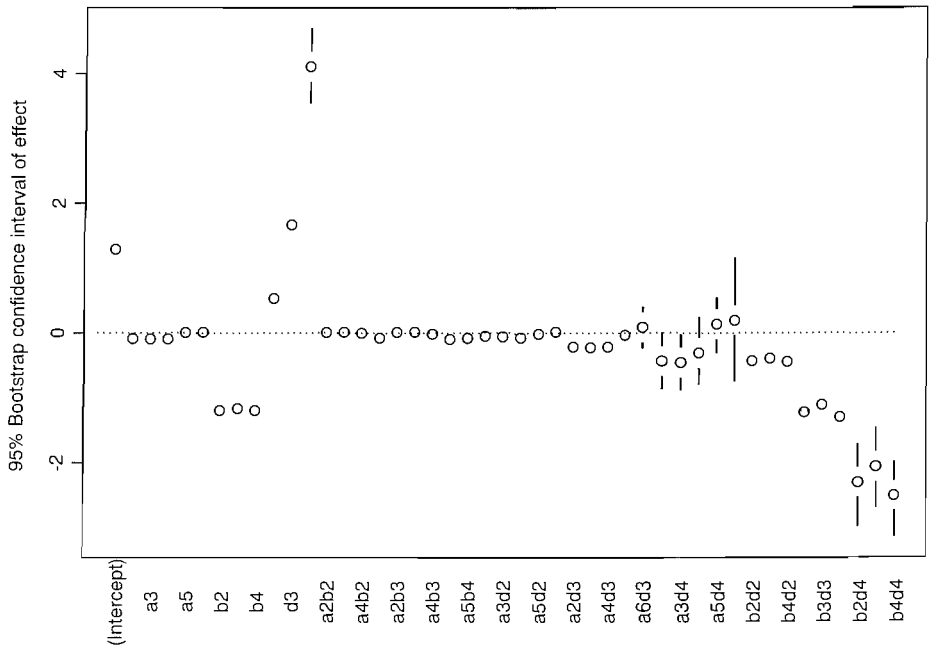


Figure 7.6: Experiment II: Holding time, WLS model coefficients

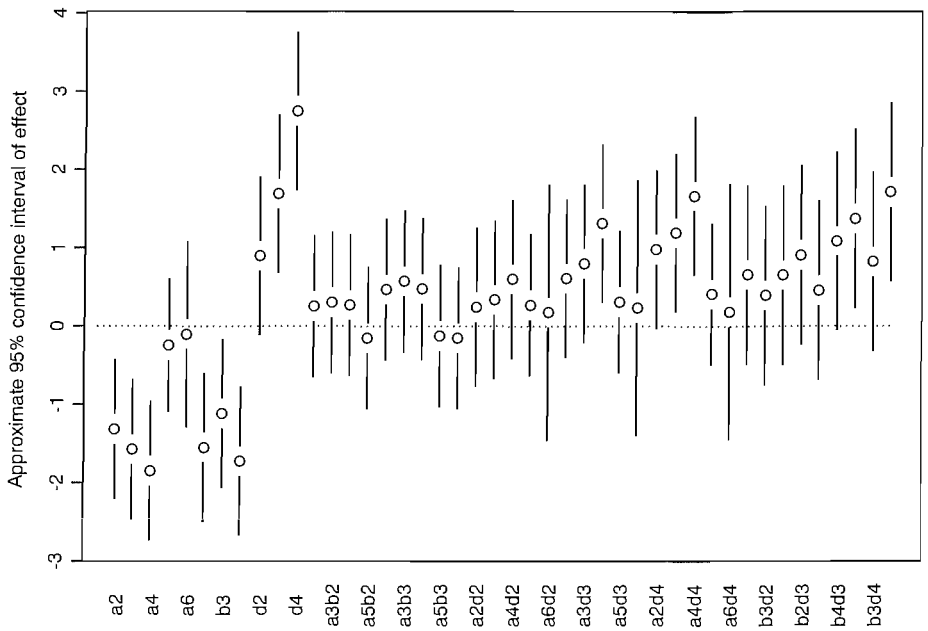


Figure 7.7: Experiment II: Holding time, variance model coefficients

A plot of the maximum likelihood estimates of the coefficients and approximate 95% confidence intervals is shown in Figure 7.7. Their size and direction show relative effects on variance. The δ_k coefficients are significant, positive and increase with k, so variance in holding time increased with traffic level k. Coefficients $(\alpha\delta)_{43}$, $(\alpha\delta)_{44}$, $(\alpha\delta)_{24}$ and $(\alpha\delta)_{34}$ are significant and positive, so as traffic level increased, the reduction in variance attributable to these algorithms decreased. Algorithm 4 is more affected because it is significant at traffic level 3. Similar interpretation may be made for the significant and positive $(\beta\delta)_{24}$ and $(\beta\delta)_{44}$ effects. Notice how the signs of the $(\beta\delta)_{jk}$ coefficients are opposite to the model of mean. That is, as traffic level increased delay-share strategy had more effect on reducing mean, but less effect on reducing variance.

Approach sectors

Visual inspection of output boxplots showed a shift in location as arrival intensity increased. The model $\mathrm{E}[Y_{ijk}] = \alpha_i + \beta_j + \delta_k + (\alpha\beta)_{ij} + (\beta\delta)_{jk}$ was fitted by WLS and predicts design point sample mean with correlation coefficient 0.999. Plots of the coefficient estimates and their 95% confidence intervals are shown in Figure 7.8. The size of the effects are not very large in comparison to the overall

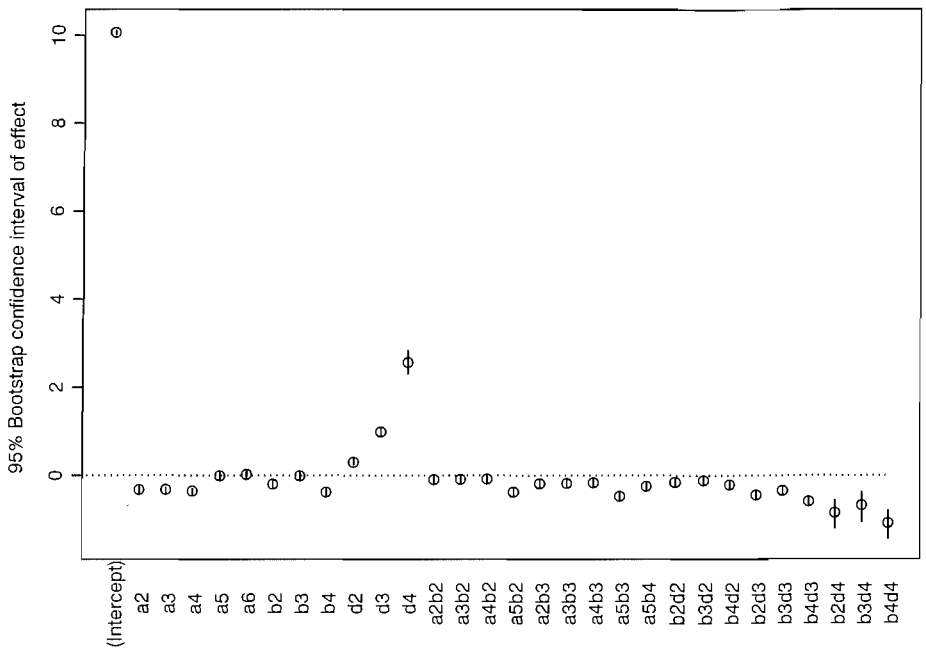


Figure 7.8: Experiment II: Time in approach sectors, WLS estimated coefficients

mean μ. However, the $(\beta\delta)_{jk}$ estimates show that the pre-hold delay-share strategies have a larger effect on reducing the time in approach sectors as traffic level increases. Or, as traffic level increases more delay is shifted back to pre-approach sectors.

The correlation coefficient between predictions from $\mathrm{Var}\,(Y_{ijkl}) = \sigma^2 \exp(\delta_k)$ and design point sample variance is 0.993. Traffic level δ_k was the only term to return significant when trying alternative models. Fitted values in Table 7.15 show that the variance of time in approach sectors increased with arrival traffic intensity. This occured because the variability in time aircraft spend around IAF points (inside the approach sectors) also increases with traffic intensity.

Stability of AMAN advice

Boxplots showed an increase in location, variance and skewness of this indicator as traffic intensity increased. The fit of a model
$\mathrm{E}\,[Y_{ijk}] = \alpha_i + \beta_j + \delta + (\alpha\delta)_{ik} + (\beta\delta)_{jk}$ by WLS predicts sample mean with correlation coefficient 1. Bootstrap inference found the same terms significant as EDFIT analysis. The estimated coefficients and their confidence intervals are

Table 7.15: Experiment II: Time in approach sectors, variance model coefficients

Coefficient	Point Estimate	95% C.I. Lower	Upper
σ^2	0.0465	-	-
δ_2	0.605	0.355	0.855
δ_3	0.796	0.547	1.046
δ_4	1.972	1.722	2.222

Table 7.16: Experiment II: Stability, variance model coefficients

Coefficient	Point Estimate	95% C.I. Lower	Upper
σ^2	0.000726	-	-
δ_2	1.175	0.902	1.448
δ_3	2.097	1.824	2.370
δ_4	3.005	2.732	3.278

shown in Figure 7.9. In Experiment I algorithm 4 was found to be different to other algorithms. In this experiment the interaction $(\alpha\delta)_{44}$ is significantly different to $(\alpha\delta)_{24}$ and $(\alpha\delta)_{34}$ pairwise at 90%, suggesting that as traffic level increases the variation in advice given by algorithm 4 also increases more than algorithms 2 and 3. This is more evidence of the consequence of allowing greater choice in landing sequence. Modelling variance only found traffic intensity significant. The estimated coefficients in Table 7.16 show how variance of the variability of AMAN advice increased as traffic intensity increased. The model correlates to sample variance with coefficient 0.932.

7.3.4 Findings

The following conclusions are drawn based on analysis of this experiment. The results might be applicable to an airport with similar arrival traffic characteristics to that of Stockholm Arlanda.

Delay and landing rate Sequencing algorithms and delay-share strategies did not affect delay and landing rate differently as levels of arrival intensity increased. A change in distribution of arrival rate or delay was attributable to traffic intensity. A result from Experiment I indicated that sequencing at IAF points produced significantly higher landing rates than sequencing pre-IAF. Further analysis with output from this experiment rejected this finding because little difference was found in the predictive power of a statistical model with or without the algorithm split.

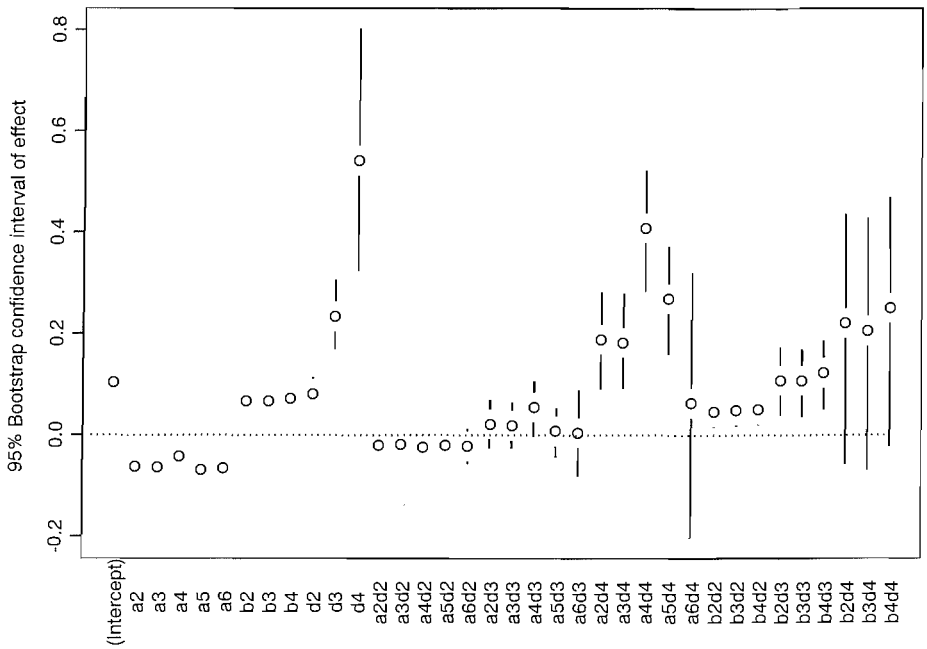


Figure 7.9: Experiment II: Stability, WLS model coefficients

Holding time Holding time was reduced with greater effect by pre-hold delay-share strategies as traffic level increased. Experiment I found that new algorithms and pre-hold delay-share strategies reduced the variability of holding time. This experiment showed that as traffic level increased, the size of these reductions decreased.

Time in approach sectors If pre-hold delay-share strategies were used, aircraft were shown to spend proportionally more time in pre-approach sectors as traffic intensity increased. This occurred because the system had more delay as arrival intensity increased, so aircraft were more likely to delay further out with a greater delay. As a consequence, when demand is high, controllers in pre-approach Sectors would be asked to do relatively more work than at low arrival demand periods. Variability in holding time was also found to change. However, it was not affected by AMAN choice, rather it increased with traffic intensity as a result of the increase in holding time variability.

Stability When there are more aircraft to sequence, and an Arrival Manager might be most useful, the stability of advice indicator is highest. Individual algorithms had effects on stability and their behaviour was found to change according to arrival intensities. This was especially true of the heuristic

algorithm 4, strengthening the conclusion drawn in Experiment I. That is, CPS constraints are necessary to ensure stability of advice when changing the sequence from FCFS. A more general algorithm trend was also picked up: as traffic level increased, the algorithms delaying before the IAF points became more stable relative to those holding, but remained more unstable overall.

The results from this experiment were not dramatic. Algorithm and delay-share strategy choice was shown to affect the Air Traffic Control system in different, important ways as traffic intensity increased. But no further benefits to delay or landing rate were found over and above those in Experiment I.

7.4 Experiment III

7.4.1 Question

What would be the effect of changing the traffic intensity and the type of aircraft arrivals?

7.4.2 Design

Type of arrivals The IAF points at which aircraft exit the simulation model are quite informative on the arrival aircraft type. The IAF points are roughly situated north, south, east and west of the Arlanda runways, and aircraft departing from similar locations enter the TMA through the same IAF point. Country of departure percentages, split by wake-vortex category and IAF point, are shown in Table 7.17. The table is based on Flight Plan data of arrivals to Arlanda between 14th June and 11th July 2001. One way to define type of arrival using the information in this table is classify arrivals as Local or International traffic. An arrival route/wake-vortex category combination is defined as "Local" when the majority of its traffic is from the Nordic countries Sweden, Norway, Denmark, or Finland. A combination is classified "International" if the majority departs elsewhere. The results of this classification are shown in Table 7.18. Three criteria are used to choose an experimental design to investigate change in these arrival types.

Estimation The purpose of this experiment is to ask whether algorithm i and delay-share strategy j behave differently according to aircraft arrival type l. The full linear model under investigation is:

$$\begin{aligned} Y_{ijkl} &= \mu + \alpha_i + \beta_j + \delta_k + \eta_l \\ &\quad +(\alpha\beta)_{ij} + (\alpha\delta)_{ik} + (\beta\delta)_{jk} + (\alpha\eta)_{il} + (\beta\eta)_{jl} + \epsilon_{ijklm} \end{aligned} \tag{7.6}$$

Table 7.17: Experiment III: Country of departure split by aircraft type and IAF

IAF	Country	Wake Vortex Category	Percentage (%)
1	Belgium	M	8.66
1	England	M	21.63
1	France	M	10.36
1	Ireland	M	4.43
1	Netherlands	M	9.92
1	Norway	M	18.82
1	Spain	M	4.55
1	Sweden	M	17.00
1	Other	M	4.63
1	Norway	L	54.55
1	Sweden	L	42.42
1	Other	L	3.03
1	England	H	14.47
1	USA	H	71.70
1	Other	H	13.84
2	Finland	M	15.16
2	Sweden	M	84.77
2	Other	M	0.06
2	Sweden	L	100.00
3	Denmark	M	14.42
3	Germany	M	12.73
3	Greece	M	3.09
3	Italy	M	4.27
3	Sweden	M	41.41
3	Switzerland	M	7.85
3	Other	M	16.23
3	Sweden	L	92.31
3	Other	L	7.69
3	Belgium	H	5.78
3	Cyprus	H	6.67
3	Denmark	H	17.33
3	Germany	H	15.11
3	Greece	H	22.67
3	Spain	H	12.44
3	Turkey	H	10.67
3	Other	H	9.33
4	Estonia	M	10.77
4	Finland	M	78.38
4	Russia	M	10.41
4	Other	M	0.44
4	Finland	H	30.51
4	Thailand	H	27.12
4	United Arab Emirates	H	22.03
4	Other	H	20.34

Table 7.18: Experiment III: Classification of traffic as local or international

IAF	Wake-Vortex category		
	Heavy	Medium	Light
1	International	International	Local
2	-	Local	Local
3	International	Local	Local
4	International	International	-

for $i = 1, \ldots, 6$, $j = 1, \ldots, 4$, $k = 1, \ldots, 4$, $l = 1, \ldots, l_u$ and $m = 1, \ldots, 50$, where l_u is the number of arrival type levels chosen. The important effects to estimate are $(\alpha\eta)_{il}$ and $(\beta\eta)_{jl}$ interactions. A good design will separate these as much as possible from the others.

Reuse The analysis will reuse observations taken in Experiment II.

Run time A similar number of runs as made in Experiment II would be acceptable.

Choice of design One way to proceed would be to repeat the full factorial design of Experiment II at different levels of arrival type. However, this is computationally prohibitive. If total number of runs is to be the same as Experiment II this only permits one further level of arrival type. To use the same number of runs as Experiment II and have more than one extra arrival type level therefore requires further runs to be fractional factorial. The unbalanced situation means standard fractional factorial designs are not applicable.

The runs could be split up to ensure each combination (i, j) has each traffic level k and new arrival route level l run exactly once. If all three traffic levels in Experiment II are used with three further levels of arrival route the total number of additional runs is the number made in Experiment II, and every $(\alpha\eta)_{il}$ interaction is estimable. This part of the design is shown in the right three columns of Table 7.19. To make this design as good as possible careful choice of how traffic level and arrival route pairs (k, l) are assigned to algorithm pairs (i, j) is needed. Several arrangements ensure each combination (i, j) has each traffic level k and new arrival route level l run exactly once, but a good design also balances traffic level and arrival type counts as much as possible by delay-share strategy. The algorithm and delay-share strategy columns in Table 7.19 show the choice of algorithm and delay-share strategies to run over the basic (k, l) design. The six (i, j) pairs pairs run at the same (k, l) levels are identified within the same block A_i or B_i.

Table 7.19: Experiment III: Fractional factorial part of experimental design

Pairing block	Algorithm i	Delay-share Strategy j	Arrival type l for traffic level k		
			$k=2$	$k=3$	$k=4$
A_1	3	2	2	3	4
A_2	3	3	3	4	2
A_3	3	4	4	2	3
B_1	2	3	2	4	3
B_2	2	4	3	2	4
B_3	2	2	4	3	2
A_1	4	4	2	3	4
A_2	4	2	3	4	2
A_3	4	3	4	2	3
B_1	1	2	2	4	3
B_2	1	3	3	2	4
B_3	1	4	4	3	2
A_1	5	3	2	3	4
A_2	5	4	3	4	2
A_3	5	2	4	2	3
B_1	5	1	2	4	3
B_2	1	1	3	2	4
B_3	6	1	4	3	2

Table 7.20 visualizes this choice, and show counts of combinations run across algorithm and delay-share strategies.

In this design all the (i,j,l) and (i,j,k) combinations are run, all interaction effects are estimable and the balance in each (j,k,l) combination is as good as possible. However, to achieve this, some (i,j) combinations are run at the same design points. Unfortunately the effect of these pairs cannot be estimated as they are confounded with other effects in the output model. The correlation matrix from an OLS fit to data from this design is helpful, though large. To quantify how separable important estimates are (mean, standard deviation) pairs of the absolute value of correlation coefficients are presented. Separation between $(\alpha\eta)_{..}$ interaction and main effect $\alpha_{.}$ is (0.007,0.008), and between $(\alpha\eta)_{..}$ and $\eta_{.}$ is (0.119,0.135). Separation between $(\beta\eta)_{..}$ and $\beta_{.}$ is (0.006,0.004), and between $(\beta\eta)_{..}$ with $\eta_{.}$ is (0.278,0.276).

Choice of arrival type levels The three further levels of arrival type chosen were to increase international traffic, relative to local traffic, by -10%, 10% and 20%. The actual numbers used are found in the Appendix.

Table 7.20: Experiment III: Visualization of fractional factorial part of experimental design

Traffic Level	Algorithm	Combinations run marked X Those not possible are shaded out												Count
2	6								X					1
	5	X		X					X		X			4
	4				X		X					X		3
	3			X					X		X			3
	2		X					X					X	3
	1		X			X		X					X	4
3	6					X								1
	5		X					X		X			X	4
	4			X					X		X			3
	3				X		X					X		3
	2				X		X					X		3
	1	X		X					X		X			4
4	6	X												1
	5				X	X	X					X		4
	4		X					X					X	3
	3		X					X					X	3
	2			X					X		X			3
	1				X		X			X		X		4
Delay-share strategy		1	2	3	4	1	2	3	4	1	2	3	4	
Arrival type		2				3				4				
Count		3	5	5	5	3	5	5	5	3	5	5	5	54

Table 7.21: Experiment III: Delay, reduced EDFIT Table

Coefficient	T^2	C_1	C_2	C_3	C_R
α	**124.45 (.000)**	**117.05 (.000)**	**5.75 (.000)**	**1.16 (.000)**	**0.5 (.000)**
β	**14.11 (.000)**	**13.19 (.000)**	**0.68 (.000)**	0.12 (.071)	0.13 (.714)
δ	**343.81 (.000)**	**316.45 (.000)**	**17.98 (.000)**	**3.64 (.000)**	**5.73 (.000)**
η	**1.89 (.001)**	**1.15 (.007)**	**0.35 (.004)**	**0.11 (.019)**	**0.28 (.000)**
$(\alpha\beta)$	**13.97 (.000)**	**13.12 (.000)**	**0.54 (.013)**	0.1 (.449)	0.21 (.904)
$(\alpha\delta)$	**64.46 (.000)**	**17.83 (.000)**	**36.92 (.000)**	**8 (.000)**	**1.71 (.000)**
$(\alpha\eta)$	2.42 (.398)	1.07 (.736)	0.56 (.068)	**0.34 (.006)**	0.46 (.113)
$(\beta\delta)$	**7.21 (.000)**	**2.01 (.014)**	**4.3 (.000)**	**0.57 (.000)**	0.33 (.037)
$(\beta\eta)$	1.53 (.414)	0.33 (.949)	**0.86 (.000)**	0.1 (.478)	0.24 (.691)

7.4.3 Analysis

The analysis in this section is focused on arrival type η_l effects and their interactions with algorithm $(\alpha\eta)_{il}$ and delay-share strategy $(\beta\eta)_{jl}$. Unless otherwise stated similar results as Experiment II are found for other main effects and interactions.

Delay

It is not clear from examination of boxplots whether the interaction effects $(\alpha\eta)_{il}$ and $(\beta\eta)_{jl}$ are significant. A reduced EDFIT table shown in Table 7.21 suggests difference in distribution is not attributable to these interactions. Variance in distribution by design point is not constant so WLS was used to fit models for the mean. The model fitting process lead to reduction of model (7.6) to only include main effects and the $(\alpha\beta)_{ij}$ interaction. An effect for arrival type is picked up, but it is very small: η_2 is not significant, η_3 has point estimate 0.07 and 95% bootstrap empirical confidence interval [0.012,0.144] and η_4 point estimate 0.16 with a empirical bootstrap 95% interval [0.090,0.240]. In other words, increasing international traffic is found to increase mean delay by around 10 seconds. Correlation between sample means at each design point and predictions from the linear model is 0.997 with η terms, and 0.995 without.

Landing rate

EDFIT analysis suggested that landing rate depends on arrival type of traffic and arrival intensity. A fitted WLS model $\mathrm{E}\left[Y_{ijkl}\right] = \delta_k + \eta_l$ returned with the coefficient bootstrap confidence intervals shown in Table 7.22, where η_2 and η_4 return significant. Predictions from this model follow design point sample mean with correlation coefficient 0.999. EDFIT analysis was further borne out in fitting a

Table 7.22: Experiment III: Landing rate, linear model coefficients

Coefficient	Point Estimate	95% C.I.* Lower	Upper
μ	24.532	24.248	24.862
δ_2	4.772	4.294	5.208
δ_3	12.754	12.277	13.224
δ_4	21.823	21.417	22.255
η_2	0.477	0.049	0.850
η_3	0.279	-0.130	0.715
η_4	0.605	0.167	0.994

*1000 Bootstraps

Table 7.23: Experiment III: Landing rate, variance model coefficients

Coefficient	Point Estimate	95% C.I. Lower	Upper
σ^2	22.752	-	-
δ_2	0.262	0.088	0.435
δ_3	0.312	0.139	0.485
δ_4	0.154	-0.019	0.327
η_2	-0.209	-0.364	-0.054
η_3	-0.133	-0.288	0.022
η_4	-0.102	-0.257	0.052

variance model. Table 7.23 shows fitted coefficient values for $\mathrm{Var}\,(Y_{ijkl}) = \sigma^2 \exp(\delta_k + \eta_l)$. Arrival type effect η_2 is negative and significant at the 0.05 level suggesting that as more local aircraft make up arrivals the variability in landing rate decreases. This model has correlation coefficient of 0.8678 between predicted standard deviation and observed sample standard deviation at each design point.

Holding

The models reported in Experiment III explain the output.

Approach sectors

Analysis from EDFIT suggested that arrival type of traffic does not affect the time aircraft spend in Stockholm approach sectors. This was borne out with further model fitting.

Stability of AMAN advice

Analysis of distribution through EDFIT showed some evidence that arrival type of traffic will affect the distribution of stability of AMAN advice. The simplest fit for a model of mean was to add η to the Experiment II model, resulting in output model $\mathrm{E}[Y_{ijkl}] = \alpha_i + \beta_j + \delta_k + \eta_l + (\alpha\beta)_{ij} + (\alpha\delta)_{ik} + (\beta\delta)_{jk}$. Only η_4 is significant with a bootstrap 95% confidence intervals on its point estimate 0.0281 of [0.0144, 0.0424]. This model explains design point sample means extremely well, the correlation coefficient is 0.999.

7.4.4 Findings

Delay and landing rate Algorithms and delay-share strategies did not affect these indicators differently as type of arrival traffic changed. Increasing international traffic increased mean delay and landing rate. Increase in local traffic increased mean of landing rate and decreased its variability.

Holding time and approach sectors Results were identical to Experiment II.

Stability of AMAN advice Change in arrival types increased mean variability of advice given by the AMAN, but the effect was dwarfed by others found in Experiment II.

The outcome from this experiment is largely negative. Changing traffic by the local or international categories involved changing arrival route directions much more than wake-vortex traffic mix. A general conclusion to draw is that the sequencing techniques are fairly stable to change in arrival route, when overall wake-vortex mix does not stray much from current levels.

7.5 Experiment IV

7.5.1 Question

What would be the effect of varying the wake-vortex traffic mix and traffic intensity?

7.5.2 Design

Experiment III looked at changes to arrival route and wake-vortex mix, so any effect solely due to change in wake-vortex mix of traffic was bound up in estimates for arrival type. Given that the aircraft landing sequence problem depends on

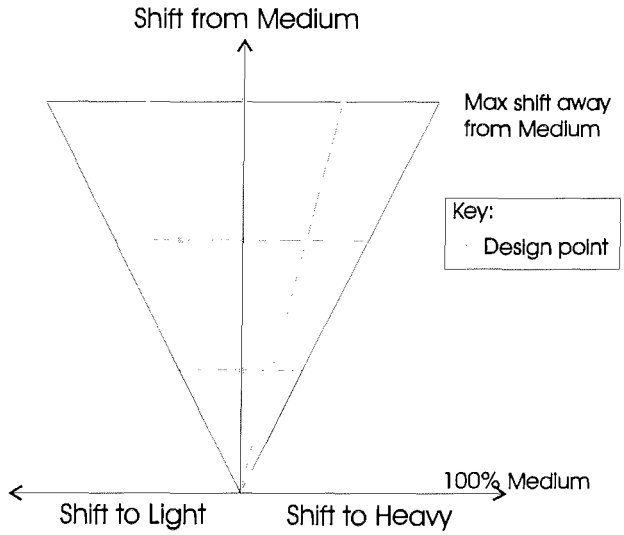


Figure 7.10: Experiment IV: Representation of wake-vortex space and design points

wake-vortex traffic mix, it is interesting to see what might happen if this is changed. Also, the differences found between sequencing algorithms so far have been quite subtle. This may be due to the relatively large proportion of medium type aircraft in the traffic mix. This experiment is run to examine any trends that occur as wake-vortex mix moves away from current-day high proportion of medium type aircraft arrivals in the simulation model of Stockholm Arlanda. Three criteria are used to choose the design.

Estimation Tests on whether algorithm i and delay-share strategy j behave differently when aircraft wake-vortex mix l changes are made on estimates of wake-vortex mix ϕ_l and $(\alpha\phi)_{il}$ and $(\beta\phi)_{jl}$ interactions. The interactions should be as separable from other effects as possible.

Reuse Analysis will reuse observations taken in Experiment II.

Run time The additional runs in Experiment II took about 2 days to run.

Wake-vortex mix is abstracted to an overall proportion, rather than the proportion through particular IAF routes. Proportions of aircraft types sum to unity, so the wake-vortex design space simplex is a triangle. Figure 7.10 shows a representation of this. In this diagram, as the proportion of aircraft moves away from the bottom corner of the triangle, the proportion of heavy and light type aircraft increases. The problem is to decide which design points to run in the triangle.

Table 7.24: Experiment IV, Repeat of Experiment II η levels r_1, r_2, r_3

κ effect	λ effect		
	2	1	3
2	r_1	r_2	r_3
3	r_2	r_3	r_1
4	r_3	r_1	r_2

A repeat of the Experiment III design would allow estimation of three levels of wake-vortex mix effects. For instance, the three evenly spaced points on the vertical centre line in Figure 7.10 might be chosen. In this case estimated coefficients would represent the effect of increasing proportion of heavy and light type aircraft equally. Another design might choose points along a horizontal line in the diagram. Here coefficients would relate the effect of decreasing proportion of medium aircraft, and shifting this towards heavy or light aircraft. Both these designs and coefficient interpretations would be useful - so useful that a third design is preferred where the two coefficients types may be estimated. Nine design points are chosen, as in Figure 7.10. These nine design points allow estimation of both coefficient types.

A full factorial experiment with nine further wake-vortex mix design points run on top of Experiment II output would be computationally prohibitive, so a fractional factorial experiment is preferable. If interpretation of wake-vortex coefficients is ignored then to estimate the interesting interactions each algorithm and delay-share strategy pair (i, j) needs to be run at least once at each of the nine wake-vortex levels l. It follows that each traffic level k will then be run three times for each pair (i, j). Looking at the problem this way it is possible to avoid pairing algorithms as in the Experiment III design. However, analysis is focused on estimation of a four level effect κ_m of the shift away from medium type flights, and a three level effect λ_n of the bias towards heavy or light aircraft. It is important estimates of κ_m and λ_n are not biased towards particular α_i, β_j or δ_k, i.e. the same number of (i, j, k) combinations should be run at each level of m and n. One design that balances m, n and j levels as well as possible is a three fold repetition of Experiment III. Represent the runs made at the three η_l levels in this experiment r_1, r_2, r_3. Then rearranging these levels as in Table 7.24 ensures that each m and n has all (i, j) pairs. In this design the six sets of (i, j) pairs are still present and confounded with κ_m and λ_n, so cannot be estimated.

Choice of Wake-vortex mix levels Proportion of medium aircraft is decreased by 3%, 7% and 10 % evenly across arrival routes. Heavy and light proportions are increased with the ratio 1:1, 2:1 and 1:2. This is pictured in Figure 7.10, with the maximum shift away from medium set to 10%. The actual numbers used are found in the Appendix. The levels run represent significant change in traffic mix but are deliberately not too different from current operations so that the solution space represents a range of traffic scenarios that might occur (Robinson 2004).

7.5.3 Analysis

Since the data analyzed is built on top of Experiment II only significant κ_m, λ_n effects and their interactions with α_i or β_j are commented on here. Unless otherwise stated, the same trends for α_i, β_j, δ_k and their interactions found in Experiment II are again seen. Statistical output from this experiment is copious due to the number of effects, so most is not included in the text. Some is found in the Appendix for reference.

Delay

Analysis by EDFIT found κ_m and some $(\alpha\kappa)_{im}$ significant, but not λ_n or other interactions with α_i or β_j. A linear model fit to $\mathrm{E}\left[Y_{ijkmn}\right] = \alpha_i + \beta_j + \delta_k + \kappa_m + (\alpha\beta)_{ij} + (\alpha\kappa)_{im}$ by WLS found all terms significant from 1000 bootstraps. The κ_m point estimates are 1.15, 2.12 and 3.51. Interactions $(\alpha\kappa)_{im}$ were significant for for all algorithms against the base with $m = 4$, and all algorithms except $(\alpha\kappa)_{62}$ and $(\alpha\kappa)_{63}$. There was no significant difference between the significant terms. Mean of the significant point estimates were $(\alpha\kappa)_{.2} = -0.22$, $(\alpha\kappa)_{.3} = -0.43$ and $(\alpha\kappa)_{.4} = -0.56$. Correlation coefficient between the sample mean at each design point and the prediction from the linear model was 0.9730. A variance model $\mathrm{Var}\left(Y_{ijkmn}\right) = \sigma^2 \exp\left(\delta_k + \kappa_m\right)$ correlated with observed design point sample variance at 0.958. Table 7.25 shows how variance increased significantly as both δ_k and κ_m levels increased, with δ_k having a larger effect.

Landing rate

Analysis of distribution through EDFIT only found δ_k significant, and a linear model also reduced to $\mathrm{E}\left[Y_{ijkmn}\right] = \delta_k$.

Holding

Use of EDFIT found κ_m and some $(\alpha\kappa)_{im}$ and $(\beta\kappa)_{im}$ significant. A model fit by WLS to $\mathrm{E}\left[Y_{ijkmn}\right] = \alpha_i + \beta_j + \delta_k + \kappa_m + (\alpha\beta)_{ij} + (\alpha\delta)_{ik} + (\alpha\kappa)_{im} + (\beta\delta)_{jk} + (\beta\kappa)_{jm}$

Table 7.25: Experiment IV: Delay, variance model

Coefficient	Point Estimate	95% C.I. Lower	Upper
σ^2	0.075	-	-
δ_2	0.508	0.248	0.769
δ_3	1.283	1.023	1.544
δ_4	2.550	2.290	2.810
κ_2	0.125	-0.048	0.299
κ_3	0.205	0.031	0.378
κ_4	0.201	0.028	0.375

yielded point estimates for the κ_m increasing from 0.41 to 1.21 as m increased and traffic moved to fewer medium aircraft. The size of these effects compared to traffic level effects δ_k that ranged from 0.30 to 5.59 as k increased. The $(\beta\kappa)_{jm}$ interactions decreased from -0.31 to -0.87 with no significant difference between $(\beta\kappa)_{ij}$ for $i = 1, \ldots, 3$ given a value j. The $(\alpha\kappa)_{im}$ interactions also decreased as κ_m increased to -0.56. A fitted model for variance
$\mathrm{Var}\,(Y_{ijkmn}) = \sigma^2 \exp(\alpha_i + \beta_j + \delta_k + \kappa_m + (\beta\delta)_{jk})$ correlated to observed variance with coefficient 0.933. All terms were significant with the κ_m estimates negative.

Approach sectors

The EDFIT methodology found κ_m significant. A fitted variance model $\mathrm{Var}\,(Y_{ijkmn}) = \sigma^2 \exp(\delta_k + \kappa_m)$ had correlation at 0.947 with design point sample variance. The κ_m coefficients were positive but small in comparison to δ_k coefficients.

Stability of AMAN advice

Looking for difference in distribution with EDFIT found $(\alpha\kappa)_{i3}$ and $(\alpha\kappa)_{i4}$ for $i = 1, \ldots, 5$ significant at the 95% level with Fourier components significant in mean. This indicated that the stability of delay advice of algorithms, in comparison to algorithm 1, might change as wake-vortex traffic mix moved away from medium type aircraft domination. Fitting a linear model
$\mathrm{E}\,[Y_{ijkmn}] = \alpha_i + \beta_j + \delta_k + \kappa_m + \lambda_n + (\alpha\delta)_{ik} + (\alpha\kappa)_{im} + (\beta\delta)_{jk}$ returned all coefficients significant with ANOVA. Errors were not normal so the WLS bootstrap was used to test for significance of individual coefficients. The $(\alpha\kappa)_{im}$ were negative and all significant at $m = 2, 3$. Coefficients values and their confidence ranges are shown in Figure 7.11. As traffic mix became more complicated, the new algorithms stability increased more than the FCFS at IAF sequencing algorithm.

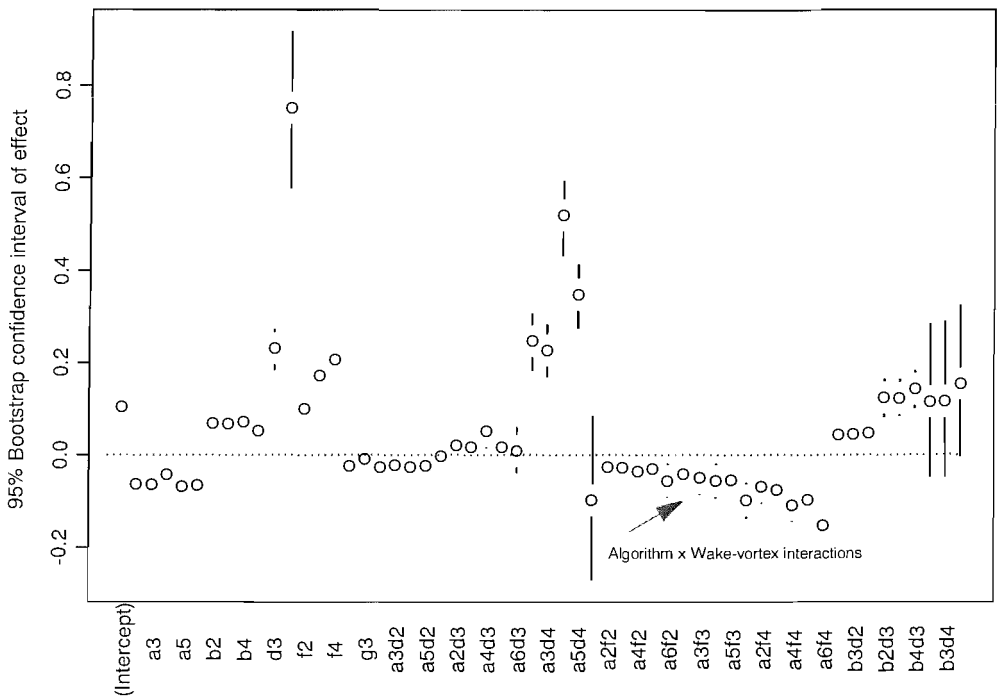


Figure 7.11: Experiment IV: Stability, linear model coefficient estimates with empirical 95% confidence intervals from 1000 bootstraps

7.5.4 Findings

The following findings are based on the range of traffic samples simulated. The conclusions may be applicable to an airport with similar aircraft arrival characteristics.

Delay There is some evidence to suggest that the proportion of medium type aircraft in the traffic mix affects delay, and sequencing algorithms reduce mean delay more in comparison with FCFS at IAF when traffic mix contains a lesser proportion of medium type aircraft.

Landing rate The only factor affecting landing rate was traffic intensity.

Holding time As aircraft traffic mix moved away from medium type aircraft, the mean amount of holding increased while its variability decreased. Use of a non holding delay-share strategy was found to have a greater sized effect in reducing amount of holding, as traffic-mix moved away from medium type aircraft domination.

Time in approach sectors There was no evidence to suggest that choice of sequencing algorithm or delay-share strategy affected the time in the approach sectors, as wake-vortex level changed across the experimental range, in different ways to those found in Experiment II.

Stability of AMAN advice As traffic mix became more complicated, moving away from medium type aircraft domination, the new algorithms stability increased more that the FCFS at IAF sequencing algorithm.

7.6 Findings and general conclusions

The computer simulation experiment work reported in this chapter was made to help answer the question posed at its start. Some benefits and risks from change to landing sequence and delay-share strategy were analyzed on a validated simulation model of arrival traffic into airspace around Stockholm Arlanda airport. These changes were looked at in conjunction with potential movements in arrival rates, local and international mix of the traffic, and wake-vortex category mix of arrival traffic. The results depend on the data used and are particular to the model, but may also be applicable to other arrival airspace with similar characteristics. From the experimental results some general conclusions of interest to policy makers in Air Traffic Control may be drawn. Here the specific results are summarized and some of these general conclusions formed.

Under the range of traffic conditions simulated no benefit to landing rate or delay was found using a different sequencing algorithm to FCFS at runway. Delay to aircraft was significantly greater when sequencing arrivals FCFS at IAF points than from use of other methods. However, nothing was gained through improved sequencing algorithms: sequencing FCFS at runway performed as well as any other. The effect for FCFS at IAF became more pronounced as wake-vortex mix shifted up to 10% away from the validated models medium aircraft-type domination. Landing rate was not found to be affected by any sequencing algorithm or delay-share strategy.

System behaviour was found to vary by simulation AMAN setup. Sequencing algorithm affected time spent in holds: sequencing FCFS at IAF increased time in holds relative to other algorithms because it also increased delay. Delay-share strategy also affected holding time. As traffic intensity increased, or traffic became less predominantly dominated by medium type aircraft, delaying aircraft before IAF points had greater effect in reducing time holding. Variability of the time holding was reduced by delaying pre-IAF, but the relative difference decreased as traffic intensity increased. Time in approach sectors was affected by delay-share strategy only. Analysis found that pre-IAF delay-share strategies increased the amount of time aircraft would spend in pre-approach sectors and, as traffic intensity increased, the proportion of time in pre-approach sectors also increased.

Different AMAN systems advise different landing sequences and update the sequence with new aircraft as time progresses. A risk to implementation of the sequencing algorithms is the stability of the advice they form. Choice of sequencing algorithm had an effect on the stability of delay-to-lose advice. First, algorithms sequencing at IAF points were more stable that those at the system boundary. This included a FCFS at runway algorithm run at the system boundary, where predicted time at runway was not subject to error. That is, even a FCFS algorithm was not as stable as others operating later on in airspace. Secondly, for those algorithms operating pre-IAF, significant differences were found between FCFS at IAF, a heuristic algorithm and another group that included FCFS at runway. The FCFS at IAF algorithm was the most unstable as a result of generating larger delay than other algorithms. The heuristic suffered from lack of CPS constraints on the maximum shift an aircraft may make from its FCFS position. This constraint was used by the other smart algorithms where there was no difference in stability advice to FCFS at runway. Finally, the variability in advice given by all algorithms increased as traffic intensity increased, and the heuristic and FCFS at IAF became even more unstable relative to others.

The findings are now summarized in more general terms.

Conclusion 1 Improved sequencing techniques should not be regarded as a panacea to reduce delay and increase land rate. The ability to realize these benefits depends on arrival airspace and traffic characteristics.

The arrival aircraft sequencing problem has been studied in order that real systems be developed where advised landing sequences reduce delays and increase landing rate. However, results from the series of experiments in this chapter have not found these expected benefits.

Conclusion 2 Different sequencing algorithm and delay-share strategies in an AMAN system may cause different system behaviour.

If sequence advice is to be made then methods of implementing landing sequences and assigning delay become important, and may cause system behaviour to change. For example, moving delay back from holding points necessitated delay in pre-approach sectors in the model. This might have knock-on effects elsewhere such as for Eurocontrol CFMU slot allocation for aircraft taking off at other airports.

Conclusion 3 Choice of sequencing algorithm and delay-share strategy will affect stability of advice to controllers, and quality of information to other users.

Improvements to landing rate or delay indicators are not the only reasons for bringing in AMAN technology. Other users in the ATC system such as airlines or baggage handlers may benefit from accurate forecast landing times. Sequencing further back from the airport with an AMAN tool could provide this information. However, the further back from the airport the AMAN system begins, the more variable the information will be. This was found in the model even when sequencing FCFS. Also, as more aircraft used the runway, the advice from all the experimental AMAN systems became more unstable. Choice of sequencing algorithm affected the variability in delay advice: smart algorithms with CPS constraints were shown to produce more stable advice than otherwise.

Chapter 8

Airport runway capacity experiments

Question What is the capacity of the simulation model of Stockholm Arlanda runway to service arrivals?

This question is investigated through experimentation on the computer simulation model. The experiments are based on the Eurocontrol Performance Review unit (PRC) view of airport capacity. This unit is responsible for choosing performance indicators for various parts of the ATC system. Their report on an ATM Performance measurement system (Performance Review Unit 1999) defines:

> "Declared airport capacity is the maximum sustainable capacity during periods of normal weather, taking into account all the various limiting parameters. There is a trade-off between declared airport capacity and delay, and the declared capacity should be set at a level which does not lead to unacceptable delay in any component parts of the capacity chain ... Unconstrained runway capacity is defined as the maximum movements per hour attainable from the configuration of runways."

Airport runway capacity is thus linked with delay. The first experiment reported in Section 8.1 looks at airport capacity through aircraft delay attributable to runway sequencing. Airport capacity also relates to the maximum landing rate possible. That is, if improved sequencing results in more aircraft landing per hour with acceptable levels of delay than FCFS, then airport capacity is seen to increase. The possibility of increasing airport capacity through increase in landing rate is examined in the second experiment of Section 8.2. Section 8.3 completes the chapter by describing the overall results of the experimentation in relation to the above question. It concludes with some remarks on what significance the results may have for AMAN technology at other airports.

8.1 Airport runway capacity experiment: Delay

8.1.1 Design

Analysis of delay has already been made on a number of detailed experiments in the previous chapter. This experiment is run to add more data to previous output. Specifically, the experiment is run to investigate how sequencing algorithm i, delay-share strategy j, traffic intensity k and wake-vortex traffic mix l affect delay Y_{ijkl}. Extra traffic levels are run between the top two levels from Experiment IV in Section 7.5. The analysis of Section 7.5.3 did not find the wake-vortex level effect of increasing mix towards heavy or light significant, so these effects are not built into this experiment, and a total of four wake-vortex levels are run. Three basic criteria are used to decide the design.

Estimation The interest is in algorithm α_i and delay-share strategy β_j main effects and their interactions with traffic level δ_k and wake-vortex mix κ_l effects. A good design will separate estimates as much as possible.

Reuse Use observations from the top two traffic levels in Experiment IV and corresponding levels of κ_l.

Run time The experiment in Section 7.5 took about a week to complete.

A fractional factorial design allows more traffic levels to be run, so no further runs were made at Experiment IV (in Section 7.5) design points. Experiment IV was a mixed fractional/full factorial design where all algorithm and delay-share strategy combinations were run for the base wake-vortex level, but only certain combinations for new wake-vortex levels. This design continues the idea. To balance the fractional part of the design all traffic levels k require the same number of (i, j, l) combinations. This constraint coupled with reuse of Experiment IV levels means the total number of traffic intensity levels must be a multiple of three. Six in total is a reasonable number for run time. Given this, a new problem is to choose combinations of (i, j) to run at traffic k and wake-vortex l levels. The goals of combination choice are:

1. Each (j, k) combination in the fractional part of the design sees every possible algorithm i once.
2. Delay-share strategies $j = 2, 3, 4$ have the same number of runs with algorithms $i = 1, 2, 3, 4, 5$.
3. Each wake-vortex level l has the same number of runs made by each algorithm i.

Table 8.1: Airport runway capacity experiment: Delay, fractional factorial design with new design points in bold type

Traffic	2	2	2	2	3	3	3	3	4	4	4	4	W-V Level
intensity	1	2	3	4	1	2	3	4	1	2	3	4	D-S strategy
6	1	2	1,5	3,4	6	3,4	2	1,5	5	1,5	3,4	2	
5	**5**	**2,3**	**5**	**1,4**	**1**	**1,4**	**2,3**	**5**	**6**	**5**	**1,4**	**2,3**	
4	**6**	**4,5**	**1,2**	**3**	**5**	**3**	**4,5**	**1,2**	**1**	**1,2**	**3**	**4,5**	
3	**1**	**1**	**3,4**	**2,5**	**6**	**2,5**	**1**	**3,4**	**5**	**3,4**	**2,5**	**1**	
2	**6**	**3,5**	**4**	**1,2**	**5**	**1,2**	**3,5**	**4**	**1**	**4**	**1,2**	**3,5**	
1	5	1,4	2,3	5	1	5	1,4	2,3	6	2,3	5	1,4	

4. Pairs of algorithms run at (j,k) combinations are randomized. This is a problem for delay-share strategies 2 to 4: two sequencing algorithms need be run for these at some (j,k,l) levels. For delay-share strategy 1 there are three algorithms and three wake-vortex levels so no pairs are needed.

For delay-share strategies 2 to 4, 18 algorithm pairs need be run at each wake-vortex level, with 15 possible. A good randomization might run all pairs at least once with no pair more than twice. A computer program was written to loop through an upper bound of 155,520 possible algorithm pair designs. This found 4 designs where all pairs were run at least once and none more than twice. However, none were satisfactory because the arrangement of algorithms across two traffic levels was repeated, introducing a potential traffic level effect bias. Table 8.1 shows the design points chosen for the fractional-factorial part of the design. Algorithm pairs (1,3) and (2,4) are not run, pairs (1,2), (2,3), (3,4) and (5) are run twice, all others once. This design does not balance pairs of algorithms as well as technically feasible, but does not repeat combinations across traffic levels, randomizes the algorithms well and satisfies the first three design goals.

8.1.2 Analysis

Mean positive delay

Similar results to those in Experiment IV are found using the analysis methodology of Section 6.2, since only new factor levels are introduced. The final fitted linear model is $\mathrm{E}\left(Y_{ijkl}\right) = \alpha_i + \beta_j + \delta_k + \eta_l + (\alpha\beta)_{ij}$. A Box-Cox transformation (-0.25) that stabilizes variance and normalizes the output finds the same coefficients significant as results from the bootstrap. These are presented in Table 8.2, based on 1000 bootstraps. The model predicts the simulation mean of mean positive delay at each design point well with a correlation coefficient at 0.9724.

Table 8.2: Airport runway capacity experiment: Delay, mean positive delay linear model significant coefficients

Coefficient	Value	95% C.I.*	
		Lower	Upper
μ	3.604	3.426	3.743
α_2	-1.514	-1.727	-1.315
α_3	-1.578	-1.797	-1.341
α_4	-1.643	-1.859	-1.444
δ_3	0.736	0.636	0.800
δ_4	1.226	1.064	1.287
δ_5	2.001	1.845	2.087
δ_6	2.653	2.490	2.762
δ_7	2.805	2.580	2.926
κ_2	0.162	0.081	0.274
κ_3	0.399	0.312	0.510
κ_4	0.977	0.850	1.102
$(\alpha\beta)_{52}$	-1.746	-2.057	-1.414
$(\alpha\beta)_{53}$	-1.603	-1.918	-1.291
$(\alpha\beta)_{54}$	-1.730	-2.048	-1.416

*1000 bootstraps

Interpretation Runway capacity as measured by mean of mean positive delay is not seen to increase through choice of sequencing algorithm or delay-share strategy under the conditions simulated. The difference attributable to algorithm is due to a technical inefficiency in the system as-is described in Section 7.2.4. That is, if it were possible to achieve minimum separations between all aircraft then capacity would increase regardless of the sequence method chosen from those in the simulation.

Threshold

No difference was found using the analysis methodology of Section 6.2. However, the methods in Section 6.3 to further analyse in-run outputs did reveal differences. Charts are used to display point estimates and confidence intervals for estimated coefficients in a number of cases. In these charts the x-axis Latin letters are used to represent Greek coefficients used in the text, where a = α, b = β, d = δ and e = κ. Choice of logit model was made based on Aitken's information criteria, step-wise removing terms from the complete model (Venables & Ripley 1999). The models chosen are shown in Table 8.3. Diagnostics on these model fits based on Pearson residuals and comparison to bootstrap coefficient confidence intervals were satisfactory, so the asymptotic theory results are presented. The models have no interaction terms between α_i or β_j and δ_k or κ_l so sequencing algorithms and

Table 8.3: Airport runway capacity experiment: Delay, logit model summaries

α_t	α_s	Model	df	Residual deviance
13	0	$\alpha_i + \beta_j + \delta_k + \kappa_l + (\alpha\beta)_{ij}$	190	122.44
13	0.05	$\alpha_i + \beta_j + \delta_k + \kappa_l + (\alpha\beta)_{ij}$	190	160.94
15	0	$\alpha_i + \beta_j + \delta_k + \kappa_l + (\alpha\beta)_{ij}$	190	129.02
15	0.05	$\alpha_i + \beta_j + \delta_k + \kappa_l$	199	172.05
17	0	$\alpha_i + \beta_j + \delta_k + \kappa_l + (\alpha\beta)_{ij}$	190	163.72
17	0.05	$\alpha_i + \beta_j + \delta_k + \kappa_l$	199	188.73

delay-share strategies do not appear to affect the probability of exceeding a threshold delay differently as traffic level or wake-vortex mix change.

Sensitivity S did not cause the relative difference in coefficient estimates to change as it increased from 0 to 10%. The effect of increasing S towards 10% was to reduce the differences between coefficients. Figure 8.1 shows estimated coefficient values for $T = 13$ minutes and $S = 0.05$. It shows that as traffic level and wake-vortex mix increased, the probability that greater than 5% of aircraft were delayed above 13 minutes increased. Algorithms 5 and 6 also increased this probability over the base. A different trend is shown in Figure 8.2 for $T = 15$ and $S = 0$. Here algorithm 4 increased the probability of failure, but algorithms 5 and 6 actually decreased the probability. The same pattern occurred at threshold $T = 17$. This suggests that algorithms 5 and 6 delay a large proportion of aircraft between 13 and 15 minutes. Table 8.4 shows 95% simultaneous confidence intervals on the difference in proportion of aircraft delayed between 13 and 15 minutes by algorithm. These confidence intervals are based on pairing delay proportion by traffic sample and running 9,000 bootstraps per pair. The Bonferroni inequality is used to get 95% simultaneous limits. There is a significant difference between algorithms 5 and 6 and others. Thus, algorithms 5 and 6 delay a greater proportion of aircraft between 13 and 15 minutes than do the other algorithms in the simulation, while algorithm 4 delays more aircraft greater than 15 minutes than do the other simulation sequencing algorithms.

Interpretation The analysis above has shown that there is a difference in the number of aircraft delayed above a threshold. Algorithms 2, 5 and 6 have deadlines set to 15 minutes. Algorithms 5 and 6 also have a constraint on the order of aircraft (FCFS at IAF), a feature that appears to make them delay more aircraft close to their deadline than algorithm 2 and other sequencing algorithms. Algorithm 4 has no algorithmic constraint on deadline or order at IAF points. Lack of the deadline constraint seems to cause this heuristic (minimizing total delay) to

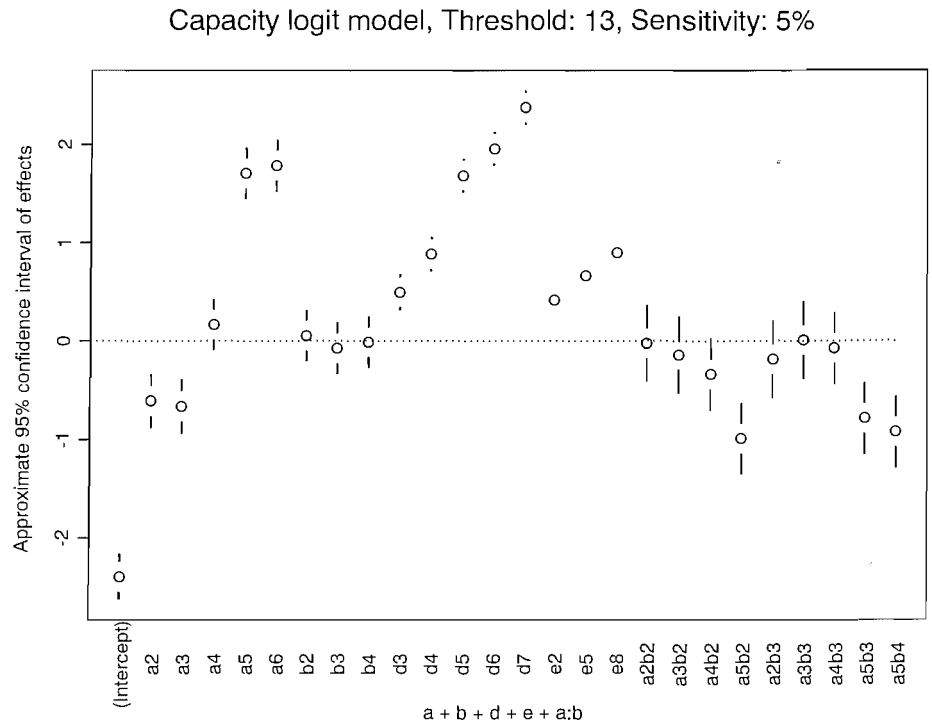


Figure 8.1: Airport runway capacity experiment: Delay, logit model, $T = 13$, $S = 0.05$

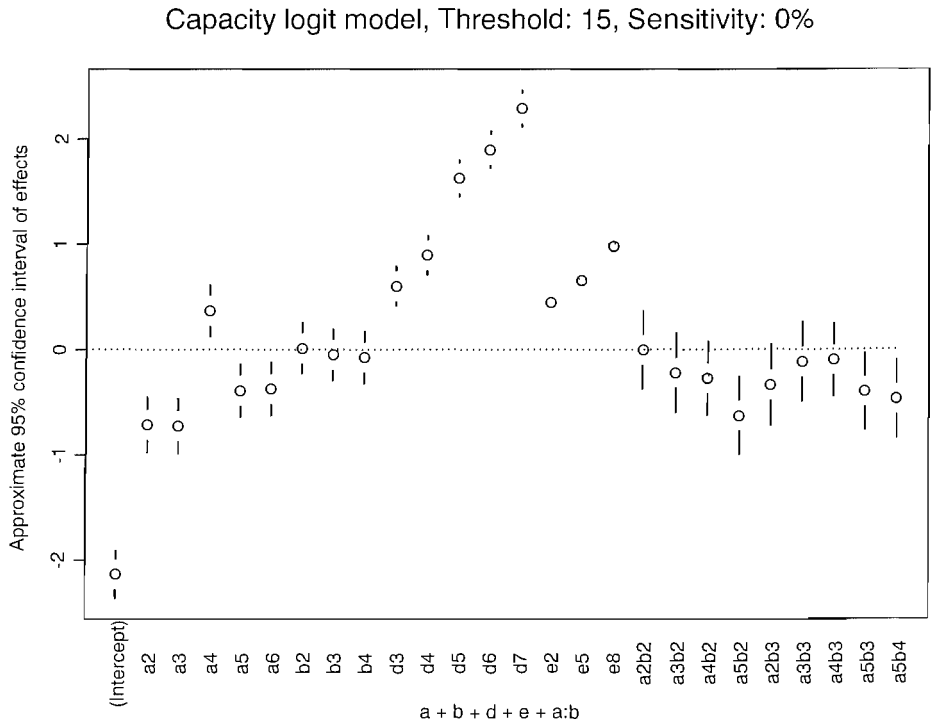


Figure 8.2: Airport runway capacity experiment: Delay, logit model, $T = 15$, $S = 0$

Table 8.4: Airport runway capacity experiment: Delay, 95% bootstrap* simultaneous confidence intervals on difference in mean percentage(%) delayed in [13min, 15min]

Algorithm	2	3	4	5	6
1	(0.744, 0.922)	(0.822, 0.997)	(0.940, 1.159)	(-2.811, -2.497)	(-5.259, -4.615)
2		(0.031, 0.124)	(0.099, 0.332)	(-3.732, -3.334)	(-6.195, -5.394)
3			(0.014, 0.261)	(-3.814, -3.400)	(-6.267, -5.454)
4				(-3.919, -3.584)	(-6.349, -5.618)
5					(-2.596, -2.077)

*9000 Bootstraps per pair

delay aircraft above an arbitrary threshold more often than FCFS algorithms and other smart sequencing algorithms with deadline constraints.

Target

The proportion of aircraft delayed above a target, given the target is not met is investigated here. The methodology used is described in Section 6.3. Results are presented for the interesting case where algorithm 4 is found to delay more aircraft with delay greater than 15 minutes than other algorithms. The target must have a probability of being exceeded for a beta distributed model to make sense, so only output from the highest traffic level is used in fitting models with covariates α_i, β_j and κ_l.

Analysis Maximum likelihood beta distribution fits to the proportion of aircraft delayed greater than 15 minutes at each design point were tested by comparing plots of empirical and fitted cumulative distribution functions and using Kolmogorov-Smirnov goodness-of-fit statistics. The p-values from each test are not independent so Fisher's composite hypothesis test (D'Agostino & Stephens 1986) cannot be used. However, visual inspection of the fits and Kolmogorov-Smirnov statistics do not lead to a rejection of the validity of the beta models: p-values are all contained in the range [0.2,0.97]. Fits of linear models to the parameters find algorithm 4 and 5 main effects significant. This general test is borne out with a linear model for the mean shown in Table 8.5, where the multiple R^2 value is 0.8478. The assumption that residuals are normal cannot be rejected with a Shapiro-Wilks test W = 0.949 and p-value 0.0971. An important result from this analysis is the α_4 coefficient: it is significant and negative. That is, if aircraft delay greater than a target of 15 minutes, then a significantly lower proportion will delay greater than 15 minutes when sequenced by algorithm 4 than the other sequencing algorithms.

Table 8.5: Airport runway capacity experiment: Delay, linear model of mean proportion above 15 min target, given target missed

Coefficient	Value	SE	t score	p-value
μ	0.164	0.013	12.661	0.000
α_2	0.004	0.012	0.291	0.773
α_3	0.019	0.012	1.590	0.125
α_4	-0.059	0.012	-4.831	0.000
α_5	-0.031	0.011	-2.822	0.009
α_6	-0.010	0.021	-0.511	0.614
κ_2	0.051	0.010	4.856	0.000
κ_3	0.061	0.010	5.849	0.000
κ_4	0.077	0.011	7.305	0.000
β_2	-0.007	0.014	-0.473	0.640
β_3	-0.001	0.014	-0.048	0.962
β_4	-0.014	0.014	-0.972	0.341

Interpretation Given the target that all aircraft delayed less than 15 minutes has a positive probability of being exceeded, and the target is exceeded, a lower proportion of aircraft are delayed greater than 15 minutes using the heuristic than the other sequencing algorithms. In the simulation when a dynamic program could not make a sequence with deadline 15 minutes it ignored the constraint. Thus, it seems sequencing FCFS or with CPS constraints without a deadline constraint results in a higher proportion of aircraft delayed greater than 15 minutes, when this happens, in comparison to the heuristic.

8.1.3 Findings

Under the range of traffic conditions investigated on a simulation model of arrivals into Stockholm Arlanda the following general delay-related capacity conclusions may be drawn:

- No significant reduction in mean positive delay was found through clever sequencing.
- Adding deadlines resulted in significantly more aircraft delayed 2 minutes under the deadline for two dynamic programs with FCFS at IAF constraints.
- The heuristic failed to meet a threshold used as a deadline in dynamic programs a significantly greater proportion of times than FCFS and the dynamic programs.
- If a target is set such that there is a positive probability it is exceeded then, given it has been exceeded, the heuristic delays a lower proportion of aircraft

greater than the target than the dynamic programs or FCFS.

These results could be applicable on arrival traffic into a similar airspace.

8.2 Airport runway capacity experiment: Landing rate

Despite changing the arrival traffic mix significantly from current mix at Stockholm Arlanda, both in arrival rates and wake-vortex mix, the only difference in landing rate has been attributable to achieving minimum separations where possible. *A priori* it is known that the algorithms should produce different sequences with different characteristics. As a result, it becomes interesting to ask:

Question Under what conditions would an AMAN sequencing algorithm increase runway landing rate over FCFS?

8.2.1 Basic design

The design chosen has the following characteristics.

Algorithms To cut down on run time the only computationally expensive algorithm run is the dynamic program minimizing makespan C_{MAX}. The FCFS rules and heuristic are also run.

Delay-share strategies Use "In hold" for FCFS at IAF and "Early as possible" for the other sequencing algorithms.

Constrained position shifting Set to three for the heuristic. Dynamic program CPS constraints are specified in the experiments that follow.

Deadlines Set to ∞.

Update criteria Every aircraft.

Wake Vortex mix Validated mix and three further levels. The further levels have equal percentage of heavy and light type aircraft where medium percentages are 75%, 50% and 25%. The exact parameters used are found in the Appendix.

Arrival route The validated proportions in Table 5.4 are used.

Runs 50 repeats per design point.

Landing rate is calculated using the landings in the second hour. The basic output model is of the form:

$$Y_{ijk} = \alpha_i + \kappa_j + (\alpha\kappa)_{ij} + \epsilon_{ijk} \tag{8.1}$$

for $i = 1, \ldots, 8$, $j = 1, \ldots, 4$ and $k = 1, \ldots, 50$ where α_i is the effect of algorithm combination i, κ_j the effect of wake-vortex mix j and ϵ_{ijk} a random error. This ensures a complete, balanced design that covers a wide range of the wake-vortex space. It was chosen to find a difference in landing rate, if it exists. Special attention is paid to set-up parameters of the dynamic program minimizing makespan C_{MAX} because this algorithm should produce the highest possible landing rate: if it is possible to increase landing rate this algorithm should do so.

8.2.2 Analysis methods

The analysis methods of Section 6.2 are used to investigate simulation output.

8.2.3 Question A

If aircraft arrival rate is fixed around the runway capacity of 100% medium type aircraft, does wake-vortex mix cause algorithms to have different landing rates?

Experiment parameters Two hour period with arrival rate set to 50 in both hours. The CPS constraint is set to ∞ in the dynamic program.

Analysis Output histograms, split by algorithm and wake-vortex level, are presented in Figure 8.3. From left to right in the plot the wake-vortex levels run from 1 to 4. Going from bottom to top the algorithms also run from 1 to 4, i.e. FCFS at IAF - dynamic program minimizing C_{MAX} - FCFS at runway - heuristic. Inspection of the output histograms appears to show a difference in landing rate distribution due to wake-vortex level and algorithm. It appears that the difference in landing rate occurs more with the heuristic than the dynamic program. Testing for an overall difference with Friedmans test (Rice 1995) by blocking on traffic sample yields a $\chi^2 = 136.1091$ on 3 degrees of freedom with p-value = 0. That is, the test rejects the hypothesis that there is not a difference. An EDFIT table formed from 1000 bootstraps is shown in Table 8.6. This finds that the differences in landing rate distributions are attributable to the wake-vortex level κ_l, the heuristic sequencing algorithm α_4 and its interactions with κ_l.

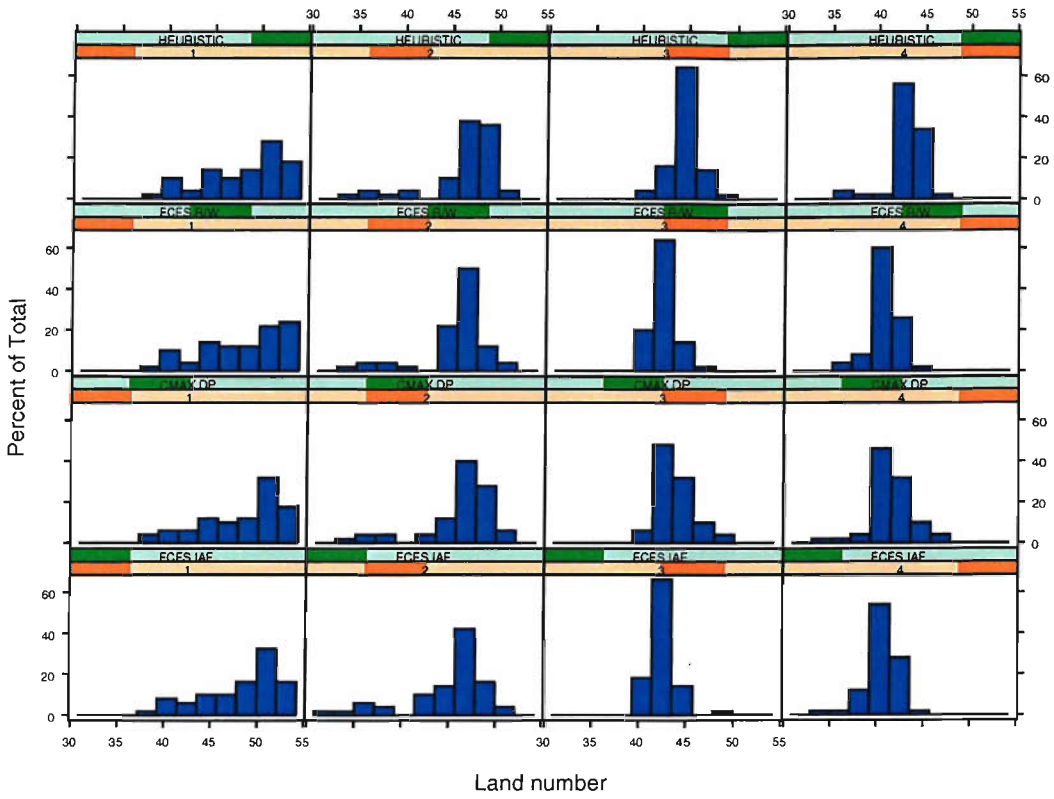


Figure 8.3: Airport runway capacity experiment: Landing rate, Question A, Histogram of landing rate, arrival rate hour 1 = 50

Table 8.6: Airport runway capacity experiment: Landing rate, Question A, full EDFIT table, 1000 bootstraps

Coefficient	T^2	C_1	C_2	C_3	C_R
α_2	8.67 (.159)	5.03 (.202)	3.15 (.053)	0.25 (.416)	0.24 (.987)
α_3	0.22 (1)	0.03 (.926)	0.05 (.817)	0.02 (.799)	0.11 (.999)
α_4	**81.52 (.000)**	**39.42 (.002)**	**35.51 (.000)**	**2.74 (.005)**	**3.85 (.000)**
κ_2	**298.59 (.000)**	**294.35 (.000)**	0.05 (.804)	0.98 (.106)	**3.22 (.002)**
κ_3	**200.65 (.000)**	**184.49 (.000)**	**8.65 (.000)**	**3.35 (.005)**	**4.17 (.000)**
κ_4	**47.76 (.000)**	**18.78 (.015)**	**21.21 (.000)**	**5.94 (.000)**	1.82 (.047)
$(\alpha\kappa)_{22}$	9.51 (.369)	5.66 (.324)	3.39 (.132)	0.1 (0.73)	0.36 (.997)
$(\alpha\kappa)_{32}$	0.84 (.998)	0.02 (.949)	0.06 (0.84)	0.05 (0.79)	0.71 (.945)
$(\alpha\kappa)_{42}$	**86.11 (.000)**	**43.28 (0.01)**	**35.6 (.000)**	2.55 (.069)	**4.69 (.014)**
$(\alpha\kappa)_{23}$	6.06 (.605)	0.11 (0.87)	4.99 (.083)	0.2 (.614)	0.76 (.935)
$(\alpha\kappa)_{33}$	2.5 (.917)	0.83 (0.73)	0.7 (.503)	0.03 (.804)	0.94 (.833)
$(\alpha\kappa)_{43}$	**73.51 (.003)**	13.6 (.144)	**50.16 (.000)**	2.84 (.047)	**6.9 (.001)**
$(\alpha\kappa)_{24}$	6.85 (.534)	1.42 (.602)	0.54 (0.57)	3.31 (.032)	1.58 (.542)
$(\alpha\kappa)_{34}$	0.32 (1)	0.02 (.948)	0.07 (.818)	0 (.965)	0.22 (1)
$(\alpha\kappa)_{44}$	**43.99 (.011)**	0.24 (.848)	**11.3 (.007)**	**24.36 (.000)**	**8.09 (.000)**

Table 8.7: Airport runway capacity experiment: Landing rate, Question A, simultaneous 95% confidence interval of coefficient of variation from 9600 bootstraps

Algorithm	Wake-vortex level	Point Estimate	95% C.I. Lower	Upper
1	1	0.09	0.0634	0.1110
1	2	0.09	0.0477	0.1320
1	3	0.03	0.0184	0.0464
1	4	0.04	0.0231	0.0587
2	1	0.09	0.0619	0.1154
2	2	0.08	0.0351	0.1160
2	3	0.04	0.0277	0.0521
2	4	0.05	0.0276	0.0685
3	1	0.09	0.0668	0.1131
3	2	0.08	0.0366	0.1083
3	3	0.03	0.0191	0.0406
3	4	0.04	0.0213	0.0520
4	1	0.09	0.0634	0.1114
4	2	0.08	0.0340	0.1118
4	3	0.03	0.0188	0.0425
4	4	0.04	0.0207	0.0598

Some difficulties arise in building parametric models to estimate the size of the effects on location. Log-linear models where $\ln(\mathrm{E}(Y_{il})) = \beta X_{il}$ with $\mathrm{Var}(Y_{il}) = \sigma^2 \mathrm{E}(Y_{il})$ may be built to consider the discrete data. The assumption constant coefficient of variation is questionable for landing rate. Bootstrap experiment confidence intervals on the coefficient of variation split by design point shown in Table 8.7 find some significant differences. However, they are not very large in size. Another potential flaw is that known cut-off maximum landing rate is not included in the model. Bearing these flaws in mind, Generalized Linear model estimation routines found the full model $\ln(\mathrm{E}(Y_{il})) = \alpha_i + \kappa_l + (\alpha\kappa)_{il}$ to have residual deviance of 166.56 on 784 degrees of freedom. The relatively small residual deviance means that a Poisson model of variance is not appropriate. However, a more general model with $\hat{\sigma} = 0.2059$ may be considered. Table 8.8 shows estimates and standard errors for this model. The main effects α_i do not return significant, but the interactions $(\alpha\kappa)_{4j}$ also picked up by EDFIT do.

Findings The dynamic program is an optimal algorithm - given a list of aircraft waiting to land it will always minimize the makespan of that sequence. However, the experiment results shows how in a real implementation it does not necessarily produce an optimal global sequence, and more surprising, it may be beaten by a heuristic minimizing delay. Further analysis on the data shows this is not due to

Table 8.8: Airport runway capacity experiment: Landing rate, Question A, log-linear model coefficients with $\hat{\sigma} = 0.2059$

Coefficient	Point Estimate	Std. Error
μ	3.8762	0.0092
α_2	-0.0004	0.0131
α_3	-0.0004	0.0131
α_4	-0.0017	0.0131
κ_2	-0.0726	0.0133
κ_3	-0.1253	0.0135
κ_4	-0.1685	0.0137
$(\alpha\kappa)_{22}$	0.0216	0.0188
$(\alpha\kappa)_{32}$	0.0097	0.0188
$(\alpha\kappa)_{42}$	0.0294	0.0188
$(\alpha\kappa)_{23}$	0.0227	0.0190
$(\alpha\kappa)_{33}$	-0.0001	0.0191
$(\alpha\kappa)_{43}$	0.0435	0.0190
$(\alpha\kappa)_{24}$	0.0213	0.0193
$(\alpha\kappa)_{34}$	0.0004	0.0193
$(\alpha\kappa)_{44}$	0.0524	0.0192

choosing an odd performance indicator: in a number of instances the makespan of the landing sequence (i.e. the land time of the last aircraft) is also less for the heurisitic. The reason for this is to do with choice of aircraft and imperfect information.

Consider an example where there are 3 aircraft ready to land with the same release dates, after an aircraft type M, with FCFS order $M - H - L$. Then the minimum makespan based on the separation matrix in Table 5.4 is 11, and the dynamic program may choose sequence A: $(M) - L - M - H$ with makespan of $5 + 3 + 3 = 11$. Suppose that with the heuristic the maximum shift is 1. Then sequence B with minimum total delay output by the heurisitic is $(M) - M - L - H$, also with makespan $3 + 5 + 3 = 11$. Time passes and the first two aircraft in the sequence land. From sequence A an aircraft type M is the last aircraft to land and type H is waiting. Sequence B has aircraft type L as the last to land and type H waiting to land. A new aircraft type L arrives. Then the dynamic program would choose final sequence $(M) - L - M - L - H$ with makespan $5 + 3 + 5 + 3 = 16$ but the heuristic final sequence is $(M) - M - L - L - H$ with makespan $3 + 5 + 3 + 3 = 14$. That is, it is possible to beat the exact deterministic algorithm with a heuristic when an updating process is present.

In the example above the heuristic betters the dynamic program because of imperfect information. If the algorithms had both known the full arrival sequence before it had occurred then the dynamic program would have minimized the final sequence makespan. The problem is related to choice of aircraft. In the example there are only a limited number of aircraft available to land. This is the case with the experiment just run as arrival rates are round about the runway capacity. Consequently the choice of aircraft to land is often limited to FCFS in order to ensure there are no gaps in the landing sequence. This is also the reason why there is no great difference between the dynamic program and FCFS algorithms. However, if many aircraft are waiting to land then there is a greater choice for the sequencing algorithm, and the kinds of situations where a heurisitic might do better than the dynamic program or other deterministic optimal algorithms will be more rare. This is illustrated in the next experiment, where arrival rate is such that there is a great choice of aircraft.

There was some evidence from this experiment that sequencing algorithms may increase runway capacity with the right wake-vortex traffic conditions. However, this did not occur systematically for arrivals with rates around the runway capacity - a dynamic program that maximizes runway throughput did not have a significant effect on landing number above FCFS at IAF points. The mean positive delays in this experiment ordered by wake-vortex level were 5.46, 8.34, 13.50 and 14.10 minutes. These are already high. How much higher do they need to be before there is enough choice of aircraft for sequencing algorithms to increase runway capacity?

8.2.4 Question B

If airspace is saturated with aircraft then will smart sequencing algorithms make a difference?

Experiment parameters Two hour period with arrival rate set to (100, 50). The CPS constraints are set to ∞ in the dynamic program.

Analysis Doubling arrival rate in the first hour lead the second hour landing aircraft to have mean delays of 32.54, 33.91, 36.41 and 36.36 minutes ordered by wake-vortex level. Figure 8.4 shows output histograms of the land number in the second hour. Notice how there are definite differences in location, and the dynamic program minimizing makespan has the highest landing rate. This shows how if the sequencing algorithm has sufficient choice of aircraft it is possible to

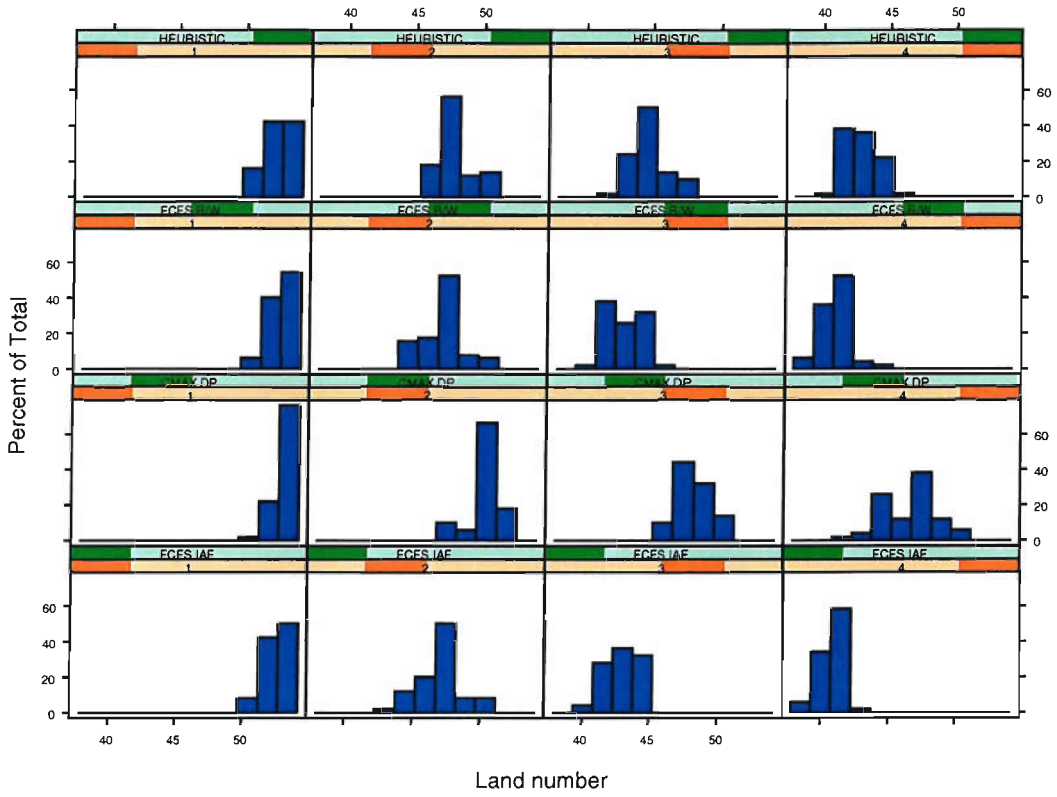


Figure 8.4: Airport runway capacity experiment: Landing rate, Question B, histogram of landing rate, arrival rate hour 1 = 100

Table 8.9: Airport runway capacity experiment: Landing rate, Question B, full EDFIT table, 1000 bootstraps

Coefficient	T^2	C_1	C_2	C_3	C_R
α_2	**236.15 (.000)**	**127.28 (.000)**	**95.06 (.000)**	**9.73 (.000)**	**4.08 (.000)**
α_3	0.23 (1)	0.02 (.951)	0.05 (.817)	0.05 (.719)	0.1 (1)
α_4	**54.69 (.000)**	6.56 (.154)	**16.57 (.000)**	**17 (.000)**	**14.56 (.000)**
κ_2	**574.69 (.000)**	**549.57 (.000)**	2.57 (.075)	**14.84 (.000)**	**7.71 (.000)**
κ_3	**268.43 (.000)**	**148.66 (.000)**	**107.9 (.000)**	**8.13 (.000)**	**3.75 (.000)**
κ_4	**69.35 (.000)**	9.2 (.091)	**22.66 (.000)**	**21.87 (.000)**	**15.63 (.000)**
$(\alpha\kappa)_{22}$	**242.1 (.000)**	**113.51 (.000)**	**116.84 (.000)**	3.74 (.024)	**8 (.000)**
$(\alpha\kappa)_{32}$	0.53 (1)	0 (1)	0.2 (.739)	0 (.972)	0.33 (.998)
$(\alpha\kappa)_{42}$	**56.48 (.005)**	8.14 (.244)	**12.63 (.007)**	**22.55 (.000)**	**13.16 (.000)**
$(\alpha\kappa)_{23}$	**245.5 (.000)**	13.58 (.152)	**180.37 (.000)**	**37.95 (.000)**	**13.6 (.000)**
$(\alpha\kappa)_{33}$	0.46 (1)	0 (.986)	0 (0.98)	0.08 (.773)	0.38 (.997)
$(\alpha\kappa)_{43}$	**61.88 (.001)**	0.09 (.923)	**16.65 (.001)**	**31.64 (.000)**	**13.5 (.000)**
$(\alpha\kappa)_{24}$	**93.28 (.000)**	1.99 (.577)	**20.31 (.001)**	**55.82 (.000)**	**15.17 (.000)**
$(\alpha\kappa)_{34}$	1.49 (.983)	0 (.984)	0.05 (0.88)	0.23 (.586)	1.2 (.719)
$(\alpha\kappa)_{44}$	36.92 (.024)	0.26 (.837)	0.5 (.541)	**8.52 (.000)**	**27.64 (.000)**

increase landing rates. An EDFIT test for overall significance returns $T = 96.41$ and a bootstrapped (1000 boots) p-value = 0. The full EDFIT Table 8.9 finds all the $(\alpha\kappa)_{2j}$ and $(\alpha\kappa)_{4j}$ interactions significant at the 0.05 level for $j = 1, \ldots, 4$, as well as α_2 and α_4. A log-linear model fit returns with residual deviance 26.45 on 784 degrees of freedom and an estimate $\hat{\theta} = 0.0337$. Table 8.10 shows the point estimates and their standard errors. The α_i main effects are not significant so this fitted model rejects a hypothesis that sequencing algorithm may increase runway capacity whatever the wake-vortex traffic mix. However, the significant $(\alpha\kappa)_{2j}$ and $(\alpha\kappa)_{4j}$ interactions for $j = 2, \ldots, 4$ point to significant improvements through use of the two smart algorithms that vary in size according to the traffic mix.

Findings The possibility of choice has a large effect on how well a sequencing algorithm may perform. In this experiment there was sufficient choice to enable increase in runway capacity. However, this came at a price. The delay the AMAN system gave to aircraft averaged at 34.8 minutes. This is very large when considering Figure 8.5, the distribution of time aircraft spend in the simulated AMAN airspace if there is no delay. This experiment showed that any increase in runway capacity depends on the wake-vortex characteristics of arrival traffic. Even with airspace completely saturated, no statistically significant improvements in location of landing number were found when the wake-vortex mix was set as autumn 2003 at Stockholm Arlanda.

Table 8.10: Airport runway capacity experiment: Landing rate, Question B, log-linear model coefficients with $\hat{\sigma} = 0.033715$

Coefficient	Point Estimate	Std. Error
μ	3.9616	0.0036
α_2	0.0068	0.0051
α_3	0.0011	0.0051
α_4	-0.0046	0.0051
κ_2	-0.1110	0.0052
κ_3	-0.2004	0.0053
κ_4	-0.2544	0.0054
$(\alpha\kappa)_{22}$	0.0626	0.0073
$(\alpha\kappa)_{32}$	-0.0007	0.0074
$(\alpha\kappa)_{42}$	0.0231	0.0074
$(\alpha\kappa)_{23}$	0.1102	0.0074
$(\alpha\kappa)_{33}$	-0.0002	0.0075
$(\alpha\kappa)_{43}$	0.0393	0.0075
$(\alpha\kappa)_{24}$	0.1276	0.0075
$(\alpha\kappa)_{34}$	0.0003	0.0077
$(\alpha\kappa)_{44}$	0.0525	0.0076

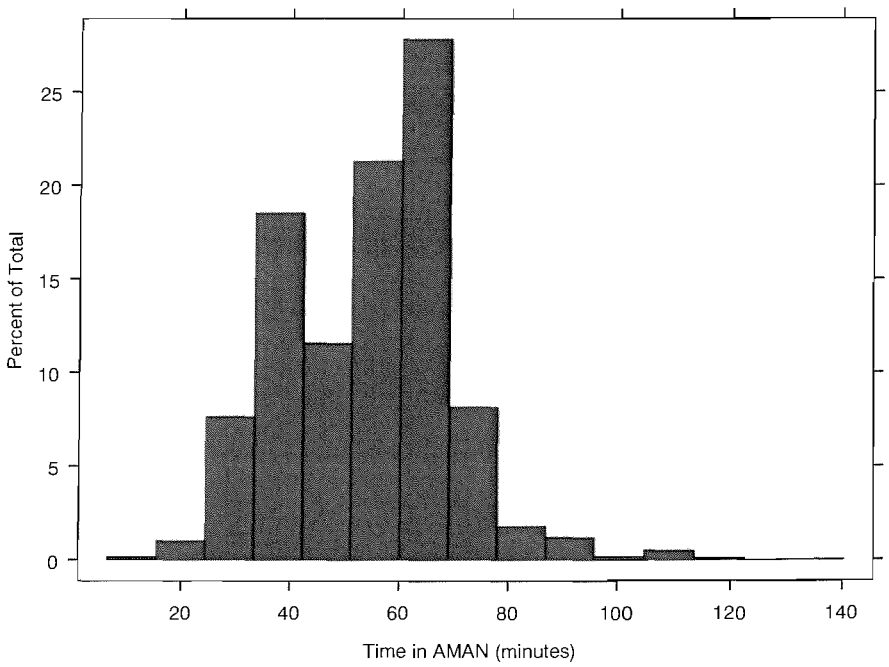


Figure 8.5: Histogram of flight plan time in simulation airspace

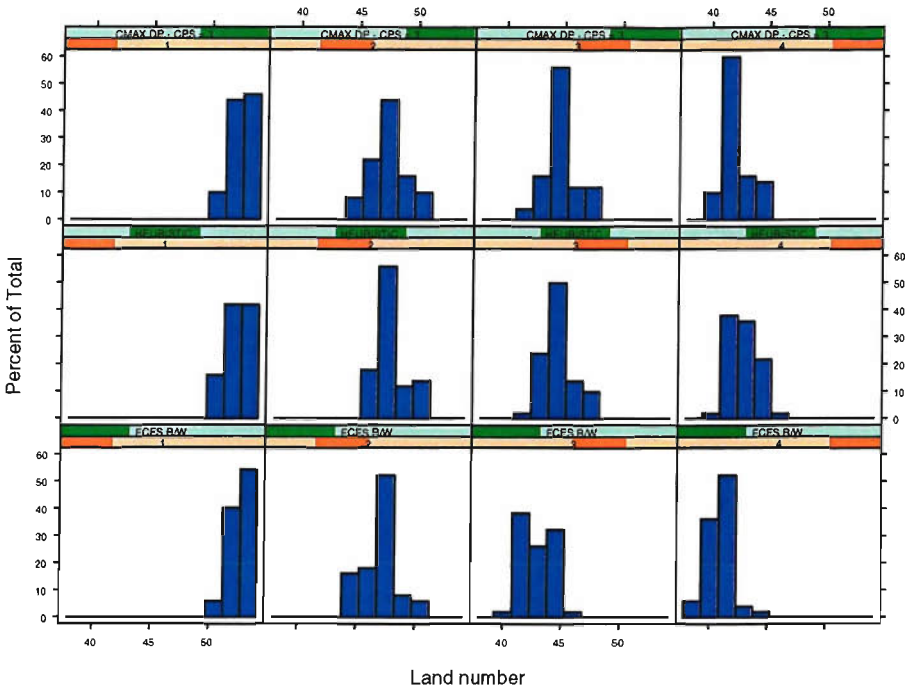


Figure 8.6: Airport runway capacity experiment: Landing rate, Question C, histogram of land number

8.2.5 Question C

What difference will CPS constraints make to potential increases in runway landing capacity?

Experiment parameters Add an additional run to the last experiment with the dynamic program with CPS set to 3 for all aircraft types.

Analysis Visual inspection of output histograms in Figure 8.6 does not show a clear difference between FCFS at runway and the dynamic program with CPS constraints. This conclusion is also drawn from EDFIT tests and log-linear model building: when FCFS at runway is set as a base, algorithm and wake-vortex interactions involving the dynamic program with CPS return insignificant. Table 8.11 illustrates the non-significant difference in overall mean landing rates. This table shows the results from a bootstrap experiment with 10,000 repeats. Landing rates were resampled by traffic sample and algorithm. Although the landing rate is different in means, the difference is not statistically significant pairwise at 90% (using the Bonferroni inequality). However, another bootstrap experiment resampling by traffic sample from wake-vortex categories does find significant

Table 8.11: Airport capacity experiment: Landing rate, Question C, mean landing rate, empirical bootstrap confidence intervals

Algorithm	Mean	Bootstrap empirical C.I.*			
		2.5%	5%	95%	97.5%
FCFS	45.87	45.24	45.34	46.41	46.52
Heuristic	46.87	46.34	46.42	47.32	47.41
DP with CPS=3	46.66	46.11	46.20	47.14	47.24

*10,000 Bootstraps

Table 8.12: Airport runway capacity experiment: Landing rate, Question C, mean landing rate by wake vortex level, empirical bootstrap confidence intervals

Algorithm	WV Level	Mean	95% C.I.*		Multiple 95% C.I.*	
			Lower	Upper	Lower	Upper
FCFS	2	47.04	46.64	47.44	46.44	47.62
Heuristic	2	47.90	47.52	48.30	47.36	48.48
DP with CPS=3	2	47.50	47.10	47.92	46.90	48.10
FCFS	3	43.04	42.66	43.42	42.50	43.60
Heuristic	3	44.52	44.16	44.90	44.00	45.08
DP with CPS=3	3	44.68	44.30	45.08	44.12	45.24
FCFS	4	40.80	40.50	41.10	40.36	41.24
Heuristic	4	42.74	42.40	43.06	42.26	43.22
DP with CPS=3	4	42.04	41.72	42.36	41.56	42.52

*14,400 bootstraps

differences in mean landing rate at wake-vortex levels 3 and 4. The differences are shown in Table 8.12. Notice how the improvement found at wake-vortex level 2, when using the dynamic program without CPS constraints, is no longer significant with CPS constraints set to three.

Findings When airspace is saturated with aircraft and there exists choice of aircraft it is theoretically possible to increase runway capacity over FCFS at runway through a deterministic optimal dynamic program. However, when a CPS constraint is added to this algorithm with a view to make the generated sequence implementable, the limited sequence choice may result in no significant difference in landing rate. Improvements in landing rate depend on the wake-vortex mix of arrival traffic.

8.2.6 Question D

The landing sequences produced by the sequencers are different. Are there any other observable differences in landing rate?

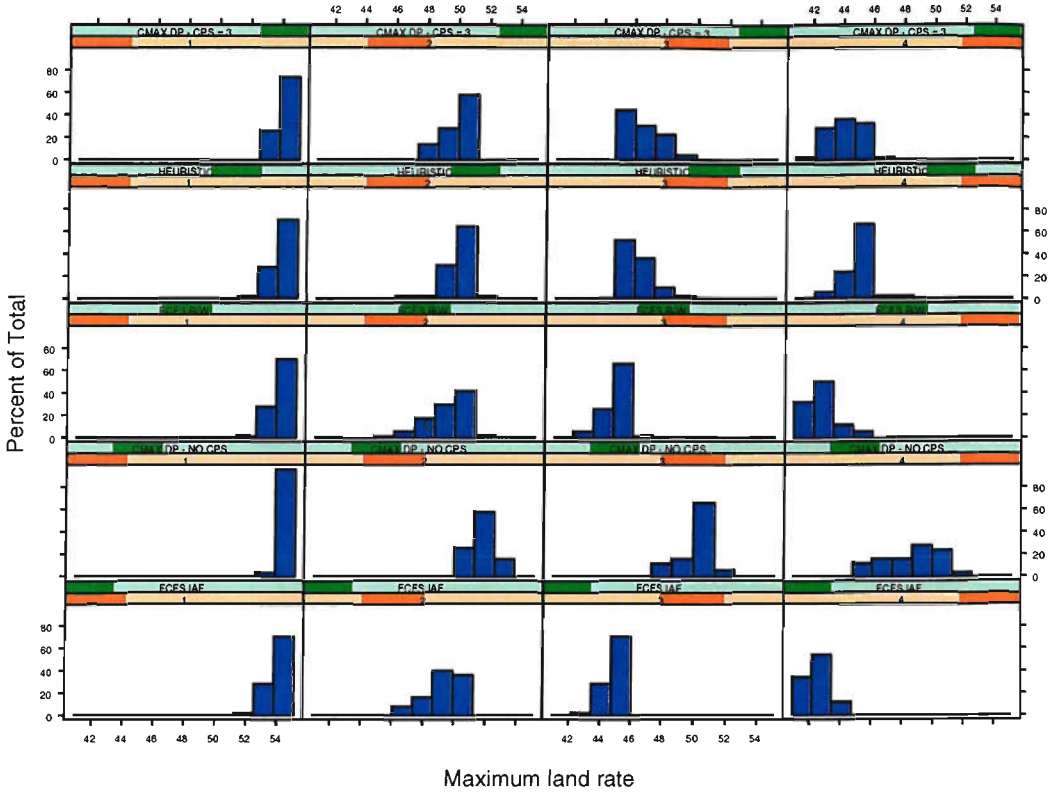


Figure 8.7: Airport runway capacity experiment: Landing rate, Question D, histogram of maximum land rate

Experiment parameters Analyze the previous experiment looking at maximum moving average landing rate. A significant difference for this indicator would show improvement in localized landing rate is possible.

Analysis Figure 8.7 shows histograms of the maximum moving average landing rate. The dynamic program with CPS constraints increases the maximum landing rate over FCFS at runway, depending on the wake-vortex mix. The full EDFIT Table 8.13 shows how this depends on wake-vortex mix: algorithm wake-vortex interactions are significant, not main effects. However, algorithm 5 and wake-vortex interactions are statistically significant.

Findings There is some evidence of the expected gains in clever sequencing over FCFS. Locally, at certain periods of time, clever sequencing produces tighter landing sequences than FCFS.

Table 8.13: Airport runway capacity experiment: Landing rate, Question D, maximum landing rate, full EDFIT table, 1000 bootstraps

	T	C_1	C_2	C_3	C_R
α_2	8.76 (.239)	0.58 (.694)	1.7 (.221)	2.22 (.026)	**4.25 (.000)**
α_3	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)
α_4	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)
α_5	0.22 (1)	0.02 (.946)	0.05 (.825)	0.06 (.675)	0.09 (1)
κ_2	**269.26 (.000)**	**138.46 (.000)**	**111.54 (.000)**	**4.03 (.004)**	**15.23 (.000)**
κ_3	**584.41 (.000)**	**553.81 (.000)**	**9.68 (.000)**	0.25 (.435)	**20.66 (.000)**
κ_4	**757.28 (.000)**	**708 (.000)**	4 (.054)	**25.7 (.000)**	**19.58 (.000)**
$(\alpha\kappa)_{22}$	**143.16 (.000)**	**52.95 (.003)**	**12.51 (.014)**	**22.32 (.000)**	**55.39 (.000)**
$(\alpha\kappa)_{32}$	0.69 (1)	0.07 (.928)	0.04 (.897)	0.05 (.791)	0.54 (.983)
$(\alpha\kappa)_{42}$	14.38 (.301)	7.1 (.328)	0.52 (.609)	3.78 (.039)	2.99 (.187)
$(\alpha\kappa)_{52}$	7.53 (.616)	3.06 (.548)	0.07 (.842)	2.56 (.105)	1.84 (.535)
$(\alpha\kappa)_{23}$	**296.36 (.000)**	**184.69 (.000)**	**64.25 (.000)**	**34.56 (.000)**	**12.87 (.000)**
$(\alpha\kappa)_{33}$	1.05 (.998)	0.07 (0.93)	0.08 (.816)	0 (.961)	0.9 (.916)
$(\alpha\kappa)_{43}$	**57.25 (.015)**	16.07 (.173)	**22.75 (.001)**	**4.54 (0.02)**	**13.89 (.000)**
$(\alpha\kappa)_{53}$	**65.92 (.012)**	19.91 (0.11)	**29.17 (.001)**	2.54 (.068)	**14.3 (.000)**
$(\alpha\kappa)_{24}$	**343.47 (.000)**	**153.87 (.000)**	**167.43 (.000)**	**7.29 (.005)**	**14.88 (.000)**
$(\alpha\kappa)_{34}$	0.58 (1)	0.04 (.947)	0.11 (.818)	0.15 (.655)	0.29 (1)
$(\alpha\kappa)_{44}$	**97.69 (.000)**	9.41 (.302)	**25.04 (.000)**	**28.5 (.000)**	**34.75 (.000)**
$(\alpha\kappa)_{54}$	40.31 (.022)	2.66 (.536)	10.16 (.024)	**9.18 (.000)**	**18.31 (.000)**

8.2.7 General conclusions

The following conclusions are drawn based on the simulation model of aircraft flying into Stockholm Arlanda. They could be applicable for airports with similar arrival airspace characteristics.

- No statistically significant increase in landing rate is found from improved sequencing when arrival levels are around maximum runway capacity for a wide range of wake-vortex traffic mix conditions.
- Airspace needs to be saturated with a sufficient wake-vortex mix of aircraft so there exists enough choice of aircraft for sequencing algorithms before increase in landing rate in peak periods is possible.
- Algorithmic CPS constraints to make the landing sequence workable may make any increase in runway landing rate over FCFS impossible.
- The dynamic environment may lead "optimal" algorithms to make sub-optimal landing sequences.
- Locally, at certain periods of time, using CPS constraints may produce tighter landing sequences than FCFS. Difference in hourly rates is not as marked.

8.3 Conclusions

This chapter has investigated how Arrival Manager tools might impact on airport runway capacity through experiments on a computer simulation model of airspace around Stockholm Arlanda airport. Conclusions are based on data used to run experiments on the model. The effect of an Arrival Manager on airport capacity was examined by looking at delay and landing rate indicators.

No clear difference in mean delay was found between sequencing aircraft FCFS and using the alternative methods under the traffic conditions simulated. However, the distribution of individual aircraft delay was found to change. The heuristic algorithm increased the probability that aircraft are delayed above a threshold of 15 minutes over all other sequencing methods, where the threshold was used in other algorithms as a deadline constraint. It was also found that if the threshold is broken, a smaller proportion of aircraft will miss it when sequenced by the heuristic than other algorithms. Further, dynamic programs with constraints on the FCFS order to IAF points delayed a higher proportion of aircraft 2 minutes under a 15 minutes threshold than all other algorithms.

No significant improvement in landing rate was found throughout the experiments in the previous chapter. Work here found that this is not only due to wake-vortex traffic mix. Traffic needs to be saturated with sufficient wake-vortex mix to a point where algorithms have enough choice of sequence position to make a difference. This did not happen when arrival rates were around airport runway capacity. The dynamic nature of sequence updating was shown to produce situations where an optimal deterministic algorithm may produce sub-optimal sequences, and be bettered by a heuristic. In situations when it is possible to increase landing rate over FCFS, the addition of the CPS constraint that an aircraft may only be sequenced a maximum of 3 positions either side of its FCFS position, resulted in no improvement at certain wake-vortex levels. This was again due to lack of choice. However, running with the CPS constraint bunched aircraft together locally and so increased maximum landing rate over FCFS.

Overall, no clear increase in the airport runway capacity of the simulation model airspace surrounding Stockholm Arlanda was found. System behaviour would change if landing sequence is altered, but there was not enough evidence to support a claim that the simulation airspace model may better cope with more arrivals than it would when sequencing aircraft FCFS. These results may be applicable to airports with a similar arrival airspace.

Capacity is not the only factor considered by Air Traffic Control bodies when investigating change to controller work patterns. Indeed, hypothesized capacity increases are not the only reason for using AMAN technology. This chapter has demonstrated that it may be more advisable to focus on other aspects than airport runway capacity when deciding upon an AMAN strategy. For instance, human factors texts such as Hopkins (1995) argue that for new decision support technologies to be successful, controllers must be able to interrogate the system to better understand why the advice has been made. Sequencing FCFS is an easy to understand rule, complicated objective function based optimization routines may be less so. Other considerations may include some of the system behaviour characteristics investigated in the previous chapter, or the difference in individual aircraft delays found in this chapter. In any case, the behaviour of the system will depend on the particular airspace and so investigation needs to be tailored to the airspace in question.

Chapter 9

Conclusion

9.1 Summary

Landing aircraft must respect minimum separation distances based on their weight, so some landing sequences produce more delay than others. This arrival aircraft sequencing problem gives rise to a related problem, termed the delay-sharing problem, of how to assign delay that results from the landing sequence to individual aircraft. The scheduling problem has been well studied, and several optimization sequencing algorithms have been developed. These algorithms contrast with the majority of developed ATC systems, where arrivals are sequenced using a projected FCFS arrival sequence. Very little rigorous work has been carried out to assess how advanced sequencing algorithms might perform, in conjunction with different delay-sharing strategies, in the real, variable and dynamic world. This work has investigated the performance of different sequencing algorithms and delay-sharing strategies in such an environment, through use of a simulation model of arrival airspace. The effects on Air Traffic Control performance indicators delay, landing rate, holding time, time in approach sectors and the stability of the delay-to-lose advice have been investigated. A caveat applies to all the findings presented: they are based on data used in the modelling process, not on a detailed operational study.

The process of formulating and reviewing the sequencing and delay-sharing problems identified a number of algorithms and problem constraints that had not been previously made. These gaps were exploited in several polynomial-time dynamic programming algorithms, proposed for determining optimal landing sequences for three different machine job scheduling models. In the first, aircraft were assumed to be sequenced out of holds onto several runways for any regular objective function. In the second, aircraft were sequenced onto a single runway based on their release date, to minimize makespan and total tardiness (assuming

each job's release and due date were the same). In the final model, aircraft were sequenced based on their approach stream FCFS order and release dates onto several runways for any regular objective function. Modifications to the formulations to allow deadline and Constrained Position Shifting (CPS) constraints were also presented.

On the basis of discussions with Eurocontrol personnel a discrete-event simulation model of airspace surrounding Stockholm Arlanda airport was built in Visual Basic. The simulation model is viewed as an analysis tool to investigate scheduling and delay-sharing strategies when landing aircraft at airports. Previous work has focused on the effects in the TMA area, but the area of airspace considered by the simulation model is two sectors back from the TMA. The simulation has been built to enough level of detail to enable some performance indicators on delay, landing rate, efficiency and controller workload to be extracted for investigation through experimentation.

Statistical validation procedures have been used to lend credibility to the model results. Statistical input routines, and delay and landing rate output performance indicators from the model were validated against real radar track data, recorded in autumn 2003, using hypothesis tests, confidence intervals and tests for dynamic behaviour. None of the tests lead to the conclusion that inference on the real world cannot be made from experimentation on the model for the specific issues to be addressed.

Experiment analysis methods that make use of a large number of replicates at each design point were reviewed and used. These included (1) the Empirical Distribution Integral Test (EDFIT) method (Cheng & Jones 2004), (2) Monte-Carlo simulation, (3) linear regression models for mean, (4) estimation through weighted least squares with weights estimated from simulation output, (5) variance models fitted using design point sample means, (6) models for skewness, (7) resampling routines based on the large number of design point repeats to test for significance when standard asymptotic result assumptions did not hold, and (8) logit and beta models for the proportion of data points that miss thresholds or targets.

The first series of computer simulation experiments were made to investigate the effects of change to landing sequence algorithm and delay-share strategy on the simulation model of aircraft arrivals into Stockholm Arlanda airport. These changes were looked at in conjunction with potential movements in arrival rates, local and international mix of the traffic, and wake-vortex category mix of arrival

traffic. The results were particular to the model, but may also be applicable to other arrival airspace with similar characteristics. Very little benefit from improved sequencing algorithms was found; sequencing FCFS at runway performed as well as any other. However, system behaviour was found to vary by sequencing algorithm and delay-sharing strategy. Holding time and its variability reduced by delaying aircraft before the TMA. As traffic intensity increased the gain in reducing the mean time holding increased, but the gain in reduction in variability of hold time decreased. Delaying aircraft pre-hold resulted in more traffic for controllers in sectors further back from the airport, even when delaying as late as possible without holding. This may have implications on other Air Traffic Control issues, such as slot allocation for aircraft departing at different airports that need to fly in the sectors affected. Delay-to-lose advice through time was found to be more stable when delaying aircraft in holds, than earlier in airspace. The CPS constraint was shown to be a good method to limit the variability of advice from advanced sequencing algorithms, to that found from a FCFS at runway algorithm. Three general conclusions were drawn from the experiment results. First, improved sequencing techniques should not be regarded as a panacea to reduce delay and increase landing rate because the ability to realise these benefits depends on arrival airspace and traffic characteristics. Second, different sequencing algorithm and delay-share strategies in an AMAN system may cause different system behaviour. Last, choice of sequencing algorithm and delay-share strategy will affect stability of advice to controllers, and quality of information to other users.

The second series of experiments were run to investigate how Arrival Manager tools might impact on airport runway capacity. The effect of an Arrival Manager on airport capacity was examined by looking at delay and landing rate indicators. No clear difference in mean delay was found between sequencing aircraft FCFS and using the alternative methods under the traffic conditions simulated. However, the distribution of individual aircraft delay was found to change. Traffic needed to be saturated with sufficient wake-vortex mix to a point where algorithms had enough choice of sequence position to make a difference. This did not happen when mean arrival rates were around airport runway capacity. The dynamic nature of sequence updating was shown to produce situations where an optimal deterministic algorithm may produce sub-optimal sequences, and be bettered by a heuristic. In situations when it was possible to increase landing rate over FCFS, the addition of the CPS constraint that an aircraft may only be sequenced a maximum of 3 positions either side of its FCFS position, made improvement impossible for some wake-vortex traffic mix levels. However, running with the CPS constraint bunched aircraft together locally and so increased

maximum landing rate over FCFS. Overall, no clear increase in the airport capacity of the simulation model airspace surrounding Stockholm Arlanda was found. System behaviour would change if landing sequence is altered, but there was not enough evidence to support a claim that the simulation airspace model may better cope with more arrivals than it would when sequencing aircraft FCFS.

9.2 Main contributions of thesis

This thesis has made a number of contributions to the literature on the Aircraft Arrival Management problem.

Sequencing New dynamic program sequencing algorithms have been developed for differing models of the sequencing problem.

Scope Previous work has looked at the problem of sequencing aircraft close to the runway, in the TMA. This thesis has examined the problem for real-world operational AMAN systems such as MAESTRO, where advice is made to controllers much further back from the TMA.

Simulation model A computer simulation model has been developed and used to examine the dynamic implementation of different sequencing techniques. The model has been applied to a specific airspace, taking account of the variability in aircraft arrivals. The validity of the model has been tested.

Experimentation Interactions between sequencing algorithm, delay-share strategies and arrival traffic mix have been examined through experimentation.

Statistical methodology A number of statistical methods have been used to analyse output from the simulation model. These have been developed, or selected, for output with high design point replication.

Conclusions The thesis has argued that improved sequencing techniques should not be regarded as a panacea to reduce delay and increase landing rate because the ability to realise these benefits depends on arrival airspace and traffic characteristics. Different sequencing algorithm and delay-share strategies in an AMAN system may cause different system behaviour, and choice of sequencing algorithm and delay-share strategy will affect stability of advice for controllers, and quality of information for other users.

Interpretation The conclusions drawn from experimentation on the simulation model differ from many previously presented. This highlights a need to test

optimization algorithms back on the problem, where algorithmic assumptions are dropped.

9.3 Strengths and Weaknesses

The approach taken to investigate how the ATC system reacts to changes in sequencing algorithms, optimization criteria and delay-sharing strategy has a number of strengths.

Effects on real airspace The results presented are based on analysis of arrivals to a real airspace.

Type of airspace / airport The model of pre-TMA airspace, where there is a concept of k points that aircraft fly through before they begin descent onto the runway, is generally quite valid. It describes the situation at Stockholm Arlanda where there are four IAF points, and London Heathrow where there are four holding points.

Credibility Statistical validation methods have lent credibility to the model output.

Wide range of sequencing methods A variety of different sequencing algorithms have been examined. This increases the probability that effects due to sequencing algorithms have been detected.

Confidence in results Appropriate design of experiment ideas have been used to take variation into account when looking for significant differences due to algorithms, algorithmic constraints delay-share strategies and traffic conditions on pre-TMA airspace. Conclusions are drawn based on analysis using a number of statistical methods.

Some important caveats apply to the work.

Not an operational study Conclusions drawn from the model are based on data used in the modelling process, not on a detailed operational study.

Airspace Only a single arrival airspace was considered.

Airport Only a single arrivals-only runway was considered.

Arrival Traffic Since the purpose of the work was to look at Arrival Management, departures were not considered. So results do not hold for sequencing onto mixed-mode runways.

Algorithms Although a wide-range of sequencing algorithm and delay-share strategies were compared, it may be that other algorithms perform differently.

9.4 Further work

Two avenues of research might take this thesis forward. Firstly, the developed tool might be used to address some of the weaknesses identified above, or for related ATC problems such as departure management. This may involve data collection and additional code development. Secondly, the findings presented might be used to inform on additional analysis to compare AMAN systems.

Address weaknesses Arrival Management is an environment-specific problem, and the effects of different systems may depend on the traffic mix, arrival profiles and airspace. The developed simulation model may be used to examine how different airports and airspace reacts to changes in sequencing algorithms, optimization criteria and delay-sharing strategy. To carry out further analysis on different airports and airspace, the model would need to be set-up with data from the Eurocontrol Central Flow Management Unit (CFMU), and a further data source used for validation. The effect of additional sequencing algorithms or delay-share strategies may be examined by coding them into the model. The tool might also be used to look at problems related to the Arrival Management problem, such as the departure management problem of sequencing aircraft take-offs. This would require change to the structure of the simulation code.

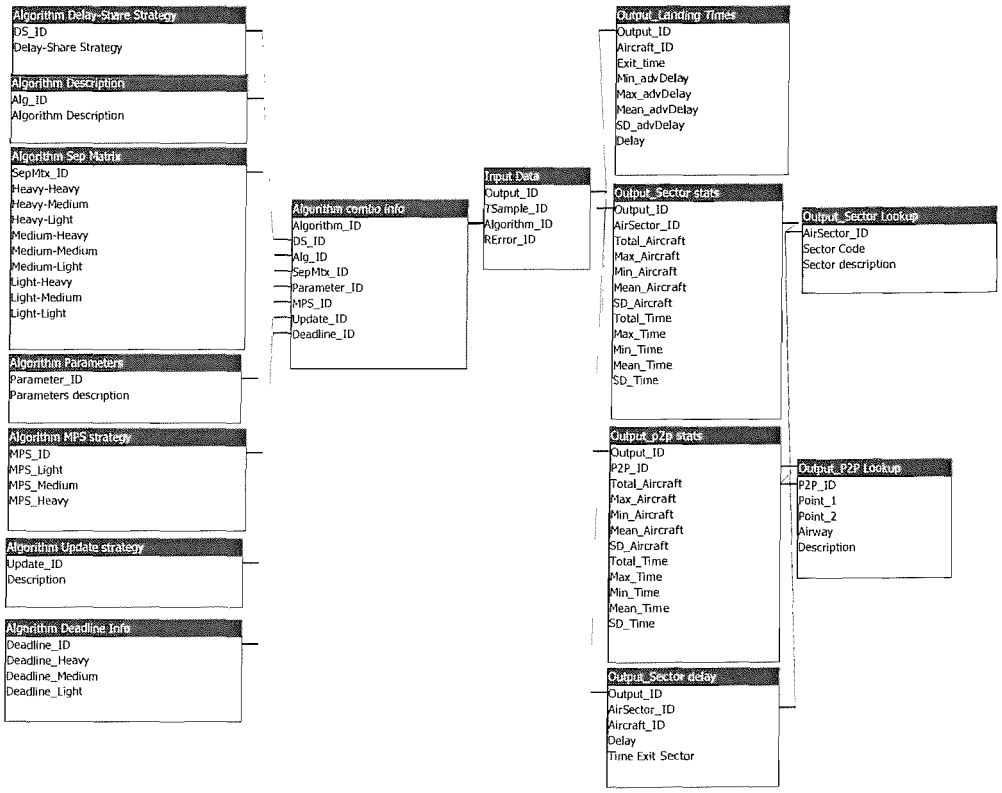
Alternative analysis Little benefit to delay or landing rate was found from improved sequencing of arrivals into Stockholm Arlanda airport. However, changes were found in system behaviour performance indicators. Further analysis might take the thesis forward by looking for other differences in system behaviour between arrival sequencing methods. For example, airline operators may hope for accurate forecast land times to help plan operations. What system might ensure forecast land times be made more accurate? Controllers might have preferences on how aircraft move through their sectors. What system ensures that aircraft arrive in a steady stream? Questions might relate to the assumptions made in Section 4.1.4. For instance, what happens if there is a time lag between controllers being given advice and when they take action? What happens if aircraft do not follow the advice they are given exactly - how accurate do they need to be? The simulation model may be used to examine these questions, and others about the ATC system.

The package of work to address weaknesses should help increase confidence in the general conclusions drawn in this thesis. The second type of further work would require experts to ask the questions, but could, potentially, provide powerful guidance.

Appendix

Appendix A: Chapter 4

Output database relationships

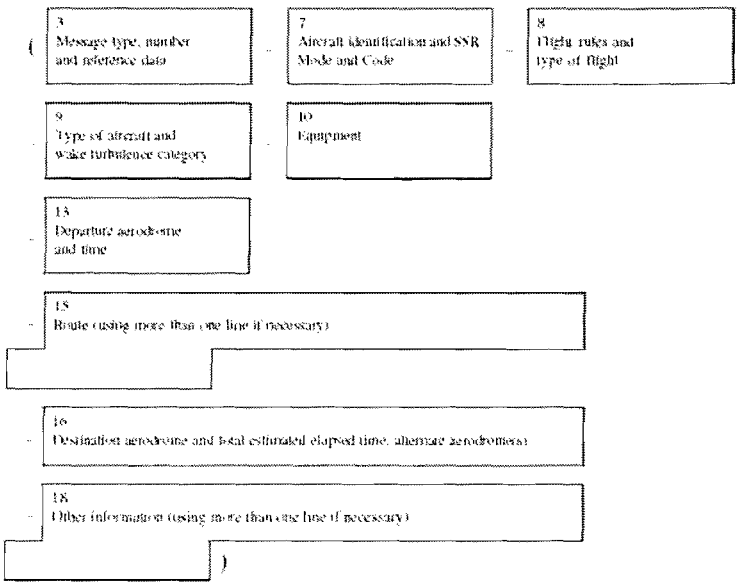


Appendix B: Chapter 5

ICAO FPL data

2.3.1 *Filed flight plan (FPL) message*

2.3.1.1 *Composition*



2.3.1.2 *Example*

The following is an example of a filed flight plan message sent by London Airport to Shannon, Shanwick and Gander Centres. The message may also be sent to the London Centre or the data may be passed to that centre by voice.

(FPL-TPR101-IS
-B707M-GHOPV/C1
EGLL1400
-N0450F310 G1 UG1 STU2N50W6M082F310 UG1 52N015W
52N020W 52N030W 50N040W 49N050W
-CYQX0455 CYYR
-ELT/EINN0426 EGGX0111 20W0136 CYQX0228 40W0330 50W0415 SEL/EJLI)

ICAO FPL data description

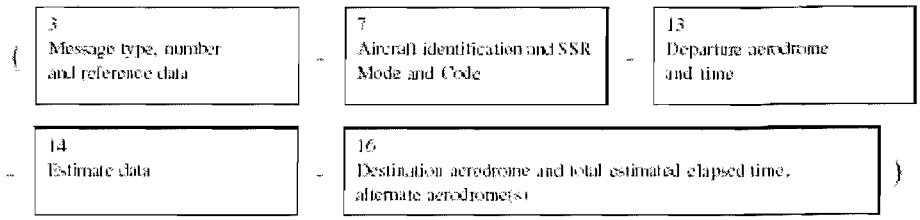
2.3.1.2.1 *Meaning*

Filed flight plan message — aircraft identification TPR101 — IFR, scheduled flight — a Boeing 707, medium wake turbulence category equipped with Loran C, HF RTF, VOR, Doppler, VHF RTF and SSR transponder with Modes A (4 096 code capability) and C — ADS capability — departure aerodrome is London, estimated off-block time 1400 UTC — cruising speed and requested flight level for the first portion of the route are 450 knots and FL 310 — the flight will proceed on Airways Green 1 and Upper Green 1 to a point bearing 285 degrees magnetic and 36 NM from the Strumble VOR. From this point the flight will fly at a constant Mach number of .82, proceeding on Upper Green 1 to 52N15W; then to 52N20W; to 52N30W; to 50N40W; to 49N50W; to destination Gander, total estimated elapsed time 4 hours and 55 minutes — alternate is Goose Bay — captain has notified accumulated estimated elapsed times at significant points along the route, they are at the Shannon FIR boundary 26 minutes, at the Shanwick Oceanic FIR boundary 1 hour and 11 minutes, at 20W 1 hour and 36 minutes, at the Gander Oceanic FIR boundary 2 hours and 28 minutes, at 40W 3 hours and 30 minutes and at 50W 4 hours and 15 minutes — SELCAL code is EJLI.

ICAO EST data description

2.4.2 *Estimate (EST) message*

2.4.2.1 *Composition*



2.4.2.2 *Example*

The following is an example of an estimate message sent from Paris Centre to London Centre. It is assumed that London Centre has received a filed flight plan message relating to this flight. Both centres are equipped with computers.

(ESTFMI027-BAW671/A5631-LFPG-ABB/1548F140T110A-EGLL)

2.4.2.2.1 *Meaning*

Estimate message [with sending unit identity (F) and receiving unit identity (L), followed by the serial number of this message (027)] — aircraft identification BAW671, last assigned SSR Code 5631 operating in Mode A — departure aerodrome Paris de Gaulle — estimating Abbeville VOR 1548 UTC, cleared FL 140, flight will cross the Abbeville VOR at FL 110 or above, ascending — destination aerodrome London.

Track data fields

TRK data content (simple radar tracks) : 1. Time (hours : minutes : seconds : milliseconds) 2. Area ID (integer) 3. Station ID (integer) 4. Track number (integer) 5. Latitude (float in degrees) 6. Longitude (float in degrees) 7. Altitude (float in feet) 8. Valid mode C (0—1) 9. Flight level (integer) 10. Track angle (float in degrees) 11. Ground speed (float in knots : NM — hour) 12. Rate Of Climb (float in feet per minute) 13. Attitude indicator : 0 (Levelled off) — 1 (Descending) — 2 (Climbing) — 3 (Unknown) 14. Valid SSR code (0—1) 15. SSR code (4 digits) 16. Track status - Simulated flag (0—1) 17. Track status - Manoeuvring flag (0—1) 18. Track status - End of track (0—1) 19. Track status - Special Position Indicator flag (0—1) 20. Track status - Update kind : 0 (Extrapolated) — 1 (Only PSR) — 2 (Only SSR) — 3 (PSR And SSR) — 4 (ADS-B) 21. Track origin : 0 (Undefined) — 1 (Radar) — 2 (ADS) — 3 (Combined) — 4 (Fpl Interpolation) 22. Operational Flight Plan ID (integer) 23. Callsign (8 characters or empty)
Additional ADS data (ADS-B information) :
24. Barometric altitude (float in feet) 25. FOM : position accuracy (0 to 10) 26. FOM : ACAS Operational flag (0—1) 27. FOM : Multiple Navigation Aids Operating flag (0—1) 28. FOM : Differential Correction flag (0—1) 29. Velocity accuracy (0 to 10) 30. Number of points in the projected profile (0 to 4) 31. Top of Descent position in the projected profile (0 to 4) 32. 1st projected point : altitude (float in feet) 33. 1st projected point : latitude (float in degrees) 34. 1st projected point : longitude (float in degrees) 35. 1st projected point : time to go (float in seconds) 36. 2nd projected point : altitude (float in feet) 37. 2nd projected point : latitude (float in degrees) 38. 2nd projected point : longitude (float in degrees) 39. 2nd projected point : time to go (float in seconds) 40. 3rd projected point : altitude (float in feet) 41. 3rd projected point : latitude (float in degrees) 42. 3rd projected point : longitude (float

in degrees) 43. 3rd projected point : time to go (float in seconds) 44. 4th projected point : altitude (float in feet) 45. 4th projected point : latitude (float in degrees) 46. 4th projected point : longitude (float in degrees) 47. 4th projected point : time to go (float in seconds) 48. Link technology : 0 (Other) — 1 (Mode S) — 2 (UAT) — 3 (VDL) 49. ADS-B latitude (float in degrees) 50. ADS-B longitude (float in degrees) 51. ADS-B altitude (float in feet) 52. ADS-B flight level (long) 53. ADS-B heading (float in degrees) 54. ADS-B ground speed (float in knots) 55. ADS-B rate of climb (float in feet per minute) 56. Emitter category : 0 (unkonwn) — 1 (light aircraft) — 3 (medium aircraft) — 5 (heavy aircraft) — 10 (rotocraft) — 20 (surface emergency vehicle) — 21 (surface service vehicle) — 22 (fixed ground) 57. Event report reason : 0 (lateral deviation) — 1 (vertical rate change) — 2 (altitude threshold) — 3 (way point change) — 4 (air speed change) — 5 ground speed change) — 6 (heading change) — 7 (projected profile change) — 8 (FOM change) — 9 (track angle change) — 10 (altitude change) — 11 (unknown) 58. Selected flight level (long) 59. Turn indicator : 0 (Left) — 1 (Right) — 2 (Straight) — 3 (unknown) 60. Rate of turn (float in degrees per second) 61. Air Vector : heading (float in degrees) 62. Air Vector : speed (float in knots) 63. Air Vector : rate of climb (float in feet per min) 64. Aircraft type (char [4]) 65. Target status : 0 (no emergency) — 1 (general emergency) — 2 (lifeguard or medical) — 3 (minimum fuel) — 4 (no communications) — 5 (unlawful interference) 66. Aicraft Address (Hexadecimal value as 6 characters) 67. Age of the last SSR plot (float in seconds) 68. Age of the last ADS-B report (float in seconds)

Technical report validation summary

Strengths	**Quantitative** methods to validate. **Statistical tests** **Confidence** intervals on the differences.
Weaknesses	**Hypothesis tests: Data Vs Accuracy** Lot of data ⇒ reject, Little data ⇒ accept. **Data** Confidence in validation is limited by amount of data available. **Dynamic tests** Subjective.

Overall conclusion	Justification	Confidence
AMAN simulation model is valid	Quantitative testing on differences between model and track data for a set of actual data could not find significant differences for model inputs or important outputs. Some sensitivity and dynamic tests were passed.	Probability we would see the hypothesis test scores in this report if there is no difference between model and reality is greater than 0.05.

Category	Techniques	Justification	Reference	Conclusions	Confidence
Inputs ◇ Arrival rates ◇ IAF ◇ Wake--vortex category	i. Hypothesis testing ii.Goodness-of-fit tests iii. Bootstrapping	To see if there is enough Track data to suggest that the sampling procedure is invalid. ⇒ Testing hypothesis that procedures in place are valid using GOF and bootstrapping.	Chapter 5 Section 5.2	1. A non-stationary Poisson process generates an arrival sequence at IAF's 2. The probability of aircraft flying to IAF points during a day follows a multinomial distribution 3. The probability of aircraft of different wake vortex category flying to an IAF follows a multinomial model.	0. Data: 16 traffic samples of variable length (1hr 5m - 3h 37m) 1. Test scores using the data each had probabilities > 0.05 that the the variation in test score would be seen if the model is correct. 2. Test score using the data had a probability $= 0.90$ that the the variation in test score would be seen if the model is correct. 3. Test scores using the data each had probabilities > 0.15 that the the variation in test score would be seen if the model is correct.
Outputs ◇ Delay ◇ Landrate	i. Hypothesis testing ii. Goodness-of-fit tests iii. Bootstrapping iv. EDF charts v. Boxplots vi. t-tests vii. Regression viii. Autocorrelation ix. Time series	Used techniques to investigate: ▷ Confidence in output ▷ Sensitivity analysis ▷ Dynamic behaviour	Chapter 5 Section 5.3	1. Not enough evidence to support hypothesis of difference in delay 2. Not enough evidence to support hypothesis of difference in landing rate. 3. Landing rate in model and the Track data follow the same trend in time 4. Landing rate in model and track have similar time series characteristics.	1a. Test scores using the data each have a probability ≥ 0.0845 that the the variation in test scores would be seen if delays are equivalent. 1b. Test for difference in mean delay has a probability $= 0.368$ that the the variation in test scores would be seen if mean delays are equivalent. 1c. 95% Confidence interval on difference in means is [-0.159, 0.319]. 2. Test scores using the data are not rejected at the 95% level. 3. Model mean and Track landing rates are positively correlated. A linear regression estimate of coefficient of model mean to predict actual landing rate is 0.765 with p-val 0.001.

Appendix C: Chapter 7

Experiment III: Parameters used for wake-vortex levels and IAF routes

l	Category	IAF 1 (%)	IAF 2 (%)	IAF 3 (%)	IAF 4 (%)
2	Route	20.11	22.26	47.33	10.31
2	H	3.10	0.00	1.59	4.75
2	M	96.90	99.68	97.16	95.25
2	L	0.00	0.32	1.25	0.00
3	Route	33.01	15.79	34.28	16.92
3	H	3.10	0.00	3.61	4.75
3	M	96.90	99.68	95.17	95.25
3	L	0.00	0.32	1.22	0.00
4	Route	39.46	12.56	27.75	20.23
4	H	3.10	0.00	5.33	4.75
4	M	96.90	99.68	93.47	95.25
4	L	0.00	0.32	1.20	0.00

Experiment IV: Parameters used for wake-vortex levels

m	n	Increase Percentage	WV category	IAF 1 (%)	IAF 2 (%)	IAF 3 (%)	IAF 4 (%)
2	3	3	H	6.10	0.00	5.44	8.75
2	3	-4	M	92.90	95.68	92.33	91.25
2	3	1	L	1.00	4.32	2.24	0.00
2	1	2	H	5.10	0.00	4.44	8.75
2	1	-4	M	92.90	95.68	92.33	91.25
2	1	2	L	2.00	4.32	3.24	0.00
2	2	1	H	4.10	0.00	3.44	8.75
2	2	-4	M	92.90	95.68	92.33	91.25
2	2	3	L	3.00	4.32	4.24	0.00
3	3	5.25	H	8.35	0.00	7.69	11.75
3	3	-7	M	89.90	92.68	89.33	88.25
3	3	1.75	L	1.75	7.32	2.99	0.00
3	1	3.5	H	6.60	0.00	5.94	11.75
3	1	-7	M	89.90	92.68	89.33	88.25
3	1	3.5	L	3.50	7.32	4.74	0.00
3	2	1.75	H	4.85	0.00	4.19	11.75
3	2	-7	M	89.90	92.68	89.33	88.25
3	2	5.25	L	5.25	7.32	6.49	0.00
3	3	7.5	H	10.60	0.00	9.94	14.75
4	3	-10	M	86.90	89.68	86.33	85.25
4	3	2.5	L	2.50	10.32	3.74	0.00
4	1	5	H	8.10	0.00	7.44	14.75
4	1	-10	M	86.90	89.68	86.33	85.25

Experiment IV: Parameters used for wake-vortex levels (continued)

m	n	Increase Percentage	WV category	IAF 1 (%)	IAF 2 (%)	IAF 3 (%)	IAF 4 (%)
4	1	5	L	5.00	10.32	6.24	0.00
4	2	2.5	H	5.60	0.00	4.94	14.75
4	2	-10	M	86.90	89.68	86.33	85.25
4	2	7.5	L	7.50	10.32	8.74	0.00

Experiment IV: Delay, mean model coefficients, 1000 bootstraps

Coefficient	Point Estimate	Percentile point 0.25	0.5	0.95	0.975
μ	2.556	2.475	2.484	2.615	2.630
α_2	-1.410	-1.512	-1.496	-1.319	-1.305
α_3	-1.435	-1.541	-1.521	-1.347	-1.330
α_4	-1.420	-1.520	-1.506	-1.331	-1.318
α_5	0.168	0.062	0.077	0.256	0.266
α_6	0.062	-0.048	-0.032	0.171	0.186
β_2	-0.006	-0.104	-0.086	0.075	0.098
β_3	-0.019	-0.117	-0.104	0.066	0.085
β_4	-0.053	-0.158	-0.141	0.033	0.052
δ_2	0.104	0.020	0.028	0.173	0.185
δ_3	1.274	1.155	1.171	1.362	1.378
δ_4	4.987	4.595	4.646	5.189	5.229
κ_2	0.412	0.297	0.317	0.515	0.533
κ_3	0.867	0.735	0.755	0.975	0.998
κ_4	1.052	0.935	0.951	1.157	1.173
$(\alpha\beta)_{22}$	-0.046	-0.175	-0.156	0.064	0.088
$(\alpha\beta)_{32}$	-0.070	-0.202	-0.181	0.048	0.062
$(\alpha\beta)_{42}$	-0.097	-0.221	-0.204	0.012	0.032
$(\alpha\beta)_{52}$	-1.667	-1.788	-1.771	-1.560	-1.542
$(\alpha\beta)_{23}$	-0.092	-0.221	-0.202	0.022	0.044
$(\alpha\beta)_{33}$	-0.015	-0.150	-0.125	0.098	0.118
$(\alpha\beta)_{43}$	-0.048	-0.182	-0.148	0.062	0.091
$(\alpha\beta)_{53}$	-1.639	-1.770	-1.750	-1.519	-1.503
$(\alpha\beta)_{54}$	-1.648	-1.779	-1.754	-1.526	-1.496
$(\alpha\kappa)_{22}$	-0.252	-0.426	-0.390	-0.112	-0.091
$(\alpha\kappa)_{32}$	-0.259	-0.421	-0.399	-0.116	-0.089
$(\alpha\kappa)_{42}$	-0.290	-0.460	-0.428	-0.152	-0.132
$(\alpha\kappa)_{52}$	-0.220	-0.380	-0.348	-0.089	-0.055
$(\alpha\kappa)_{62}$	-0.039	-0.317	-0.271	0.185	0.228
$(\alpha\kappa)_{23}$	-0.447	-0.619	-0.595	-0.295	-0.264
$(\alpha\kappa)_{33}$	-0.439	-0.619	-0.592	-0.286	-0.260
$(\alpha\kappa)_{43}$	-0.519	-0.702	-0.661	-0.359	-0.328
$(\alpha\kappa)_{53}$	-0.424	-0.601	-0.567	-0.277	-0.247
$(\alpha\kappa)_{63}$	-0.060	-0.361	-0.308	0.202	0.258

Experiment IV: Delay, mean model coefficients, 1000 bootstraps (continued)

Coefficient	Point Estimate	Percentile point 0.25	0.5	0.95	0.975
$(\alpha\kappa)_{24}$	-0.652	-0.821	-0.783	-0.510	-0.487
$(\alpha\kappa)_{34}$	-0.641	-0.809	-0.775	-0.492	-0.473
$(\alpha\kappa)_{44}$	-0.768	-0.933	-0.902	-0.625	-0.599
$(\alpha\kappa)_{54}$	-0.644	-0.785	-0.761	-0.505	-0.485
$(\alpha\kappa)_{64}$	-0.390	-0.631	-0.593	-0.176	-0.130
$(\alpha\delta)_{22}$	0.155	0.026	0.043	0.252	0.271
$(\alpha\delta)_{32}$	0.150	0.024	0.040	0.247	0.262
$(\alpha\delta)_{42}$	0.165	0.046	0.061	0.260	0.276
$(\alpha\delta)_{52}$	0.133	0.012	0.029	0.224	0.244
$(\alpha\delta)_{62}$	0.023	-0.167	-0.134	0.199	0.219
$(\alpha\delta)_{23}$	0.070	-0.104	-0.080	0.201	0.225
$(\alpha\delta)_{33}$	0.063	-0.103	-0.082	0.199	0.219
$(\alpha\delta)_{43}$	0.031	-0.125	-0.098	0.168	0.191
$(\alpha\delta)_{53}$	0.006	-0.158	-0.129	0.125	0.147
$(\alpha\delta)_{63}$	0.139	-0.126	-0.081	0.372	0.413
$(\alpha\delta)_{24}$	0.016	-0.499	-0.415	0.418	0.488
$(\alpha\delta)_{34}$	0.047	-0.415	-0.361	0.427	0.473
$(\alpha\delta)_{44}$	-0.157	-0.622	-0.556	0.233	0.325
$(\alpha\delta)_{54}$	-0.265	-0.677	-0.609	0.074	0.142
$(\alpha\delta)_{64}$	-0.434	-1.009	-0.928	0.129	0.217

Experiment IV: Landing rate, skewness model

Coefficient	Value	SE	t-value	p-value
μ	0.087	0.042	2.057	0.041
$(\phi\delta)_{12}$	0.074	0.029	2.526	0.013
$(\phi\delta)_{13}$	-0.197	0.029	-6.708	0.000
$(\phi\delta)_{14}$	-0.543	0.029	-18.515	0.000
$(\phi\delta)_{02}$	-0.290	0.095	-3.044	0.003
$(\phi\delta)_{03}$	-0.266	0.085	-3.141	0.002
$(\phi\delta)_{04}$	-1.244	0.095	-13.080	0.000
$(\phi\delta)_{22}$	0.398	0.095	4.187	0.000
$(\phi\delta)_{23}$	0.595	0.095	6.252	0.000
$(\phi\delta)_{24}$	-1.119	0.085	-13.204	0.000
$(\phi\delta)_{32}$	0.636	0.095	6.684	0.000
$(\phi\delta)_{33}$	0.490	0.085	5.784	0.000
$(\phi\delta)_{34}$	-1.097	0.095	-11.526	0.000
$(\phi\delta)_{42}$	-0.335	0.085	-3.953	0.000
$(\phi\delta)_{43}$	-0.456	0.095	-4.795	0.000
$(\phi\delta)_{44}$	-1.092	0.095	-11.475	0.000
$(\phi\delta)_{52}$	-0.130	0.095	-1.364	0.175
$(\phi\delta)_{53}$	-0.108	0.085	-1.270	0.206
$(\phi\delta)_{54}$	-1.768	0.095	-18.587	0.000

Experiment IV: Landing rate, skewness model (continued)

Coefficient	Value	SE	t-value	p-value
$(\phi\delta)_{62}$	0.584	0.085	6.891	0.000
$(\phi\delta)_{63}$	0.373	0.095	3.926	0.000
$(\phi\delta)_{64}$	-0.886	0.095	-9.312	0.000
$(\phi\delta)_{72}$	-0.007	0.095	-0.079	0.937
$(\phi\delta)_{73}$	-0.024	0.095	-0.250	0.803
$(\phi\delta)_{74}$	-1.443	0.085	-17.026	0.000
$(\phi\delta)_{82}$	-0.322	0.085	-3.804	0.000
$(\phi\delta)_{83}$	-0.489	0.095	-5.145	0.000
$(\phi\delta)_{84}$	-1.090	0.095	-11.459	0.000
$(\phi\delta)_{92}$	0.313	0.095	3.295	0.001
$(\phi\delta)_{93}$	-0.412	0.095	-4.332	0.000
$(\phi\delta)_{94}$	-1.579	0.085	-18.625	0.000
α_2	-0.006	0.047	-0.130	0.897
α_3	0.036	0.048	0.747	0.456
α_4	0.077	0.047	1.653	0.100
α_5	-0.029	0.045	-0.646	0.519
α_6	-0.032	0.058	-0.548	0.585
β_2	-0.014	0.050	-0.286	0.775
β_3	0.036	0.048	0.757	0.451
β_4	-0.043	0.049	-0.874	0.383
$(\alpha\beta)_{22}$	0.025	0.053	0.470	0.639
$(\alpha\beta)_{32}$	-0.035	0.055	-0.635	0.526
$(\alpha\beta)_{42}$	-0.103	0.053	-1.936	0.055
$(\alpha\beta)_{52}$	0.021	0.054	0.393	0.695
$(\alpha\beta)_{23}$	-0.085	0.053	-1.604	0.111
$(\alpha\beta)_{33}$	-0.086	0.055	-1.564	0.120
$(\alpha\beta)_{43}$	-0.161	0.053	-3.038	0.003
$(\alpha\beta)_{53}$	0.018	0.050	0.348	0.728
$(\alpha\beta)_{54}$	0.055	0.052	1.052	0.294
$(\alpha\kappa)_{12}$	-0.110	0.108	-1.025	0.307
$(\alpha\kappa)_{22}$	-0.073	0.118	-0.615	0.539
$(\alpha\kappa)_{32}$	-0.090	0.118	-0.760	0.448
$(\alpha\kappa)_{42}$	-0.066	0.118	-0.556	0.579
$(\alpha\kappa)_{52}$	-0.107	0.108	-0.989	0.324
$(\alpha\kappa)_{13}$	-0.126	0.108	-1.167	0.245
$(\alpha\kappa)_{23}$	-0.153	0.118	-1.294	0.198
$(\alpha\kappa)_{33}$	-0.155	0.118	-1.316	0.190
$(\alpha\kappa)_{43}$	-0.135	0.118	-1.145	0.254
$(\alpha\kappa)_{53}$	-0.123	0.108	-1.145	0.254
$(\alpha\kappa)_{14}$	0.065	0.108	0.600	0.549
$(\alpha\kappa)_{24}$	-0.004	0.118	-0.034	0.973
$(\alpha\kappa)_{34}$	-0.008	0.118	-0.067	0.947
$(\alpha\kappa)_{44}$	0.008	0.118	0.068	0.946
$(\alpha\kappa)_{54}$	0.042	0.108	0.394	0.694

Experiment IV: Landing rate, skewness model (continued)

Coefficient	Value	SE	t-value	p-value
$(\alpha\lambda)_{12}$	0.113	0.100	1.127	0.262
$(\alpha\lambda)_{22}$	0.087	0.109	0.794	0.428
$(\alpha\lambda)_{32}$	0.077	0.109	0.709	0.480
$(\alpha\lambda)_{42}$	0.074	0.109	0.676	0.500
$(\alpha\lambda)_{52}$	0.076	0.100	0.758	0.450
$(\alpha\lambda)_{13}$	0.274	0.100	2.732	0.007
$(\alpha\lambda)_{23}$	0.388	0.109	3.546	0.001
$(\alpha\lambda)_{33}$	0.354	0.109	3.241	0.001
$(\alpha\lambda)_{43}$	0.346	0.109	3.163	0.002
$(\alpha\lambda)_{53}$	0.341	0.100	3.400	0.001
$(\beta\kappa)_{22}$	-0.001	0.072	-0.018	0.986
$(\beta\kappa)_{32}$	0.014	0.071	0.204	0.839
$(\beta\kappa)_{42}$	0.025	0.072	0.341	0.734
$(\beta\kappa)_{23}$	-0.013	0.072	-0.175	0.861
$(\beta\kappa)_{33}$	-0.061	0.071	-0.865	0.388
$(\beta\kappa)_{43}$	0.011	0.072	0.154	0.878
$(\beta\kappa)_{24}$	-0.058	0.072	-0.801	0.425
$(\beta\kappa)_{34}$	-0.057	0.071	-0.808	0.420
$(\beta\kappa)_{44}$	-0.087	0.072	-1.205	0.230
$(\beta\lambda)_{22}$	-0.063	0.066	-0.951	0.343
$(\beta\lambda)_{32}$	-0.011	0.065	-0.163	0.871
$(\beta\lambda)_{42}$	-0.044	0.066	-0.666	0.506
$(\beta\lambda)_{23}$	-0.040	0.066	-0.607	0.545
$(\beta\lambda)_{33}$	-0.025	0.065	-0.394	0.694
$(\beta\lambda)_{43}$	-0.116	0.066	-1.750	0.082

Experiment IV: Holding time, mean model coefficients, 1000 bootstraps

Coefficient	Point	Percentile point			
	Estimate	0.25	0.5	0.95	0.975
μ	1.264	1.218	1.225	1.301	1.307
α_2	-0.083	-0.101	-0.098	-0.069	-0.067
α_3	-0.087	-0.105	-0.102	-0.073	-0.071
α_4	-0.085	-0.101	-0.099	-0.071	-0.069
α_5	0.069	-0.001	0.011	0.129	0.140
α_6	0.045	-0.014	-0.003	0.098	0.106
β_2	-1.155	-1.206	-1.195	-1.112	-1.104
β_3	-1.119	-1.167	-1.159	-1.078	-1.072
β_4	-1.173	-1.220	-1.211	-1.130	-1.123
δ_2	0.302	0.209	0.227	0.379	0.394
δ_3	1.701	1.557	1.572	1.804	1.831
δ_4	5.594	5.022	5.108	5.939	6.011
κ_2	0.499	0.345	0.376	0.623	0.643
κ_3	0.937	0.772	0.805	1.075	1.107

Experiment IV: Holding time, mean model coefficients, 1000 bootstraps (continued)

Coefficient	Point Estimate	Percentile point 0.25	0.5	0.95	0.975
κ_4	1.217	1.054	1.075	1.347	1.377
$(\alpha\beta)_{22}$	-0.006	-0.035	-0.030	0.019	0.025
$(\alpha\beta)_{32}$	-0.010	-0.038	-0.033	0.015	0.018
$(\alpha\beta)_{42}$	-0.019	-0.050	-0.042	0.005	0.007
$(\alpha\beta)_{52}$	-0.146	-0.226	-0.211	-0.080	-0.066
$(\alpha\beta)_{23}$	-0.015	-0.050	-0.044	0.014	0.020
$(\alpha\beta)_{33}$	0.003	-0.029	-0.022	0.031	0.036
$(\alpha\beta)_{43}$	-0.044	-0.073	-0.069	-0.018	-0.010
$(\alpha\beta)_{53}$	-0.175	-0.256	-0.244	-0.111	-0.096
$(\alpha\beta)_{54}$	-0.143	-0.221	-0.210	-0.080	-0.065
$(\alpha\delta)_{22}$	0.006	-0.033	-0.028	0.034	0.042
$(\alpha\delta)_{32}$	0.002	-0.037	-0.032	0.031	0.039
$(\alpha\delta)_{42}$	0.000	-0.036	-0.032	0.029	0.034
$(\alpha\delta)_{52}$	0.022	-0.022	-0.014	0.056	0.063
$(\alpha\delta)_{62}$	0.093	-0.072	-0.051	0.217	0.243
$(\alpha\delta)_{23}$	-0.222	-0.291	-0.279	-0.170	-0.158
$(\alpha\delta)_{33}$	-0.252	-0.319	-0.306	-0.198	-0.185
$(\alpha\delta)_{43}$	-0.245	-0.318	-0.304	-0.189	-0.179
$(\alpha\delta)_{53}$	-0.062	-0.134	-0.124	0.000	0.010
$(\alpha\delta)_{63}$	0.139	-0.094	-0.050	0.361	0.392
$(\alpha\delta)_{24}$	-0.693	-1.082	-1.017	-0.369	-0.331
$(\alpha\delta)_{34}$	-0.708	-1.087	-1.036	-0.383	-0.348
$(\alpha\delta)_{44}$	-0.669	-1.052	-1.007	-0.357	-0.288
$(\alpha\delta)_{54}$	-0.160	-0.515	-0.452	0.149	0.198
$(\alpha\delta)_{64}$	-0.531	-1.167	-1.051	0.092	0.212
$(\beta\delta)_{22}$	-0.296	-0.390	-0.373	-0.219	-0.200
$(\beta\delta)_{32}$	-0.283	-0.380	-0.363	-0.208	-0.192
$(\beta\delta)_{42}$	-0.300	-0.394	-0.377	-0.224	-0.205
$(\beta\delta)_{23}$	-1.198	-1.332	-1.310	-1.082	-1.066
$(\beta\delta)_{33}$	-1.095	-1.229	-1.207	-0.975	-0.958
$(\beta\delta)_{43}$	-1.290	-1.423	-1.402	-1.170	-1.152
$(\beta\delta)_{24}$	-2.803	-3.327	-3.234	-2.367	-2.274
$(\beta\delta)_{34}$	-2.468	-2.981	-2.894	-2.052	-1.962
$(\beta\delta)_{44}$	-3.002	-3.534	-3.435	-2.572	-2.495
$(\alpha\kappa)_{22}$	-0.128	-0.186	-0.179	-0.066	-0.058
$(\alpha\kappa)_{32}$	-0.136	-0.194	-0.185	-0.081	-0.068
$(\alpha\kappa)_{42}$	-0.146	-0.203	-0.195	-0.091	-0.084
$(\alpha\kappa)_{52}$	-0.089	-0.168	-0.152	-0.025	-0.015
$(\alpha\kappa)_{62}$	-0.104	-0.353	-0.317	0.128	0.181
$(\alpha\kappa)_{23}$	-0.245	-0.315	-0.303	-0.180	-0.166
$(\alpha\kappa)_{33}$	-0.262	-0.331	-0.320	-0.199	-0.190
$(\alpha\kappa)_{43}$	-0.289	-0.356	-0.344	-0.221	-0.213
$(\alpha\kappa)_{53}$	-0.196	-0.275	-0.265	-0.128	-0.118

Experiment IV: Holding time, mean model coefficients, 1000 bootstraps (continued)

Coefficient	Point	Percentile point			
	Estimate	0.25	0.5	0.95	0.975
$(\alpha\kappa)_{63}$	-0.117	-0.416	-0.361	0.111	0.173
$(\alpha\kappa)_{24}$	-0.277	-0.335	-0.326	-0.217	-0.207
$(\alpha\kappa)_{34}$	-0.304	-0.361	-0.348	-0.247	-0.237
$(\alpha\kappa)_{44}$	-0.339	-0.394	-0.385	-0.284	-0.275
$(\alpha\kappa)_{54}$	-0.271	-0.340	-0.327	-0.206	-0.196
$(\alpha\kappa)_{64}$	-0.561	-0.783	-0.744	-0.361	-0.319
$(\beta\kappa)_{22}$	-0.323	-0.473	-0.451	-0.197	-0.175
$(\beta\kappa)_{32}$	-0.310	-0.460	-0.437	-0.186	-0.162
$(\beta\kappa)_{42}$	-0.344	-0.495	-0.471	-0.221	-0.193
$(\beta\kappa)_{23}$	-0.578	-0.749	-0.725	-0.446	-0.417
$(\beta\kappa)_{33}$	-0.541	-0.713	-0.683	-0.406	-0.381
$(\beta\kappa)_{43}$	-0.609	-0.780	-0.755	-0.476	-0.451
$(\beta\kappa)_{24}$	-0.840	-1.011	-0.976	-0.702	-0.675
$(\beta\kappa)_{34}$	-0.784	-0.954	-0.920	-0.647	-0.621
$(\beta\kappa)_{44}$	-0.867	-1.038	-1.004	-0.732	-0.709

Experiment IV: Holding time, variance model

Coefficient	Point	95% C.I.	
	Estimate	Lower	Upper
α_2	-0.467	-0.764	-0.169
α_3	-0.499	-0.797	-0.202
α_4	-0.491	-0.788	-0.193
α_5	-0.153	-0.419	0.113
α_6	-0.092	-0.572	0.388
β_2	-1.882	-2.892	-0.872
β_3	-1.457	-2.467	-0.447
β_4	-2.038	-3.049	-1.028
δ_2	0.974	0.083	1.865
δ_3	1.738	0.847	2.629
δ_4	2.962	2.071	3.853
κ_2	0.383	0.122	0.644
κ_3	0.446	0.185	0.707
κ_4	0.420	0.158	0.681
$(\beta\delta)_{22}$	0.809	-0.299	1.917
$(\beta\delta)_{32}$	0.567	-0.542	1.675
$(\beta\delta)_{42}$	0.731	-0.377	1.840
$(\beta\delta)_{23}$	1.330	0.222	2.438
$(\beta\delta)_{33}$	0.963	-0.145	2.071
$(\beta\delta)_{43}$	1.299	0.191	2.407
$(\beta\delta)_{24}$	1.897	0.788	3.005
$(\beta\delta)_{34}$	1.525	0.417	2.634
$(\beta\delta)_{44}$	2.102	0.994	3.211

Experiment IV: Holding time, variance model (continued)

Coefficient	Point Estimate	95% C.I. Lower	Upper
σ^2	0.024	-	-

Experiment IV: Approach sectors, variance model

Coefficient	Point Estimate	95% C.I. Lower	Upper
δ_2	0.476	0.244	0.707
δ_3	0.820	0.589	1.051
δ_4	2.115	1.884	2.346
κ_2	0.150	-0.004	0.305
κ_3	0.206	0.052	0.360
κ_4	0.264	0.110	0.418
σ^2	0.046	-	-

Experiment IV: Stability, mean model coefficients, 1000 bootstraps

Coefficient	Point Estimate	95% C.I. Lower	Upper
μ	0.104	0.097	0.112
α_2	-0.063	-0.069	-0.056
α_3	-0.064	-0.070	-0.057
α_4	-0.042	-0.049	-0.035
α_5	-0.069	-0.075	-0.062
α_6	-0.066	-0.075	-0.057
β_2	0.068	0.061	0.074
β_3	0.068	0.061	0.074
β_4	0.072	0.065	0.078
δ_2	0.052	0.034	0.071
δ_3	0.231	0.183	0.273
δ_4	0.751	0.576	0.916
κ_2	0.100	0.078	0.122
κ_3	0.171	0.148	0.197
κ_4	0.206	0.182	0.230
λ_2	-0.023	-0.034	-0.012
λ_3	-0.009	-0.020	0.002
$(\alpha\delta)_{22}$	-0.026	-0.046	-0.005
$(\alpha\delta)_{32}$	-0.023	-0.043	-0.005
$(\alpha\delta)_{42}$	-0.027	-0.049	-0.006
$(\alpha\delta)_{52}$	-0.023	-0.042	-0.004
$(\alpha\delta)_{62}$	-0.002	-0.030	0.026
$(\alpha\delta)_{23}$	0.021	-0.011	0.051
$(\alpha\delta)_{33}$	0.018	-0.013	0.048

Experiment IV: Stability, mean model coefficients, 1000 bootstraps (continued)

Coefficient	Point Estimate	95% C.I. Lower	Upper
$(\alpha\delta)_{43}$	0.052	0.014	0.085
$(\alpha\delta)_{53}$	0.018	-0.016	0.051
$(\alpha\delta)_{63}$	0.009	-0.039	0.061
$(\alpha\delta)_{24}$	0.247	0.182	0.307
$(\alpha\delta)_{34}$	0.227	0.169	0.284
$(\alpha\delta)_{44}$	0.520	0.432	0.594
$(\alpha\delta)_{54}$	0.348	0.274	0.414
$(\alpha\delta)_{64}$	-0.098	-0.271	0.085
$(\alpha\kappa)_{22}$	-0.026	-0.058	0.007
$(\alpha\kappa)_{32}$	-0.029	-0.058	0.003
$(\alpha\kappa)_{42}$	-0.035	-0.069	0.001
$(\alpha\kappa)_{52}$	-0.029	-0.060	0.001
$(\alpha\kappa)_{62}$	-0.056	-0.092	-0.018
$(\alpha\kappa)_{23}$	-0.041	-0.075	-0.006
$(\alpha\kappa)_{33}$	-0.050	-0.086	-0.015
$(\alpha\kappa)_{43}$	-0.056	-0.094	-0.018
$(\alpha\kappa)_{53}$	-0.054	-0.086	-0.022
$(\alpha\kappa)_{63}$	-0.099	-0.137	-0.060
$(\alpha\kappa)_{24}$	-0.069	-0.105	-0.035
$(\alpha\kappa)_{34}$	-0.076	-0.107	-0.043
$(\alpha\kappa)_{44}$	-0.108	-0.144	-0.072
$(\alpha\kappa)_{54}$	-0.096	-0.127	-0.065
$(\alpha\kappa)_{64}$	-0.153	-0.188	-0.118
$(\beta\delta)_{22}$	0.044	0.028	0.059
$(\beta\delta)_{32}$	0.044	0.029	0.059
$(\beta\delta)_{42}$	0.047	0.032	0.063
$(\beta\delta)_{23}$	0.124	0.081	0.165
$(\beta\delta)_{33}$	0.122	0.083	0.164
$(\beta\delta)_{43}$	0.142	0.101	0.182
$(\beta\delta)_{24}$	0.114	-0.049	0.284
$(\beta\delta)_{34}$	0.115	-0.049	0.288
$(\beta\delta)_{44}$	0.153	-0.005	0.324

Appendix D: Chapter 8

Wake-vortex parameters(%) for experiments on landing rate

IAF	Wake-vortex category	Wake-vortex level 1	2	3	4
1	H	3.10	14.10	26.60	39.10
1	M	96.90	74.90	49.90	24.90
1	L	0.00	11.00	23.50	36.00
2	H	0.00	0.00	0.00	0.00
2	M	99.68	77.68	52.68	27.68
2	L	0.32	22.32	47.32	72.32
3	H	2.44	13.44	25.94	38.44
3	M	96.33	74.33	49.33	24.33
3	L	1.24	12.24	24.74	37.24
4	H	4.75	26.75	51.75	76.75
4	M	95.25	73.25	48.25	23.25
4	L	0.00	0.00	0.00	0.00

Bibliography

Andreussi, A., Bianco, L. & Ricciardelli, S. (1981), 'A simulation-model for aircraft sequencing in the near terminal area', *European Journal of Operational Research* **8**(4), 345– 354.

Asano, M. & Ohta, H. (1996), 'Single machine scheduling using dominance relation to minimize earliness subject to ready and due times', *International Journal of Production Economics* **44**(1-2), 35– 43.

Azzalinin, A. (1996), *Statistical inference based on the likelihood*, Monographs on Statistics and Applied Probability, Chapman and Hall, New York.

Banks, J. (1998), *Handbook of simulation principles, methodology, advances, applications and practice*, John Wiley and Sons, New York.

Barco-Orthogon (2002), AMAN operational functions, Technical report, Barco Orthogon AG.

BBC News (2004), 'Government airport plans slammed', *BBC News Online* . Available online http://news.bbc.co.uk/1/hi/uk/3812853.stm (last accessed 26th September 2004).

Beasley, J., Krishnamoorthy, M., Sharaiha, Y. & Abramson, D. (2000), 'Scheduling aircraft landings - the static case', *Transportation Science* **34**(2), 180– 197.

Beasley, J., Krishnamoorthy, M., Sharaiha, Y. & Abramson, D. (2004), 'Displacement problem and dynamically scheduling aircraft landings', *Journal of the Operational Research Society* **55**(1), 54–64.

Beasley, J., Sonander, J. & Havelock, P. (2001), 'Scheduling aircraft landings at London Heathrow using a population heuristic', *Journal of the Operational Research Society* **52**(5), 483– 493.

Bianco, L., Dell'Olmo, P. & Giordani, S. (1999), 'Minimizing total completion time subject to release dates and sequence-dependent processing times', *Annals of Operations Research* **86**, 393– 415.

Bianco, L., Ricciardelli, S., Rinaldi, G. & Sassano, A. (1988), 'Scheduling tasks with sequence-dependent processing times', *Naval Research Logistics* **35**, 177– 184.

Bolender, M. & Slater, G. (2000), 'Evaluation of scheduling methods for multiple runways', *Journal of Aircraft* **37**(3), 410– 416.

Box, G. E., Hunter, W. G. & Hunter, J. (1978), *Statistics for experimenters*, John Wiley and Sons, Inc, New York.

Box, G. & Meyer, R. (1986), 'Dispersion effects from fractional designs', *Technometrics* **28**(1), 19–27.

Brentnall, A., Cheng, R., Drew, A. & Potts, C. (2003), The air traffic control arrival management problem, *in* D. Al-Dabbas, ed., 'Proceedings of the 2003 United Kingdom Simulation Society Conference', pp. 121–127.

Carr, G., Erzberger, H. & Neuman, F. (1999), 'Delay exchanges in arrival sequencing and scheduling', *Journal of Aircraft* **36**(5), 785– 791.

Carr, G., Erzberger, H. & Neuman, F. (2000), 'Fast-time study of airline-influenced arrival sequencing and scheduling', *Journal of Guidance Control and Dynamics* **23**(3), 526– 531.

Carroll, R. J. & Ruppert, D. (1988), *Transformation and Weighting in Regression*, Monographs on Statistics and Applied Probability, Chapman and Hall, New York.

Chatfield, C. (1980), *The Analysis of Time series: An Introduction*, 2nd edn, Chapman and Hall, New York.

Cheng, R. & Jones, O. (2000), Analysis of simulation factorial experiments by EDF resample statistics, *in* J. Joines, R. Barton, P. Fishwick & K. Kang, eds, 'Proceedings of the 2000 Winter Simulation Conference', pp. 697– 703.

Cheng, R. & Jones, O. (2004), Analysis of distributions in factorial experiments. To be published in Statistica Sinica.

Cox, D. R. & Hinkley, D. V. (1974), *Theoretical Statistics*, John Wiley & Sons, Inc., New York.

D'Agostino, R. B. & Stephens, M. A. (1986), *Goodness-of-fit techniques*, Marcel Dekker. Inc, New York.

Davidian, M. & Carroll, R. (1987), 'Variance function estimation', *Journal of the American Statistical Association* **82**(400), 1079–1091.

Davis, T., Isaacson, D., Robinson III, J., den Braven, W., Lee, K. & Sanford, B. (1997), Operational test results of the passive final approach tool, *in* 'IFAC 8th Symposium of Transportation Systems 97, Chania, Greece'.

Davison, A. C. (2003), *Statistical Models*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, Cambridge, UK.

Davison, A. C. & Hinkley, D. V. (1997), *Bootstrap Methods and their Application*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, Cambridge, UK.

Dear, R. & Sherif, Y. (1991), 'An algorithm for computer-assisted sequencing and scheduling of terminal area operations', *Transportation Research part A - Policy and Practice* **25**(2-3), 129– 139.

Draper, N. R. & Smith, H. (1998), *Applied Regression Analysis*, Wiley Series in Probability and Statistics, 3rd edn, John Wiley & Sons, Inc., New York.

Efron, B. & Tibshirani, R. J. (1998), *An Introduction to the Bootstrap*, Monographs on Statistics and Applied Probability, Chapman and Hall, New York.

Ernst, A., Krishnamoorthy, M. & Storer, R. (1999), 'Heuristic and exact algorithms for scheduling aircraft landings', *Networks* **34**(3), 229– 241.

Eurocontrol (2000*a*), AMAN feasibility study (part 1), Technical report, Eurocontrol. Programme reference index: ASA.02.AMAN.DEL01.FEA.1.

Eurocontrol (2000*b*), AMAN feasibility study (part 2) literature study and site visit reports, Technical report, Eurocontrol. Programme reference index: ASA.02.AMAN.DEL01.FEA.2.

Fahle, T. & Wong (2003), 'The aircraft sequencing problem', *Computer Science in perspective: Essays dedicated to Thomas Ottmann* **2598**, 152– 166.

Gilbo, E. P. (1993), 'Airport capactity: represenation, estimation, optimization', *IEEE Transactions on control systems technology* **1**(3), 144–154.

Gilbo, E. P. (1997), 'Optimizing airport capacity utlization in air traffic flow management subject to constraints at arrival and departure fixes', *IEEE Transactions on control systems technology* **5**(5), 490–503.

Goos, P., Tack, L. & Vandebroek, M. (2001), 'Optimal designs for variance function estimation using sample variances', *Journal of Statistical Planning and Inference* **92**(1-2), 233–252.

Graves, D. (1998), *UK Air Traffic Control*, 3rd edn, Airlife Publishing Ltd, Shrewsbury, England.

Hauck, W. & Donner, A. (1977), 'Wald's test as applied to hypotheses in logit analysis', *Journal of the American Statistical Association* **72**(360), 851–853.

Hopkins, D. (1995), *Human Factors in Air Traffic Control*, Taylor and Francis, London.

Kiefer, J. (1959), 'K-Sample analogues of the Kolmogorov-Smirnov and Cramer-V. Mises Tests', *The Annals of Mathematical Statistics* **30**(2), 420–447.

Kleijnen, J. P. C. (1995), 'Verification and validation of simulation-models', *European Journal of Operational Research* **82**, 145–162.

Law, A. M. & Kelton, W. D. (2000), *Simulation modeling and analysis*, 3rd edn, McGraw Hill, New York.

Loemker, L. (1969), *W.G. Leibniz: Philosophical Papers and Letters*, 2nd edn, Dordrecht: D. Reidel.

McCullagh, P. & Nelder, J. A. (1983), *Generalized Linear Models*, Monographs on Statistics and Applied Probability, Chapman and Hall, New York.

McCulloch, C. E. & Searle, S. R. (2001), *Generalized, Linear, and Mixed Models*, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., New York.

Milan, J. (1997), 'The flow management problem in air traffic control: a model of assigning priorities for landings at a congested airport', *Transportation Planning and Technology* **20**(2), 131– 162.

Mohleji, S. (1996), 'A route-oriented planning and control concept for efficient flight operations at busy airports', *Control Engineering Practice* **4**(8), 1143– 1151.

Myers, R. H., Montgomery, D. C. & Vining, G. (2002), *Generalized Linear Models*, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., New York.

Ovacik, I. & Uzsoy, R. (1994), 'Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times', *International Journal of Production Research* **32**(6), 1243– 1263.

Ovacik, I. & Uzsoy, R. (1995), 'Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times', *International Journal of Production Research* **33**(11), 3173– 3192.

Performance Review Unit (1999), An ATM Measurement System - Key Performace areas, Technical report, Eurocontrol, Brussells, Belgium. Available online http://www.eurocontrol.int/prc/gallery/content/public/Docs/kpa.pdf (last accessed 15th November 2004).

Performance Review Unit (2004), An assessment of Air Traffic Management in Europe during the Calender Year 2003, Technical report, Eurocontrol, Brussells, Belgium. Available online http://www.eurocontrol.int/prc/gallery/content/public/Docs/prr7.pdf (last accessed 15th November 2004).

Pinedo, M. (1995), *Scheduling Theory, Algorithms, and Systems*, Prentice-Hall International.

Potts, C. & Kovalyov, M. (2000), 'Scheduling with batching: a review', *European Journal of Operational Research* **120**(2), 228– 249.

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

Rice, J. A. (1995), *Mathematical Statistics and Data Analysis*, 2nd edn, Duxbury Press, Belmont, California.

Robinson, S. (2004), *The Practice of Model Development and Use*, John Wiley & Sons, Chichester, England.

Santner, T. J. & Duffy, D. E. (1989), *The Statistical Analysis of Disrete Data*, Springer Texts in Statistics, Springer-Verlag, New York.

Sargent, R. (2001), Some approaches and paradigms for verifying and validating simulation models, *in* B. Peters, J. Smith, D. Medeiros & M. Rohrer, eds, 'Proceedings of the 2001 Winter Simulation Conference', pp. 106– 114.

Scholz, F. & Stephens, R. (1987), 'K-Sample Anderson-Darling tests', *Journal of the American Statistical Association* **82**(399), 918–924.

Searle, S. R. (1971), *Linear Models*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Inc., New York.

STATFOR (2004), Forecast of annual number of IFR flights (2004-2010), Vol.1, Technical report, Eurocontrol, Brussells, Belgium. EATMP Infocentre reference: 040219-01, Available online from http://www.eurocontrol.int/statfor/forecasts (last accessed 15th November 2004).

Trivizas, D. (1994), 'TMSIM - a runway capacity study for Frankfurt and Chicago O'Hare airports', *Journal of Navigation* **47**(1), 70– 88.

Trivizas, D. (1998), 'Optimal scheduling with Maximum Position Shift (MPS) constraints: a runway scheduling application', *Journal of Navigation* **51**(2), 250–266.

Uzsoy, R., Lee, C. & Martinvega, L. (1992), 'Scheduling semiconductor test operations - minimizing maximum lateness and number of tardy jobs on a single-machine', *Naval Research Logistics* **39**(3), 369– 388.

Uzsoy, R., Martinvega, L., Lee, C. & Leonard, P. (1991), 'Production scheduling algorithms for a semiconductor test facility', *IEEE Transactions on semiconductor manufacturing* **4**(4), 270– 280.

Venables, W. N. & Ripley, B. D. (1999), *Modern Applied Statistics with S-PLUS*, Statistics and Computing, 3rd edn, Springer-Verlag, New York.

Venkatakrishnan, C., Barnett, A. & Odoni, A. (1993), 'Landings at Logan-airport - describing and increasing airport capacity', *Transportation Science* **27**(3), 211–227.

Wong, G. (2000), The dynamic planner: the sequencer, scheduler, and runway allocator for air traffic control automation, Technical report, NASA, Moffett Field, California. Report NASA/TM-2000-209586, Available online from http://www.ctas.arc.nasa.gov/publications (last accessed 15th November 2004).

Wong, J., Li, S. & Gillingwater, D. (2002), 'An optimization model for assessing flight technical delay', *Transportation Planning and Technology* **25**(2), 121– 153.

Wu, C. F. J. & Hamada, M. (2000), *Experiments: Planning, Analysis, and Parameter Design Optimization*, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., New York.