Ground Holding Strategies for Air Traffic Control Under Uncertainty

by

Octavio Richetta

B.S., Chemical Engineering, Columbia University (1981)
M.B.A., Finance and International Business, Columbia University (1983)

Submitted to the Department of Electrical
Engineering and Computer Science
in Partial Fulfillment of
the Requirements for the Degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

at the

Massachusetts Institute of Technology

May 1991

Signature of Author ______________________________
Department of Electrical Engineering and Computer Science
May 8, 1991

Certified by ______________________________
Amedeo R. Odoni
Professor of Aeronautics and Astronautics and of Civil Engineering
Thesis Supervisor

Accepted by ______________________________
Amedeo R. Odoni
Codirector, Operations Research Center

Ground Holding Strategies for Air Traffic Control Under Uncertainty

by

Octavio Richetta

Submitted to the Department of
Electrical Engineering and Computer Science
on May 10, 1991
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Operations Research

ABSTRACT

As air traffic congestion grows, ground-holding (or "gate-holding") of aircraft is becoming increasingly common. Ground-holding is the practice of delaying the departure of a flight due to anticipated congestion at the airport of destination, because it is both less expensive and safer, for aircraft to wait on the ground, prior to take-off, than in the air.

The problem of developing strategies for deciding which aircraft to hold on the ground and for how long (the "ground-holding policy problem," GHPP) is a difficult one. The GHPP is: *combinatorial*, because of the large number of aircraft and landing periods ("slots"); *stochastic*, because there is often a large amount of uncertainty about the acceptance rates ("landing capacity") of the airports of destination; and *dynamic*, because the state of information regarding airport capacity and expected delays is updated throughout the day.

Since ground-holding is only a relatively recent phenomenon, little has been done to date on developing sophisticated solutions to the GHPP which could potentially result in very large cost savings to the users of the ATC system. Research on probabilistic models of the GHPP has been limited to static solutions, producing "once and for all" ground-holds at the beginning of daily operations. The computational complexity of such models requires heuristic approaches in order to solve practical instances of the problem.

In this thesis, we present first a dynamic solution to the probabilistic GHPP based on a dynamic programming algorithm that exercises control on individual planes by deciding whether or not flights should be allowed to depart at each time period. The computational complexity of the algorithm limits its practicality. We were able to solve only small problems using a CRAY-2 supercomputer. However, the exact modelling approach provides insights which lead to a simplified model that captures the key elements of the real system and through which high quality solutions can be obtained.

This new model simplifies the structure of the control mechanism by exercising ground-holding on groups of aircraft instead of individual flights. Static and dynamic optimal solutions to the simplified model are derived using stochastic linear programming with recourse. The algorithms allow for general ground-hold cost functions for several aircraft classes and the cost of air delays is assumed to be identical for all planes. The resulting linear programs are manageable for practical size problems. We have been able to solve problem instances for one of the largest airports in the U.S using just a powerful PC.

Both, the static and dynamic stochastic programming formulations, indicate cost advantages when compared with deterministic solutions under different weather scenarios. The dynamic stochastic programming algorithm performed significantly better than the static motivating the development and implementation of a very fast dynamic heuristic that works with a deterministic forecast. This heuristic also gave solutions which are better than those obtained through the static stochastic programming algorithm. We also compare performance of the algorithms tested to the passive strategy of no ground-holds, i.e. to the strategy of taking all delays in the air.

Finally, we discuss the GHPP for the entire air traffic network. Possible approaches to exact and heuristic modelling are explored and new research directions are proposed.

Thesis Supervisor: Professor Amedeo R. Odoni

Title: Professor of Aeronautics and Astronautics
and of Civil Engineering
Codirector, Operations Research Center

Acknowledgements

Many names come to mind when trying to decide whom to thank for making this thesis come through. MIT is a place so special that it managed to attract the wonderful people - professors, administrators and students alike - that have made my years here some of the best I've had. Two friends deserve special mention: Professors Richard C. Larson and Amedeo R. Odoni, who have offered selfless professional, as well as personal, guidance throughout my stay here.

I still recall meeting Dick and his wife Liz in Caracas, and how they introduced me to Operations Research and the attractions of student life at MIT. Dick also provided valuable academic and personal advice during my first two years at MIT. It was from the Larsons' former Winthrop home that I had my first views of Logan Airport and Boston Harbor, which later provided many hours of work and sailing enjoyment.

More than an advisor, Amedeo has been kind of an "Italian Father". Amedeo and his wife Eleni were assiduous supporters of the ORSA student chapter activities I planned and helped me during some difficult times I went through. As thesis advisor, Amedeo always managed to "steer" my research energies into fruitful directions. I will always remember phrases such as "Octavio, there must be ways of solving problems for more than five aircraft".

I also thank: Alvin Drake for his timely support when I first came to MIT and for his lively teaching style; Karla Ballman, Jenny Wagner, and Ian Webb for their early tips on how to survive at MIT; Robert Gallager for his superb teaching and the opportunity of working with him; Rob Freund for his nice introduction to mathematical programming; Jerry Shapiro for his research and personal advice; Haris Koutsopoulos for his dedication to quality teaching and help in the early stages of my research; Dimitris Bertsimas for his help in planning the ORC Seminar Series last Fall and the interest he has shown for my work; Stephan Kolitz, of the C. S. Draper Laboratory, for helping me to focus on practical aspects of the problem; Laura Terrell who typed the manuscript doing excellent cryptography work; and all the students that make the ORC such a great place.

The C.S. Draper Laboratory is gratefully acknowledge for funding my research over the last two years.

Lastly, I would like to express my appreciation to my parents for their example, support and confidence; my brother Romano for initiating a tradition of schooling at MIT that ends with Antulio - the youngest in the family; Cecilia, my girlfriend, whose charm, love, and concern made it all seem so easy; and Incoronata Centanni, my "American Mother", who was always ready to offer loving support and advice.

This thesis is dedicated to the inquisitive spirit that lives in all of us.

Table of Contents

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Tables

Chapter 2

Chapter 5

CHAPTER 1

1. INTRODUCTION

1.1 The Need for Air Traffic Control (ATC)

The US air traffic network started to experience significant congestion during the decade of the 80's. Currently, 450 million airline passengers a year travel in the US [1] generating approximately 100,000 operations every 24 hours [2], with at least 18 hub airports exceeding their practical annual capacity and 13 additional major airports expected to be fully congested by the early 1990's [3].

Since the system is operating under significant congestion, flow must be managed so that it can proceed without risk to safety. The ATC system does a very good job in this regard. However, there is room for improving the efficiency of operations. During 1986, ground delays alone averaged 2000 hours per day, equivalent to grounding 250 airplanes which represents a carrier about the size of Delta Airlines at the time [4]. Capacity shortages are likely to continue during the next decade, even under the most ambitious airport development scenarios as the number of operations and of passengers are forecast to grow by factors of 1.5 and 2.5 respectively by the year 2010 [5]. Providing enough capacity to accommodate all demand even under severe weather conditions and/or in the face of airlines' peak scheduling practices is not cost efficient as it would result in low system utilization. Thus, improved air traffic flow management tools will be one of the keys in accommodating the expected growth in the number of operations.

In 1988, IBM was granted a $3.6 billion contract to start development of the Advanced Automation System (AAS), which will be the foundation of the ATC system by the year 2010 [1]. The AAS will offer automated decision making. Thus, the controller's function will advance to a more managerial role from that of a tactician responding to individual targets on a radar screen.

The software functions that will be incorporated into the AAS include advanced traffic management (ATMS), automated en route ATC (AERA), terminal ATC automation, and automatic dependent surveillance [5]. ATMS software functions will help air traffic managers anticipate and resolve imbalances between traffic demand and system capacity. Algorithms that assign optimal ground-hold strategies, such as the ones developed in this thesis, will eventually become a key component of the ATMS. For a detailed description of the AAS the reader is referred to [5].

1.2 Centralized Approach to ATC

Any attempt to improve the efficiency of the air traffic network requires a coordinated approach as the interests of individual ATC units are frequently in conflict with network wide optimization. Fortunately, the structure needed to effect this type of coordination in the US is already in place.

The 1981 ATC controllers' strike laid the ground for a more active control and coordination of air traffic volumes, with the ATC Central Flow Control Facility (CFCF) in Washington DC regulating traffic flow between key airports, when needed, on a daily basis. The CFCF continues to be a vital part of the ATC system today. Assigning ground-holds to flights that would otherwise be delayed at the arrival airport is one of the CFCF's main functions.

In Western Europe, where air traffic congestion problems are also significant, it is estimated that 35 of the key commercial airports are experiencing capacity problems [6]. Air traffic congestion in Europe is intensified by the lack of coordination among the 42 different control centers situated in 22 countries. A study by the West German Institute for Technology puts the avoidable cost of air traffic delays and deficiencies to European airlines at $5 billion in 1988, of which $1.5 billion resulted from delays, $1.8 billion from inefficient routing and $700 million from use of non-optimal flight profiles [7]. It is estimated that unifying the ATC system in Western Europe would result in adequate capacity through the 1990's at most key airports [8] (similar views are expressed in [7]). One of the key elements of the proposed unified (or, at the very least, tightly coordinated) system is a Central Flow Management Unit whose role will be analogous to that of the CFCF in the United States.

1.3 Current Flow Management Practices in the US ATC System

In order to exercise effective flow management, traffic volume and the capacity of the different elements in the ATC system must be assessed on a daily basis. The CFCF, in coordination with the Traffic Management Units (TMUs) in key terminals and the Air Route Traffic Control Centers (ARTCCs), attempts to manage efficiently the flow of air traffic by insuring that airport and airway capacity are not exceeded, while trying to maximize utilization of available capacity. The key traffic management tools in US ATC are comprised of the following programs [2]:

1.3.1 Expanded Quota Flow (EQF)

EQF is a ground delay program administered by the CFCF which controls the flow of traffic by holding the aircraft on the ground. This is one of the most effective and economical traffic management tools as it reduces air congestion, limiting the duration of airborne delays which are more expensive and less safe.

Implementing efficient ground-holds is difficult as uncertainties in weather affect airport capacity significantly. Currently, the EQF program uses a deterministic landing capacity forecast for each of the major airports. It assigns forecasted capacity on a first-come first-served basis with all expected delays exceeding 15 minutes assigned as ground-holds.

Due to the probabilistic nature of airport capacities, even under optimal assignment of ground-holds, there will be instances in which airport landing capacity is lost while planes sit waiting on the ground. However, these occurrences may currently be more frequent than necessary when bad weather is expected, as airports tend to provide conservative capacity forecasts to protect their airspace from saturation. As a result, carriers are often advocating more "liberal" ground-holding strategies, even to the extreme of returning to the procedures used previous to the 1981 strike [4], when lower congestion levels made it possible to absorb most delays in the air. The trade-off in establishing ground-hold delays is between conservative policies that may at times assign excessive ground-holds and optimistic ones that may result in more expensive air holds.

We see that an effective EQF program should: i) consider the cost structure of ground and air delays; ii) improve forecasting practices in order to reflect the uncertainties in airport capacity due to weather; and iii) be able to respond to a constantly changing system. The algorithms developed in this thesis will begin to address these key issues.

1.3.2 En Route Spacing (ESP), Departure Spacing (DSP), and Arrival Spacing (ASP) Programs

These three programs are real-time tactical ATC tools that interact closely in order to control traffic flow so as to decrease congestion and improve capacity utilization at the arrival airports. ESP is implemented during periods of peak arrival demand and consists of path stretching, speed control and rerouting of airborne aircraft in order to provide appropriate spacing between aircraft proceeding to the same airport. DSP controls departure flow during severe weather conditions and releases departing aircraft taking into consideration the number of en route aircraft already bound for the same airport. ASP is used to limit arrival rates into a terminal area so that these rates match the maximum airport capacity for the given weather conditions and runway configuration. ESP, DSP, and ASP are "fine-tuning" tools and should be part of an ATC system that utilizes an efficient EQF program as its strategic foundation.

1.4 Systematic View of the Congestion Problem in ATC

Due to the complexities associated with the congestion problem in ATC and the possibility of dealing with it in different ways, a systematic approach to the problem is necessary. The following discussion is based on a paper by Odoni [9], which to our knowledge offers the first systematic discussion of the problem. We start by classifying the different approaches to deal with the problem and then focus on describing the one relevant to this thesis.

Possible actions for dealing with the congestion problem in ATC can be classified according to time span.

- **Long-term approaches** are aimed at increasing network capacity through improved ATC technologies which result in a more efficient utilization of airspace and runways; and construction of new facilities such as new airports and/or additional or better airways in existing airports. These actions typically have a time horizon of 5 to 10 years, require significant capital investment and may encounter public opposition since new facilities are frequently needed close to large metropolitan areas.

- **Medium-term approaches** try to modify demand patterns in order to alleviate congestion and have a time span of 6 months to 2 years. Possible actions include imposition of time-varying landing fees and user charges to encourage off-peak use of airports and imposition of quotas on airport use [10]. These types of actions are effective in reducing congestion to the extent that airlines are willing/able to modify current scheduling practices, which tend to cluster landings and take-offs due to the hub/spoke system of operation and trying to accommodate passenger preference for flying during peak hours.

- **Short-term approaches** deal with congestion on a daily basis and comprise the ATC practices described in section 1.3. Such actions try to optimize operations for a given daily schedule of flights and network capacity. The objective is to best match demand with available capacity of the various components of the ATC network on a daily basis. This is known as the ATC Flow Management Problem (FMP).

1.4.1 Classification of Short Term ATC Practices

The ATC practices described in Section 1.3 can be classified as being either of a strategic or of a tactical nature. Implementing optimal ground-holds, as in the EQF program, requires anticipating network capacity, assessing the effects of current ground-holds as well as trying to determine future ground-holds. Thus, the EQF program can be seen as a strategic tool. On the other hand, actions taken under ESP, DSP, and ASP respond to real-time requirements of the ATC network and can be classified as tactical.

This thesis addresses strategic aspects of the FMP as they play a more important role in reducing the cost of operations. Thus, we concentrate on trying to resolve the most important strategic choice in ATC, namely the trade-off between ground-holding delays and airborne delays. We refer to this as the Ground-holding Policy Problem (GHPP). Next, we describe the model of the air traffic network used as the basis for solving the GHPP in its most generic form.

1.4.2 Model for The GHPP

The generic model presented here is discussed in [9]. The model is macroscopic in nature, yet it captures the essential elements needed to solve the GHPP. It avoids excessive detail by introducing reasonable assumptions that simplify the problem without affecting significantly the quality of the solution. For example, we assume constant flight times, ignoring the effect of tactical actions such as metering, speed control and path - stretching which have a lesser impact on operating costs than the ground-hold versus air delay trade-off.

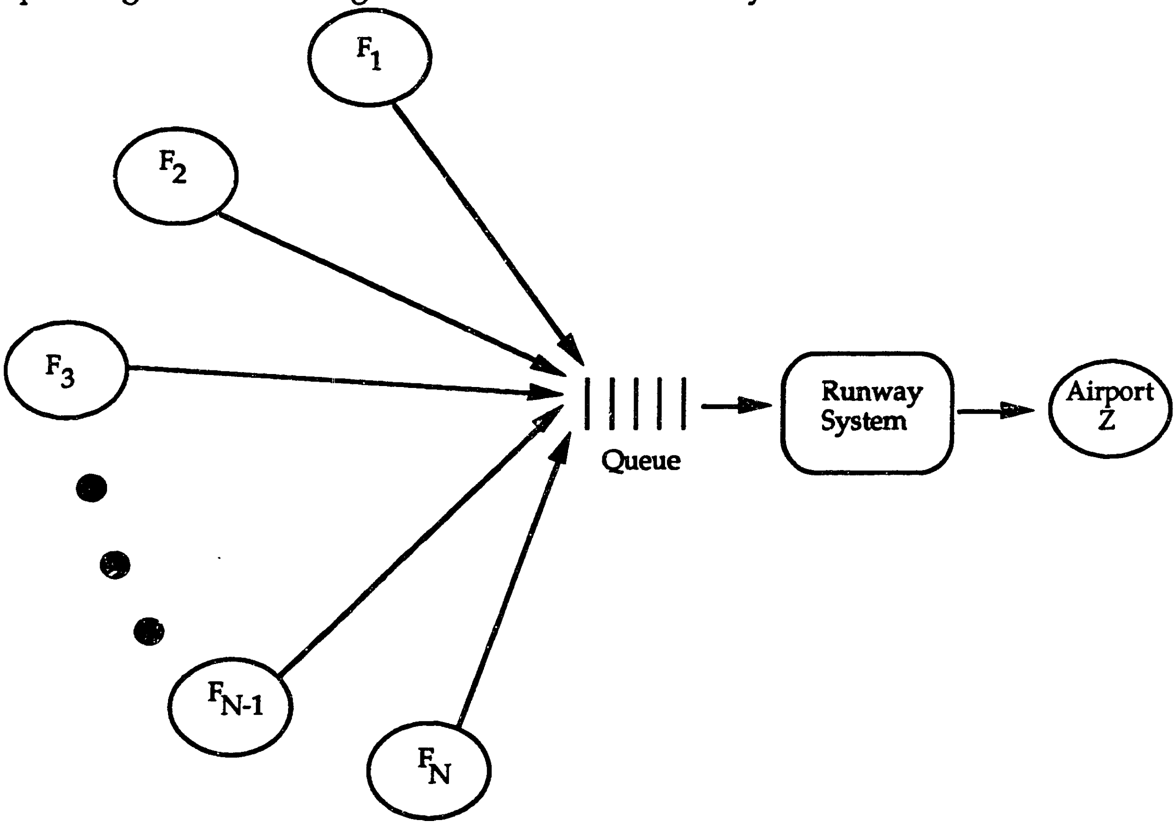


Figure 1.1 "Star" Configuration Network

The air traffic network considered under the GHPP can be described with reference to the single-destination network shown in Figure 1.1. It is understood that in the complete network all airports handle arrivals as well as departures. Assumptions are as follows:

(i) N aircraft are scheduled to arrive at the "arrivals" airport Z from the "departures" airports.

(ii) Airport Z is the only capacitated element of the network and thus the only source of delays. All other elements in the network (departure capacities, airways, etc.) have unlimited capacity.

(iii) Departure and travel time of each aircraft is deterministic and known in advance.

(iv) The time interval of interest is [0, L], with the earliest departure scheduled at 0 and the latest arrival scheduled at L. The time interval [0, L] is discretized into T time periods numbered 1, 2, . . . T.

(v) At each t, we have access to the joint probability mass function (PMF) of airport Z capacities, $K_{t+1},..., K_T$: $P_{K_{t+1} \ldots K_T}(t)$. Capacity for each period t (with t=1, 2, . . . T) becomes known at the beginning of t. Also, the capacity, K_{T+1}, at period T+1 is infinity.

(vi) Ground and air delay cost functions for each flight are known:

- $c_{gi}(j)$ is the marginal cost of delaying Flight i for the j^{th} time period, on the ground.

- c_{ai} (j, k–j) is the marginal cost of delaying Flight i for the $(k\text{–}j)^{th}$ time period in the air where j is the time period of the scheduled arrival of Flight i to airport Z.

The relevance of the above assumptions is discussed in [9] and in [11]. From this generic model we see that the following aspects must be considered in solving the GHPP:

- **The problem is stochastic.** Airport capacity is affected significantly by adverse weather conditions. Due to the uncertainty in the weather, a probabilistic forecast of airport capacity must be considered.

- **The problem is dynamic.** The probability distribution of airport capacities evolves through time. Moreover, even for a static probability distribution of airport capacities, a dynamic approach to problem solution may result in better strategies as the history of airport capacities is considered when making ground-hold decisions.

- **The problem is combinatorial.** Because ground and air delay costs may differ among aircraft, the composition of the flow arriving at airport Z must be determined.

- **Network dependencies complicate the problem.** In the complete network, the effect of delays on departure times of connecting and continuing flights must be considered.

1.5 Research to Date

Interest in solving the GHPP developed recently as congestion of the ATC network became more significant during the decade of the 1980's. Thus, the literature dealing with the problem is limited. Odoni presents the first systematic description of the GHPP in [9] and discusses the full complexity of the most generic version of the problem. Due to the complications arising in trying to solve the generic version of the problem, initial research has focused on simplified versions. The networks considered so far consists of the single destination network of Figure 1.1 which we refer to as the "star configuration" problem. Following is a review of research to date on the star configuration problem, including current CFCF ground-holding practice.

1.5.1 Current CFCF Practice

As described in Section 1.3, the CFCF assigns ground-holds using a deterministic landing capacity forecast for each airport. Available capacity is then assigned on a first-come first-served basis with expected delays, above a threshold of 15 to 20 minutes, assigned as ground-holds. We see that, if the delay threshold were zero, this approach would certainly minimize operating costs of the "star" network under the assumption of deterministic capacities and constant marginal ground and air delay costs (c_a and c_g) equal for all planes, with $c_a > c_g$.

1.5.2 The Deterministic Problem With Known Delay Cost Functions For Each Flight

In his PhD thesis "Ground-Holding Strategies For Air Traffic Control" [11], Terrab showed that this problem can be formulated as a minimum cost flow problem and can be transformed to the assignment problem solvable in time $O(N^3)$. Terrab also developed an algorithm that solves the problem in $O(N \ln N)$ steps for a special class of cost functions.

These algorithms take into consideration the differences in operating costs for different types of aircraft. However, they do not consider the probabilistic nature of airport capacities.

1.5.3 The Single Time Period Probabilistic Problem

The first attempt to solve a probabilistic version of the GHPP appears in a 1987 paper by Andreatta and Romanin-Jacur [12]. Capacity is assumed to be limited during a single time period according to a specified probability distribution. The algorithm developed is based on dynamic programming (DP) with flights ranked in decreasing priority order as the stage variable. The algorithm is polynomial in the number of flights with complexity $O(N^2)$.

This algorithm has an advantage over a deterministic approach in that it considers the stochastic nature of airport capacities. However, the analysis is limited to a single congested period. The solution is static since it determines a "once and for all" ground-hold policy at the time of the first take off, ignoring the evolution of the capacity forecast over time.

1.5.4 The Multi-Period Static Probabilistic Problem

Terrab [11] solved the multi-period version of the static probabilistic problem through a dynamic programming algorithm which is an extension of the algorithm developed by Andreatta and Romanin-Jacur. The algorithm uses flights ordered according to a fixed landing priority rule as the stage variable. The complexity of the algorithm is $O(N\ C\ (T+1)^2 M^T)$, where C is the number of capacity cases in the capacity forecast and M is the maximum number of possible capacity values for a single time period.

The magnitudes of N, M and T found in practice limit the use of the exact formulation. For example, at Boston's Logan Airport N is in the order of 500 to 600 flights; for a time span of 15 hours, with 15 minute duration periods, the number of periods T is 60; and M is as high as 15-16 landings per 15 minute period. This algorithm produces a static solution for a time invariant capacity forecast, and a static landing priority rule. A dynamic approach to solving the problem may improve the solution as the history of airport capacities is considered when making ground-hold decisions. A dynamic approach also allows landing rules that reflect ATC landing practices such as first-come first-served landing, and landing according to decreasing marginal cost.

1.5.4.1 Heuristic Solutions to the Multi-Period Probabilistic Problem

Due to the limitations of the static dynamic programming algorithm in solving practical problems, Terrab considered heuristic approaches to problem solution [11]. Here we discuss briefly the two best performing of these heuristics. Both were tested with a sample problem generated using the operations profile for a typical 1987 day at Logan Airport.

- **Limited Look-ahead:** The time period of interest is divided into Q subproblems, each with R time periods (R = T/Q). The dynamic programming algorithm is then used to solve each subproblem. The complexity of this heuristic is $O(Q\ N\ C\ (R+1)^2\ (M+1)^R)$, and we see that the heuristic is of practical value only for small values of R. Since each subproblem is solved assuming unlimited capacity for future time periods, we see that the performance of the heuristic deteriorates for problem instances showing significant congestion.

- **Greedy Heuristic:** This heuristic assigns ground-holds for each flight independently, according to a fixed landing priority rule. It starts with the highest priority flight and assigns to it a ground-hold that minimizes the expected cost of delay for that flight. It then moves down the list of flights according to decreasing priority. The complexity of this heuristic is $O(N\ C(T+1)^2)$ and it performs as well and often better than the Limited Look-ahead heuristic for a special class of cost functions.

1.6 Thesis Objective

This thesis focuses on solving probabilistic versions of the GHPP for a star configuration network. The model used is basically the one described in Section 1.4.2 modified to include operational aspects of the ATC system such as forecasting capacities, equal access policies, and air delay cost structures that are in line with the macroscopic nature of the model.

The primary objective of this thesis is to develop static and dynamic algorithms that provide high quality solutions to practical instances of the GHPP for the largest airports in the US. The secondary objective is to set the stage for future research on problems involving complete networks of airports.

1.7 Thesis Content

In Chapter 2 we will look at a dynamic optimal solution to the GHPP on a star configuration network in its most generic version. The algorithm presented is based on dynamic programming, using the beginning of time periods as the stage variable. An improvement over the static dynamic programming solution [11] is possible even for the case of a time invariant PMF for airport landing capacities, as the history of past capacities is considered when making decisions at each stage. The algorithm also allows for landing rules which reflect current ATC practice and are in line with the macroscopic aspects of the model. The algorithm was implemented on the CRAY-2 supercomputer. Although the complexity of the algorithm limits application to small problems, it has provided some valuable insights on how to revise the original model in order to develop optimal algorithms with practical relevance.

Chapter 3 presents an optimal static solution to the GHPP based on stochastic programming with single recourse. The resulting algorithm yields a linear program. Problem size is essentially independent of the number of flights. Thus, practical problems can be solved even for the largest airports in the network. The formulation allows for general ground-hold cost functions with aircraft grouped in up to three cost categories. All planes are assumed to have identical air delay cost function. We discuss why this is a sound assumption given the macroscopic nature of the model. We also present a decomposition algorithm that exploits the block angular structure of the constraint matrix for this stochastic linear programming model.

Chapter 4 presents an optimal dynamic solution based on stochastic programming with recourse. Ground-holds can be assigned up to three times during the day, resulting in an improvement over the static solutions obtained in Chapter 3. The algorithm yields a linear program of larger size than in the case of the static solution but still manageable for practical problems. The algorithm allows for general ground-hold cost functions for several aircraft classes. The cost of air delays is assumed to be identical for all planes.

In Chapter 5 we assess the performance of the stochastic programming models developed in Chapters 3 and 4 using 1988 airline schedule data for Logan Airport under a variety of weather conditions. Both, the static and dynamic stochastic programming formulations, indicate cost advantages when compared with deterministic solutions. The dynamic algorithm performed significantly better than the static motivating the development and implementation of a very fast dynamic heuristic that works with a deterministic forecast. This heuristic also gave solutions which are better than those obtained through the static stochastic programming algorithm. We also compare performance of the algorithms tested to the passive strategy of no ground-holds, i.e. to the strategy of taking all delays in the air.

Finally, in Chapter 6 we present the conclusions from the modeling and experimental work performed in this thesis and discuss possible approaches for solving the GHPP for the entire air traffic network. Possible approaches to exact and heuristic modelling are explored and new research directions are proposed.

CHAPTER 2

2. DYNAMIC SOLUTION TO THE GHPP

As discussed in Section 1.4, the GHPP has a dynamic nature. Therefore, a dynamic approach to solving the problem should yield an improvement vs. a static solution. Terrab solved the GHPP on a star configuration network through a dynamic programming algorithm that uses flights in increasing priority order as the stage variable [11]. The choice of stage variable does not address the time dynamic nature of the problem. Thus, the algorithm produces a static optimal solution for a time invariant PMF for airport landing capacities. Also, the algorithm requires the assumption of a fixed (i.e., static) landing priority rule while the real system operates closer to a first-come first-served (FCFS) fashion. This affects the quality of the solution, generating "truly" optimal static solutions only in the case of constant marginal air delay costs identical for all planes, for which the cost of air delays is independent of the landing rule.

In this chapter, we present the fully dynamic solution to the GHPP on a star configuration network, in its most generic version, by exercising ground-hold control on individual flights at the beginning of each time period. The algorithm developed is based on dynamic programming and uses the beginning of time periods as stage variable. An improvement vs. the static solution is possible even for the case of a time invariant (i.e., static) PMF as the history of airport capacities is considered when making decisions at each stage. Also, the algorithm allows for dynamic landing rules which are closer to current ATC practice (e.g., FCFS, landing according to decreasing marginal cost, etc.), improving the quality of the solution.

The chapter is organized as follows: Section 2.1 describes the dynamic problem and illustrates the advantage of solving the GHPP dynamically through a simple example. Section 2.2 specifies problem inputs, develops notation and presents the algorithm. In Section 2.3 we discuss an extension of the algorithm covering cases for which, in addition to limited landing capacity, we can also have capacity constraints at the departure airports. In section 2.4 we discuss algorithm complexity, C language implementation on the CRAY-2 supercomputer to solve small problems, and discuss the practical limitations of the fully dynamic algorithm. Finally, in Section 2.5 we suggest modeling approaches that would facilitate solution of practical problems even for the largest airports in the US ATC network.

2.1 The Dynamic GHPP in ATC

The dynamic GHPP in ATC differs from the static problem in the approach to solution. In the dynamic problem, the expected cost of ground plus air delays is minimized by deciding whether eligible[1] flights are allowed to depart or held on the ground at the beginning of each time period, while the static solution produces "once and for all" optimal ground-holds at time zero (i.e., at the beginning of the first time period).

The advantage of the dynamic solution over the static is illustrated by the following two flight - three time periods example. Figure 2.1 shows a diagram of the flight schedule. F_1 is scheduled to depart at time 1 and arrive at time 3; while F_2 is scheduled to depart at time 2 and arrive at time 3.

[1] A flight is eligible to be delayed during time period i if it is scheduled to depart during or before i and it has not yet departed.

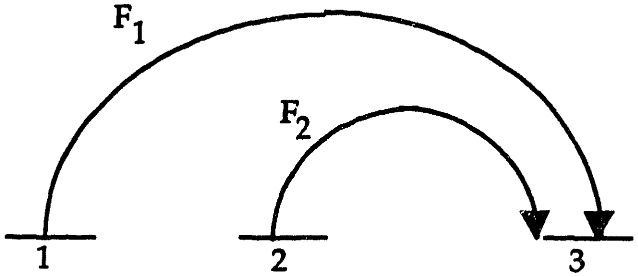


Figure 2.1

Flight Schedule

Landing capacity during the arrival period, time 3, is limited to one or two flights according to the probability tree shown in Figure 2.2. Notice that capacity during time 4 is unrestricted, and we recall that in our model landing capacity becomes known at the beginning of each time period.

We see that landing capacity during time 3 depends on the capacity during time 2. If time 2 capacity is 2 (i.e., the weather during time 2 is good), then there is a greater chance of having a high capacity during time 3; while if time 2 capacity is 1, the probability of having limited capacity during time 3 increases.

Next we specify the ground and air delay costs for F_1 and F_2. Since F_1 and F_2 are both scheduled to arrive during time 3, and time 4 capacity is unrestricted (i.e., equal or greater to 2), we only need to consider the cost of one period delays:

<u>Flight</u>	<u>Ground Delay Cost</u>	<u>Air Delay Cost</u>
F_1	cg_1 = \$1000	ca_1 = \$2000
F_2	cg_2 = \$1100	ca_2 = \$2200

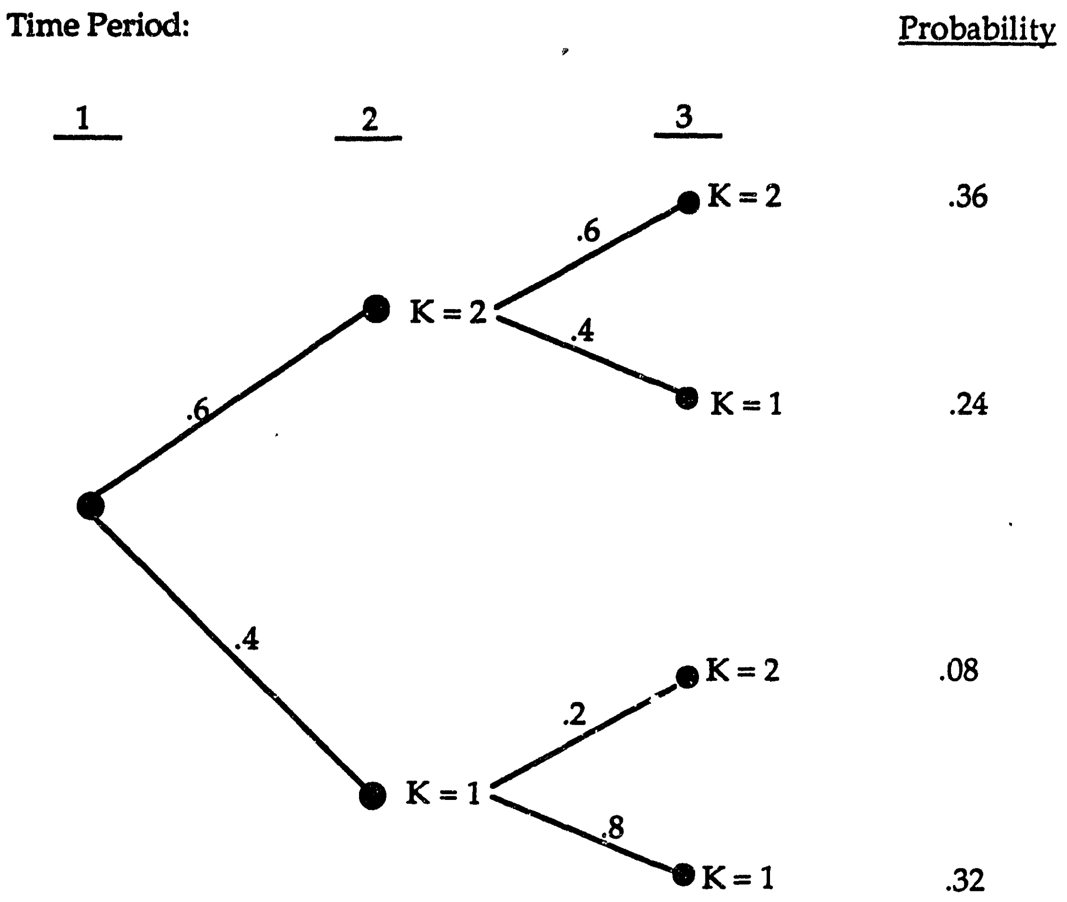


Figure 2.2

PMF of Airport Landing Capacities

In line with what we would expect in a real situation, the cost of air delays is higher than that of ground delays reflecting the higher operational cost of airborne aircraft.[2] Also, the aircraft have different cost structure reflecting factors such as aircraft type, passenger load, fuel efficiency, connection schedules, etc.

The static solution assigns optimal ground-holds at the beginning of time 1. Thus, we cannot use information on the state of the weather (i.e., capacity) during

[2] Air delay cost considers factors such as fuel consumption and risk of operation. It could also include other factors such as noise levels and cost to the environment.

time 2, when making decisions. Since the probability of having capacity limited to one landing during time 3 is .56, the optimal static strategy is to let F_2 depart according to schedule and delay F_1 one time period for an optimal static cost of $1000.

Now we explore the optimal dynamic strategy. In the dynamic problem we make ground-hold decisions on a period by period basis, using the history of airport capacities to produce an update capacity forecast in the form of a conditional PMF of future airport landing capacities based on the original static forecast. Consider the following dynamic strategy:

Let F_1 depart at the beginning of time 1 (i.e., according to schedule). At time 2, delay F_2 departure one time period if time 2 capacity is 1; but let F_2 depart according to schedule if time 2 capacity is 2.

By conditioning on the value of airport capacity at time 2 we see that the expected delay cost of this strategy is $928 (i.e., .6 (.4 x $2200) + .4 x $1100), representing a significant cost improvement vs. the optimal static strategy. We can see that this strategy is also the optimal dynamic strategy.

This example illustrates three important points:

1. A dynamic solution results in an improvement vs. the static solution. An impro'ement is possible even for the case of equal cost structure for both flights. Suppose F_2 delay costs are identical to those of F_1; the strategies above yield $1000 and $880 for the static and dynamic optimal strategies respectively.

2. The dynamic solution results in an improvement vs. the static solution even for the case of a static capacity forecast supplied at the beginning of time 1.

3. An optimal ground-hold strategy may be counter-intuitive even for the simplest problem. In this example the more expensive flight, F_2, is the one subject to ground delays under the dynamic strategy.

Observations 1-3 above point out the advantage of solving the GHPP dynamically and highlights the importance of automating the ground-hold assignment process as intuition or even intelligent heuristics may result in suboptimal ground-holds.

2.2 Problem Inputs, Notation and Algorithm

The network under consideration is the one described in Section 1.4.2 with a single "destinations" airport Z to which N aircraft are scheduled to arrive from the "departures" airports. Assumptions for the dynamic GHPP are as follows:

1. Congestion at airport Z is the only source of delays. Departure and travel times are deterministic and known in advance.
2. A landing rule is specified. This rule may be static, such as the fixed landing priority rule used by Terrab's static algorithm, or dynamic such as: FCFS, landing according to decreasing marginal cost or any other dynamic landing scheme.
3. The time interval of interest, [0, L], starts with the earliest departure time and ends with the latest scheduled arrival time, and is discretized into T periods of equal duration (1, . . . , T). Capacity for each period becomes known at the beginning of the period. Period T+1 capacity is infinite (i.e., adequate to land any aircraft unable to land during 1, . . . , T). This assumption is needed to limit the time horizon for the problem. It is a realistic assumption since traffic declines significantly during the last hours of daily operation.

2.2.1 Problem Inputs

Since capacity is unrestricted at the departure airports we do not need to keep information on the airport of departure; thus we refer to the N flights as: $F_1, \ldots, F_N$ (In Section 2.3, where we develop a dynamic algorithm that considers capacity constraints at the departure airports, we will need to consider the airport of departure when making ground-hold decisions on flights). Based on this observation we see that inputs needed for the problem are as follows:

- L_i, A_i:	scheduled departure and arrival times for each flight F_i.
- $c_{gi}(j)$:	ground marginal cost of delaying F_i for the jth time period.
- $c_{ai}(j)$:	air marginal cost of delaying F_i for the jth time period on the air.[3]
- $P_{K_1K_2\ldots K_T}$:	The static joint PMF of airport Z landing capacities.[4]

2.2.2 Notation, Dynamic Programming Recursion, and Algorithm

The dynamic programming recursion presented here is based on the principle of optimality. The reader is referred to the book *Dynamic Programming* by Dimitri Bertsekas [13] for a proof of the dynamic programming recursion and a good reference on the subject. Before presenting the DP recursion, we establish the following notation.

- **Stage Variable**: stage i is the beginning of time period i.
  Stages are: $1, 2, \ldots, T + 1$.

[3] In order to facilitate notation, the cost of air delay for F_i is considered independent of ground-holds previous to departure. This may not be the case in practice. Notation is easily modified to incorporate a two dimensional air cost function which considers ground-holds previous to departure.

[4] Allowing for time dependency (i.e., $P_{K_1\ldots K_T}(t)$) is possible. An optimal dynamic solution for this case can be found as explained in section 2.2.3.

- **State Variable:** for stage i, the state variable $X_i = \{K_1, \ldots, K_i; D_i^1, \ldots, D_i^N\}$ is a vector that fully reflects the state of the system at stage i. $K_1, \ldots, K_i$ are the airport capacities up to stage i, and $D_i^1, \ldots, D_i^N$ are the ground delays for each flight up to, but excluding, period i delays.

- **Decision Variable:** for each stage i and state X_i, the decision variable $u_i = \{u_i^1, \ldots, u_i^N\}$ is an 0–1 vector:

  $u_i^j = 1$ if F_j is delayed on the ground during period i, and 0 otherwise. Notice that $u_i^j = 0$ for non-eligible flights.

- $g_i(X_i, u_i)$: The ground plus air cost from stage i to stage i+1, given state X_i and decision u_i (i.e., this is the ground plus air cost for period i given state X_i and decision u_i).

- $J_i(X_i)$: Optimal value of delay costs for stages i through T+1 given state X_i.

- $p_{K_{i+1}/K_1 \ldots K_i}$: Probability of having airport landing capacity K_{i+1} during period i+1 given capacities $K_1, \ldots, K_i$ during periods 1, . . . , i respectively. This is a conditional PMF that can be obtained from the joint PMF $P_{K_1, \ldots, K_T}$.

Using the principle of optimality we arrive at the following DP recursion:

$$J_i(X_i) = \underset{u_i}{\mathrm{Min}} \left\{ g_i\,(X_i, u_i) + \sum_{K_{i+1}} \left\{ p_{K_{i+1}/K_1 \ldots K_i}\, J_{i+1}\,[X_{i+1}\,(X_i, u_i)] \right\} \right\} \qquad (2.1)$$

This dynamic programming recursion gives rise to the following algorithm (superscript "*" denotes the optimal ground-hold strategy):

```
Step 0:  INPUT:

         Input the joint PMF of airport capacities: $P_{K_1 \ldots K_T}$
         For each $F_i$ input, $L_i$, $A_i$, $c_{gi}$, $c_{ai}$.

Step 1:  INITIALIZE:

         Let $i = T, J_{T+1}(X_{T+1}) = g_{T+1}(X_{T+1}) = 0$ for all $X_{T+1}$.

Step 2:  For each state $X_i$ find the optimal ground-hold strategy $u_i^*$ and the
         optimal cost $J_i(X_i)$:

         -  Let:
            $u_i^* = 0;\ J_i(X_i) = \{g_i(X_i, 0) + \sum_{k_{i+1}} \{p_{k_{i+1}/k_1 \ldots k_i} J_{i+1}[X_{i+1}(X_i, 0)]\}\}$

         -  For every feasible $u_i$:
            if $\{g_i(X_i, u_i) + \sum_{k_{i+1}} \{p_{k_{i+1}/k_1 \ldots k_i} J_{i+1}[X_{i+1}(X_i, u_i)]\}\} \le J_i(X_i)$;
            Then:
               $u_i^*(X_i) = u_i;\ J_i(X_i) = \{g_i(X_i, u_i) + \sum_{k_{i+1}} \{p_{k_{i+1}/k_1 \ldots k_i} J_{i+1}[X_{i+1}(X_i, u_i)]\}\}$

Step 3:  TERMINATION CHECK:
         If $i = 1$ go to 4; else: $i = i-1$, go to 2.

Step 4:  Find the optimal dynamic policy and cost:
         For each stage $i = 1, \ldots, T$; given state $X_i$ retrieve: $u_i^*(X_i)$, $J_i(X_i)$.
```

2.2.3 Observations on the Algorithm

(i) We need to find the state space previous to the execution of Step 2. To generate the state space we notice that for state $X_i = \{K_1, \dots, K_i; D_i^1, D_i^2, \dots, D_i^N\}$, given the capacity for stage i+1, K_{i+1}, and the ground-hold decision for stage i, $u_i = \{u_i^1, u_i^2, \dots, u_i^N\}$, we can find state X_{i+1}:

$$X_{i+1} = \{K_1, \dots, K_i, K_{i+1};\ D_i^1 + u_i^1, D_i^2 + u_i^2, \dots, D_i^N + u_i^N\} \tag{2.2}$$

(ii) The one period cost $g_i(X_i, u_i)$ has a ground component and an air component calculated as follows:

$$- \quad \text{ground-cost of } g_i\,(X_i, u_i) = \sum_{j=1}^{N} u_i^j\, c_{gi}\,(D_i^j + u_i^j) \tag{2.3}$$

$$- \quad \text{air-cost of } g_i(X_i, u_i) = \sum_{\substack{\text{planes } F_j \text{ waiting} \\ \text{on air during } i}} c_{aj} \text{ (time of air delay for } F_j\text{).} \tag{2.4}$$

Both costs components of $g_i(X_i, u_i)$ are deterministic. (2.3) is simple to calculate. To implement calculation of (2.4), we notice that, given X_i, we can determine the time of arrival at airport Z for each aircraft. Then, by implementing a queue managed according to the landing rule specified, we can determine which aircraft land each period, which aircraft wait in the air during period i, and for how long each one of these aircraft has been waiting to land.

(iii) The terminal cost, $J_{T+1}(X_{T+1}) = g_{T+1}(X_{T+1}) = 0$ since the capacity of period T+1 is infinity.

(iv) We can find the dynamic solution to the dynamic problem (i.e., for a time dependent PMF, $P_{K_1 \ldots K_p}(t)$) by applying the algorithm at the beginning of each time period t at which the PMF is updated, with $P_{K_1 \ldots K_T}(t)$ and the current state, X_t, as inputs.

2.3 Dynamic Solution to the GHPP with Limited Departure Capacity

Before dealing with implementation of the algorithm developed in Section 2.2, we will discuss how the dynamic programming recursion is easily modified to cover the case of limited capacity at the departure airports, in addition to limited capacity at the arrivals airport Z.

Suppose there are L departure airports with limited departure capacities. Then, the airports of departure for each flight need to be identified. We now use a double subscript to identify flights: F_{li} indicates that flight i is scheduled to depart from airport l, with $l \in \{1, \ldots, L, L+1\}$; $l = L+1$ indicates the flight departs from an airport with unconstrained capacity. Also, the inputs for the problem become: L_{li}, A_{li}, $Cg_{li}(j)$, $Ca_{li}(j)$ for every flight F_{li}; and, the joint PMF of airport capacities is now $P_{\overline{K}_1 \ldots \overline{K}_T}$, where $\overline{K}_i = \left\{K_i^1, \ldots K_i^L, K_i^Z\right\}$ is a vector with K_i^l denoting the period i departure capacity for airport l, $l \in \{1, \ldots L\}$; andK_i^Z denoting airport Z landing capacity during period i.

2.3.1 Notation and DP Recursion

The notation presented in Section 2.2.2 is modified as follows:

- The state variable is $X_i = \left\{\overline{K}_1, \ldots, \overline{K}_i; D_i^{11}, \ldots, D_i^{lj}, \ldots D_i^{L+1N}\right\}$, where $\overline{K}_i = \left\{K_i^1, \ldots K_i^L, K_i^Z\right\}$, is the capacity vector described above; and, D_i^{lj} is the ground delay for each flight up to but excluding period i ground delays.

- The decision variable is $u_i \left\{u_i^{11}, \ldots, u_i^{lj}, \ldots, u_i^{L+1N}\right\}$ a 0-1 vector with $u_i^{lj} = 1$ if F_{lj} is delayed on the ground, 0 otherwise. Notice that$u_i^{lj} = 0$ for non-eligible flights, and the control space u_i is restricted further due to the departure capacity constraint. Specifically, $u_i \in S_i$, where S_i is the set of feasible controls u_i given state X_i, which now takes into consideration limits on the number of aircraft capable of departing from the airports with limited capacity.

- Writing the dynamic programming recursion for the limited departure capacity case is now straightforward.

$$J_i(X_i) = \underset{u_i \in S_i}{\text{Min}} \left\{ g_i\,(X_i, u_i) + \sum_{\overline{K}_{i+1}} \left\{ p_{\overline{K}_{i+1} / \overline{K}_1 \ldots \overline{K}_i} J_{i+1}\left[X_{i+1}\,(X_i, u_i)\right]\right\} \right\} \qquad (2.5)$$

Notice that the DP recursion above is analogous to (2.1). However, we see that the computational complexity and size of the state space increases exponentially with the number of capacity constrained airports. As we will explain in Section 2.4., practicality of the algorithm, even for the case of capacity constrained at the arrivals airport Z only, is limited to solution of small problems. Therefore, we will not present the algorithm that follows from equation (2.5). However, the analysis presented in this section provides valuable insights for dealing with the network-wide problem.

2.4. Algorithm Implementation and Complexity

The algorithm of Section 2.2 was implemented on the CRAY-2 supercomputer facilities at MIT, using the C programming language . Aside from speed performance, the CRAY-2 supercomputer features 256 million words of RAM memory (one word equals 8 bytes).

The size of the problem that could be solved was limited by the strong exponentiality of the algorithm. If we let N be the total number of flights, T the number of time periods, and M the number of possible capacities for any given time period; we see that the complexity of the state space size is $O(M^T T^{N+1})$. This is because the complexity of $P_{K_1 \ldots K_T}$ is $O(M^T)$, the complexity of the ground delays for period T is $O(T^N)$ and we have T periods.

The computational complexity of the algorithm is determined by the computational complexity of the state space and the computational complexity of step 2 of the algorithm ($O(N2^N)$ since, for a given state, X_i, we must find the ground-hold strategy, u_i, that minimizes expected cost). Thus, the computational complexity of the algorithm is $O(N\ M^T\ 2^N T^{N+1})$.

The maximum problem size that we were able to solve on the CRAY-2, with no further restriction on inputs, was 5 (i.e., N, M, $T \leq 5$), for which the cardinality of the state space was 2,087,260. Table 2.1 below shows the exponential increase in state space size with problem size.

Problem size	
(N, M, T ≤ Problem Size)	State Space Size
2	12
3	464
4	34,105
5	2,087,260

Table 2.1

State Space Size For DP Solution

The program developed can handle larger parameters when the PMF of airport capacities is sparse. However, notice that we must still consider $O(2^N)$ ground-hold strategies at each stage as we exercise ground hold control on individual planes. We see that the modeling approach of this chapter would be impractical for real problems for which the number of flights can be in the order of one thousand per day.

The C program developed takes inputs from a file with format as specified in Appendix 1. The landing rule implemented lands aircraft according to decreasing priority numbering of flights. However, dynamic landing rules are easy to implement. Appendix 1 also lists the C program code for the size 5 problem and a sample run for the size 5 problem[5].

The program outputs the optimal ground-hold strategy, $u_1^*(X_1)$, and the optimal cost, $J_1(X_1)$ for period 1 and asks if we want to obtain results for the next period. If yes, the program requests to input the period's capacity K_i; outputs $u_i^*(X_i)$, $J_i(X_i)$ and asks if we want to try another period. When period T is reached, the program asks if we want to try another capacity sample path starting with period 1. As we can see, the program generates an optimal ground hold policy for any feasible capacity sample path. Thus we are able to answer "what if" questions. Worth mentioning, if the initial capacity forecast gets updated at any period i the problem must be solved again using the updated capacity forecast and information on the current state.

[5] Notice that the size of the program code grows exponentially with problem size as for N planes each of the 2^N possible ground-hold strategies must be implemented.

2.5 Simplifying the Model

From the analysis in Section 2.4 we see that the sources of complexity in the dynamic programming algorithm are:

1. The number of time periods.
2. The number of capacity sample paths.
3. The number of aircraft.

The number of time periods could be reduced by increasing the duration of the discrete time intervals. However, this does not have a significant impact on complexity (notice that doubling of the time interval duration to 30 minutes reduces the number of time periods from 60 to 30). Therefore, new algorithms should concentrate on reducing the complexity of items 2 and 3 above.

In regards to the number of capacity scenarios, current state of the art weather forecasting limits the capacity forecast for airport Z to a few capacity cases. Thus, we can greatly simplify the PMF for airport landing capacities without affecting the quality of the solution.

Regarding the number of aircraft, modeling approaches that control groundholds by taking decisions on groups of planes should have a significant impact on reducing computational complexity.

Chapters 3 and 4 present algorithms for the static and dynamic probabilistic GHPP that can solve practical instances of the problem even for the largest airports in the US ATC network. This is made possible by using capacity forecasts limited to few capacity cases and grouping planes into at most three cost categories.

CHAPTER 3

3. STATIC STOCHASTIC PROGRAMMING SOLUTION TO THE GHPP

As discussed in Section 1.5.4, Terrab solved the multi-period static probabilistic GHPP through a dynamic programming formulation which uses flights ordered according to a fixed priority rule as the stage variable. The computational complexity of the algorithm is $0(NC(T+1)^2 M^T)$, where C is the number of capacity cases in the forecast, M is the maximum number of possible capacity values for a single time period, and T is the number of time periods in the time interval of interest. The complexity of the algorithm is exponential in the number of time periods T and we see that reducing the number of capacity cases has only a small impact on improving computational complexity.

Since our ultimate goal is to solve problems for the largest airports in the US ATC network, we need to consider alternatives that reduce computational complexity. One possibility is the development of heuristics based on the optimal algorithm, some of which were reviewed in Section 1.5. The major disadvantages with this approach are: (i) some of the heuristics are themselves exponential in complexity (e.g., the limited look-ahead heuristic); (ii) performance of the heuristic may not be adequate under conditions such as significant congestion (e.g., the limited look-ahead heuristic which solves each subproblem assuming unlimited capacity for future time periods); (iii) it is difficult to assure a performance warranty (i.e., an acceptable deviation from optimality). An alternative is trying to simplify the model of the problem so that: (i) the model still captures key features of the problem needed to assure good solutions, (ii) the model yields algorithms that are solvable for practical instances of the problem without having to resort to heuristics.

In this chapter, we choose the later alternative in trying to simplify the complexity of the algorithm used to solve the static GHPP. We present a model that can be solved using stochastic linear programming with one stage. Our formulation allows for airport capacity forecasts with up to three capacity scenarios, in line with current forecasting technology. Also up to three different aircraft classes, with their associated ground-hold cost functions, are possible. The formulation proposed here yields linear programs of size solvable on a personal computer even for the largest airports in the US ATC network.

The chapter is structured as follows: Section 3.1 presents key assumptions in developing the simplified problem, inputs to the model and decision variables. In Section 3.2, we develop the stochastic programming formulation, starting from a deterministic linear programming model for a single class of aircraft. Section 3.3 discusses how to incorporate into the model constraints such as maximum ground-hold delay allowed, limits on air delay for particular times of the day; and the possibility of choosing between "conservative" and "liberal" ground-hold policies by varying a single parameter in the model. Section 3.4 presents a decomposition algorithm that exploits the special structure of the constraint matrix. In Section 3.5, we extend our formulation to include up to three classes of aircraft; followed by an analysis of constraint matrix size in Section 3.6, confirming the possibility of solving practical size problems on a personal computer.

3.1 Model Assumptions, Inputs, and Decision Variables

Some of the assumptions in this model are similar to those presented when discussing the dynamic case in Section 2.2. For completeness, we will present here a full set of assumptions and the rationale behind them.

3.1.1 The Basic Network and Sources of Congestion

The network under consideration is the one discussed in Section 1.4.2 with a single "destination" airport Z (i.e., star configuration) to which a total of NTOT aircraft are scheduled to arrive from the "departures" airports. Congestion at airport Z is the only source of delays. Departure and travel times are deterministic and known in advance.

3.1.2 Discretization of Time

The time interval of interest [0, L] comprises a full day of operations at airport Z, starts with the earliest departure time and ends with the latest scheduled arrival time. The interval [0, L] is discretized into T periods of equal duration. Typical period durations are in the 15-20 minutes range resulting in total number of periods, T, in the 50-70 range.

3.1.3 Classification of Aircraft

Aircraft are classified into three classes: small (S), large (L) and Heavy (H); denoted classes 1, 2, and 3 respectively. This is in line with ATC practice in the US which defines the three classes as: Small Aircraft: less than 12,500 pounds MTOW (maximum take-off weight); Large Aircraft: between 12,500 and 300,000 pounds MTOW; and Heavy Aircraft: over 300,000 pounds MTOW.

3.1.4 Input Schedule for Daily Operations

Since we are considering the static problem, we are able to ignore departure times, defining the input schedule as:

(i) For a single aircraft class:

N_i: the number of aircraft scheduled to arrive at airport Z during period i. $i = 1, \ldots, T$.

(ii) For three aircraft classes:

N_{ki}: The number of aircraft of class k scheduled to arrive at airport Z during period i. $k = 1, 2, 3; \quad i = 1, \ldots, T$.

For (i) we have $\sum_{i=1}^{T} N_i = NTOT$; and for (ii) we have $\sum_{k=1}^{3} \sum_{i=1}^{T} N_{ki} = NTOT$.

3.1.5 The Cost of Ground-Delays

We allow for arbitrary ground-hold cost functions. Thus we have:

(i) For a single aircraft class:

$C_g(i)$: cost of delaying one aircraft for i periods on the ground. $i = 1, \ldots, T-1$

(ii) For three aircraft classes:

$C_g(k, i)$: Cost of delaying one aircraft of class k for i periods on the ground. $k = 1, 2, 3; \quad i = 1, \ldots, T-1$.

Notice that these costs are total costs of delay for one aircraft during i periods and not the marginal cost for the i^{th} period.

3.1.6 The Cost of Air Delays

Since ATC lands aircraft in a sequence that approximates FCFS, we see that within reasonable levels of air delay (i.e., maximum air delays in the 30 to 45 minute range) the marginal cost of air delay can be considered constant for each class of aircraft. Thus, we assume that the cost of delaying one aircraft of any class one time period in the air is a constant, c_a, equal to the weighed average of the marginal air delay costs for each class of aircraft, with weights equal to the proportion of each aircraft class in the schedule. In general we set c_a so that $c_a > C_g(k, i) - C_g(k, i-1)$ for all classes of aircraft (i.e. $k = 1, 2, 3$) and $i \leq T$; in order to assure that marginal air delay costs are higher than marginal ground-hold costs for all aircraft.

Once we develop the stochastic linear programming model, we will see that we can incorporate constraints assuring that the maximum air delay criterion mentioned above is met. Also, it will become clear that the relative difference between ground and air delay costs affects ground-hold policy more significantly than modeling air delay costs in greater detail.

3.1.7 Capacity Forecast

Let airport Z landing capacities be $K_1, \ldots, K_T$ for periods $1, \ldots, T$ respectively. We limit our attention to forecasts with three capacity scenarios. Thus, the forecast for airport Z landing capacity is of the form:

$$K_{q_1}, \ldots K_{q_T}; \text{ with associated probability } p_q \text{ for } q = 1, 2, 3;$$

and we let $K_{q_{T+1}} = \text{NTOT}$ (for $q = 1, 2, 3$); in order to assure that all aircraft are able to land at Z within T+1 periods. Limiting the time horizon for the problem to T+1 time periods is reasonable since air traffic declines significantly towards the end of the day, avoiding extreme congestion at airport Z during T+1, even under cases of low capacity. Important to discuss is the assumption regarding the number of capacity scenarios in our forecast.

Currently, the assignment of ground-holds is based on a deterministic capacity forecast, ignoring the effect of weather uncertainties on landing capacity. At the beginning of daily operations, each one of the major airports in the US provides the CFCF with a deterministic landing capacity profile which reflects expected weather conditions for the day. CFCF then proceeds to assign ground-holds as described in section 1.5.1.

The probabilistic forecasting system we are proposing represents an improvement versus a deterministic forecast. However, we need to justify limiting the forecast to three capacity scenarios. Our argument below is based on the fact that the predictive accuracy of weather forecasts has improved to the point that the type/severity of weather conditions on a geographical area can be forecast with reasonable accuracy; however, the timing of weather fronts is uncertain.

Suppose we are told that a weather front that will reduce landing capacity by 50% for the rest of the day will reach the vicinity of airport Z between 12:00 and 17:00 hours. With no further information on the weather front, we can assume that the weather front is equally likely to reach Z at any time during this time period. Given the discrete nature of our model, we would then originate a capacity forecast consisting of scenarios showing a 50% reduction in landing capacity starting at each of the periods P comprised in the interval 12:00 to 17:00 hours, each with probability 1/P.

Based on the macroscopic nature of our model, we see that hourly intervals are adequate to approximate the uniform distribution. Since we are limiting our forecast to three capacity cases, we would originate instead a capacity forecast consisting of three capacity scenarios, each with equal probability and airport landing capacity reduced by 50% starting at around 12:00, 14:30 and 17:00 respectively. Naturally, the closer we approximate the uniform distribution for the arrival time of the weather front, the more accurate our model will be. The point we are trying to make here is that reasonably good capacity forecasts for airport Z comprise few rather than many capacity scenarios.

Throughout this thesis we limit our attention to forecasts consisting of three capacity scenarios, keeping in mind that solving models that require inclusion of a few more capacity cases, is within the capacity of a computer work- station if not a personal computer.

3.1.8 Model Variables

Finally, we describe the variables in our model. The ground-hold decision variable is:

(i) **For one class of aircraft:**

X_{qij}: The number of aircraft originally scheduled to arrive at Z during period i which are now rescheduled to arrive during j (i.e., following a ground delay of j – i time periods) under capacity case q. $q = 1, 2, 3;\quad i = 1, \ldots, T; i \leq j \leq T+1$.

Notice that for each q and i we have: $\sum_{j=i}^{T+1} X_{qij} = N_i$; where N_i is the input schedule defined in Section 3.1.4.

(ii) **For three classes of aircraft:**

X_{qkij}: The number of aircraft of class k originally scheduled to arrive at Z during period i, and rescheduled to arrive during j under capacity case q, due to a ground delay of j–i time periods. $q = 1, 2, 3;\quad k = 1, 2, 3;\quad i = 1, \ldots, T;\quad i \leq j \leq T+1$.

In this case we notice that for each q, k, and i we have: $\sum_{j=i}^{T+1} X_{qkij} = N_{ki}$; where N_{ki} is the input schedule for three classes of aircraft.

Another set of variables is defined by the airborne queueing process at airport Z. Since the marginal cost of air delays is constant and equal for all aircraft, we only need to consider the number of aircraft that are unable to land during any period. Thus we define:

W_{qi}: The number of aircraft unable to land at airport Z during period i under capacity case q (i.e., the number of aircraft incurring airborne delay during period i). $q = 1, 2, 3;\ i = 1, \ldots, T$.

3.2 Formulating the Stochastic Programming Model

In this Section, we formulate the static GHPP as a stochastic linear programming problem with one stage. We follow the framework presented in the book, "*Principles of Operations Research,*" by H. M. Wagner [14], which provides a good introduction to stochastic programming. Here we limit our attention to the case of a single class of aircraft, and extend the formulation to cover three classes of aircraft in Section 3.4.

3.2.1 Formulate the Deterministic Problem

Suppose capacity case q, with capacities $K_{q1}, \ldots, K_{qT+1}$, occurs with probability one. Since the optimization criterion is to minimize total delay cost (ground plus air) the objective function is:

$$\text{Minimize:} \quad \sum_{i=1}^{T} \sum_{j=i+1}^{T+1} C_{g(j-i)} X_{qij} + \sum_{i=1}^{T} W_{qi}\, c_a$$

Subject to:

(i) All aircraft scheduled to land during i must be rescheduled to arrive during i, i+1,. . ., T+1:

$$\sum_{j=i}^{T+1} X_{qij} = N_i \qquad i = 1, 2, \ldots, T$$

(ii) The flow balance at airport Z at the end of each period yields:

$$W_{qi} \geq \sum_{j=1}^{i} X_{qji} + W_{qi-1} - K_{qi} \qquad i = 1, \ldots, T+1;$$

(with $W_{qo} = W_{qT+1} = 0$)

(iii) X_{qij}, $W_{qi} \geq 0$ and integer.

The formulation above is an integer programming problem with linear cost function. In the next Section we show that for the deterministic problem the constraint set has a network structure, reducing the problem to a minimum cost flow problem for which the integrality constraint can be relaxed. Also, notice how constraints (ii) model the queueing process at airport Z:

If the RHS of (ii) is less than or equal to zero (i.e., capacity K_{qi} is adequate to land all aircraft waiting to land during i), then $W_{qi} = 0$, by constraint (iii) and the positive cost coefficient c_a for W_{qi} in the objective function.

Both the linearity of the objective function and the simplicity of the model result from the assumption on the cost of air delays in Section 3.1.6.

3.2.2 Reduction to Minimum Cost Flow in an Uncapacitated Network

Figure 3.1 shows how the formulation of Section 3.2.1 is transformed to a minimum cost flow problem in an uncapacitated network (for clarity, subscript q has been omitted from all variables in Figure 3.1). An additional node S with supply $\sum_{i=1}^{T} K_{qi}$, and arcs S_{qi} i = 1, . . ., T+1, is introduced. The resulting problem has the same objective function, since the flows on S_{qi} arcs have zero cost, and the following constraint set:

1. **Supply nodes:**

(i) $X_{qii} + X_{qii+1} \cdots + X_{qiT+1} = N_i$; $i = 1, \ldots, T.$

(ii) $\sum S_{qi} = \sum_{j=1}^{T} K_{qi};$

2. **Demand nodes:**

(iii) $W_{qi} - \left(W_{qi-1} + \sum_{j=1}^{i} X_{qji} + S_{qi} \right) = -K_{qi}; \quad i = 1, \ldots, T+1,$

(with $W_{q0} = W_{qT+1} = 0$)

(iv) $X_{qij}, W_{qi}, S_{qi} \geq 0.$

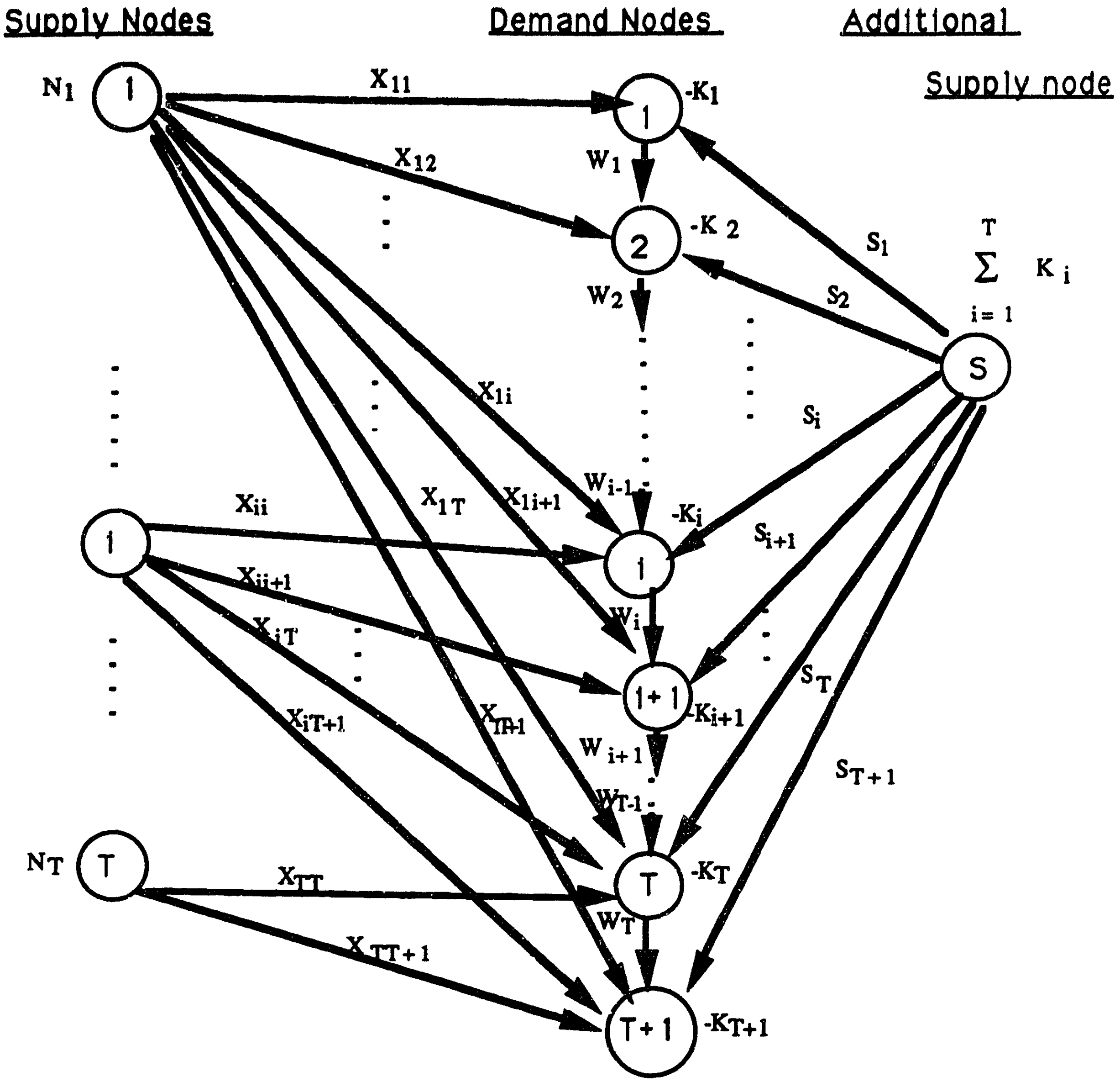


Figure 3.1

Network Diagram for Minimum Cost Flow Problem

Notice that the constraint matrix for the network formulation is totally unimodular and supply/demands are integer. Thus, we relax the integrality constraint in (iv). The deterministic formulation provides an interesting interpretation of the current CFCF practice for the assignment of ground-holds.

Suppose we have constant marginal ground and air costs, c_g and c_a respectively, with $c_g < c_a$. Then the optimal solution to the minimum cost flow problem above has $W_i=0$ for $i = 1, \ldots, T$ (since the cost on the vertical arcs, c_a, is greater than c_g). X_{ij} flows are constructed by assigning available capacity according to earliest scheduled arrival time at airport Z.

3.2.3 Formulate the Distribution Problem

In solving the distribution problem we consider the probabilistic nature of airport capacities but assume that airport capacities for all periods, $K_{q1}, \ldots, K_{qT}$, become known before we make the ground-hold decisions. The optimal solution to the distribution problem is a complete policy consisting of optimal ground-holds X^*_{qij} with associated W^*_{qij} variables for each one of the capacity scenarios $q = 1, 2, 3$. The distribution problem becomes easy to understand once we present its formulation:

$$\text{Minimize} \quad \sum_{q=1}^{3} p_q \left\{ \sum_{i=1}^{T} \sum_{j=i+1}^{T+1} C_g (j - i) X_{qij} + c_a \sum_{i=1}^{T} W_{qi} \right\}$$

Subject to:

For each $q = 1, 2, 3$:

(i) $X_{qii} + X_{qii+1} \ldots X_{qiT+1} = N_i; \quad i = 1, 2, \ldots, T+1$

(ii) $\sum_{i=1}^{T+1} S_{qi} = \sum_{i=1}^{T} K_{qi}$;

(iii) $W_{qi} - \left(W_{qi-1} + \sum_{j=1}^{i} X_{qji} + S_{qi} \right) = -K_{qi}; \quad i = 1, \ldots, T+1$

(with $W_{q0} = W_{qT+1} = 0$)

(iv) $X_{qij}, W_{qi}, S_{qi} \geq 0.$

We see that the solution to the distribution problem is equivalent to solving three separate minimum cost flow problems (i.e., one for each q = 1, 2, 3), since the constraint matrix above consists of three separate network components. Unfortunately we must assign ground-hold decisions before knowing airport capacities. Thus, we need to modify the distribution problem formulation.

3.2.4 Formulate the Static GHPP

In the static solution to the GHPP, we make ground-hold decisions at the beginning of period one, before knowing airport capacities. Since we make a single set of ground-holding decisions for the day, we see that we need to introduce the following set of constraints in the formulation above:

(v) $X_{1ij} = X_{2ij} = X_{3ij}; \ i = 1, \ldots, T; \ i \leq j \leq T+1$

Figure 3.2 shows that after adding these constraints the structure of the constraint matrix becomes block angular. Elements outside the rectangles are equal to zero.

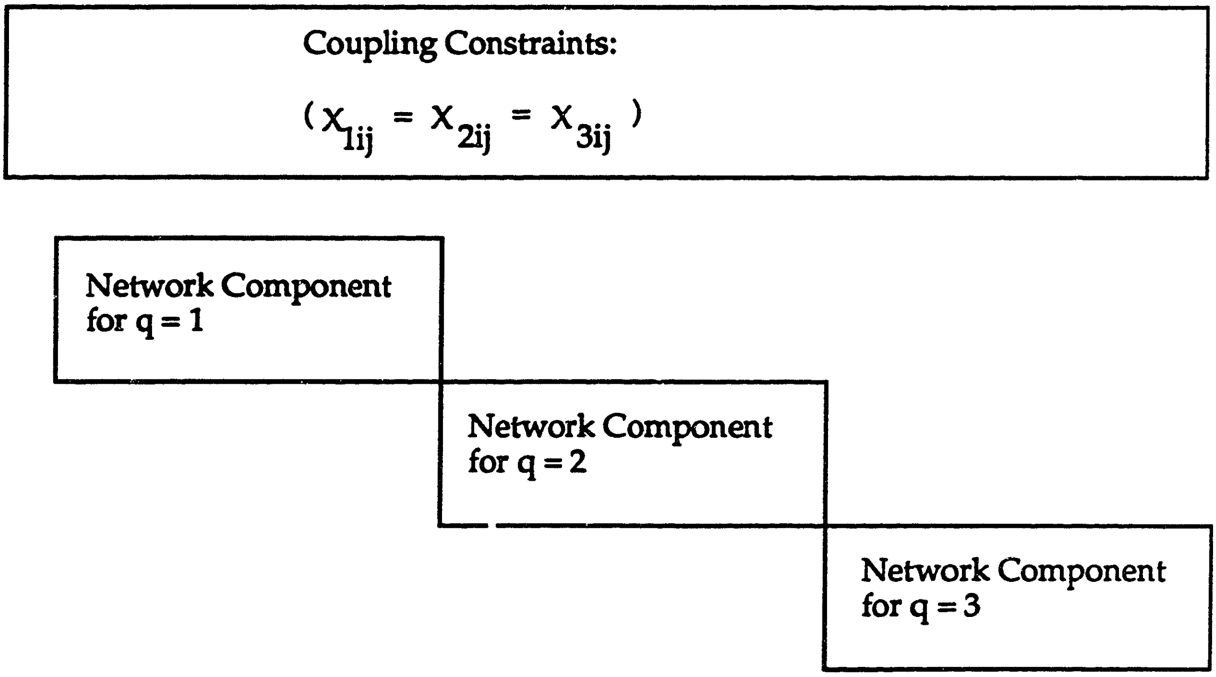

Coupling Constraints:
$(X_{1ij} = X_{2ij} = X_{3ij})$
Network Component for q = 1
Network Component for q = 2
Network Component for q = 3

Figure 3.2
Constraint Matrix Structure

In Section 3.4 we will present a decomposition algorithm that exploits the special structure of the constraint matrix. Worth noticing, the network structure of the constraint matrix is lost since variable X_{qij} now appears with additional +1 or –1 coefficients in the equations of type (v). We could still have integer solutions if the constraint matrix were unimodular. This is because total unimodularity (i.e., network structure) is a sufficient condition, while unimodularity is a necessary condition for integrality of basic feasible solutions to linear programming relaxations of integer programming problems with integer RHS.

Since the solutions to all the practical problems solved in Chapter 5 are integer we tried to prove unimodularity of the constraint matrix. We were not successful in either proving unimodularity (which requires verifying that the inverse, B^{-1}, for all basis matrices B of the constraint matrix are integer) or generating a counterexample with non-integer solutions.

Although unimodularity of the constraint matrix remains an open question, this is not critical to the results in this thesis. This is because, even if the solutions were not integer, rounding would produce good practical solutions because the ground-hold decision variables are not limited to 0-1 values.

3.3. Restating the Model and Introducing Important Modeling Constraints

Before proceeding further, we restate the stochastic programming formulation for the static GHPP. After substituting the constraint (v), introduced in Section 3.2.3, into the formulation for the distribution problem we obtain:

$$\text{Minimize} \quad \sum_{i=1}^{T} \sum_{j=i+1}^{T+1} C_g\,(j-i)\,X_{ij} + c_a \left\{ \sum_{q=1}^{3} p_q \sum_{i=1}^{T} W_{qi} \right\}$$

Subject to :

(i) $X_{ii} + X_{ii+1} \ldots + X_{iT+1} = N_i; \quad i = 1, \ldots, T$

For each q:

(ii) $$\sum_{i=1}^{T+1} S_{qi} = \sum_{i=1}^{T} K_{qi};$$

(iii) $$W_{qi} - W_{qi-1} - \sum_{j=1}^{i} X_{ji} - S_{qi} = -K_{qi} \quad i = 1, \ldots, T+1;$$

(with $W_{q0} = W_{qT+1} = 0$)

(iv) $X_{ij}, W_{qi}, S_{qi} \geq 0$, and integer.

We have omitted the subscript for the capacity case, q, from ground-hold variables X_{qij}, since we make a unique set of ground-hold decisions (i.e., $X_{1ij} = X_{2ij} = X_{3ij}$).

3.3.1 Important Model Features

Objective Function Parametric Programming

By examining the objective function we see that by changing a single parameter, c_a, we can adjust the bias of the model towards conservative (liberal) ground-holding policies. A higher value for c_a will result in a greater emphasis on ground-holds since ground delays become less expensive vis-a-vis air delays. Conversely a lower c_a will result in more liberal ground-hold times. We will illustrate this point in Chapter 5. Notice also that it is possible to let the marginal air delay cost depend on the time of the day.

Introducing Additional Modeling Constraints

We see that planners using our model will be interested in more than just minimizing expected costs. They may, for instance, be interested in safety considerations which limit the degree of congestion at the arrivals airport. Our model easily allows for constraints of this type. For example, suppose the duration of the discrete time intervals is 15 minutes, and we want to limit airborne queueing delay under capacity case q to at most 30 minutes at the end of time period i. The corresponding constraint is:

$$w_{qi} \leq K_{qi+1} + K_{qi+2}.$$

Notice how the constraint above can also be interpreted as limiting the queue length at airport Z at the end of period i, under capacity case q, to the total capacity for the next two periods, i+1 and i+2.

Suppose now that we want to limit ground delays during certain period i to at most P periods. Then the constraint in the original model:

$$\sum_{j=1}^{T+1} X_{ij} = N_i, \text{ becomes: } \sum_{j=i}^{i+P} X_{ij} = N_i.$$

Obviously by introducing limits on the duration of airborne queueing delays at Z and ground delays at the departure airports simultaneously, we could generate infeasible problems. If this is the case, we may need to relax some of the constraints on maximum air delays, or even cancel some flights in order to render the problem feasible.

3.4 Decomposition Algorithm

In Section 3.2 we saw that the constraint matrix had primal block angular structure. Due to the simplicity of the coupling constraints in these matrices, in Section 3.3 we eliminated the coupling constraints through substitution into the subproblem constraints. The resulting linear program has a significantly lower number of constraints and variables but lacks the network structure that would make it suitable for solution using faster algorithms. An alternative to this approach is the use of decomposition methods. Despite the higher number of equations and variables, the advantage of decomposing the problem is that the subproblems have network structure.

The algorithm we present is based on Dantzig-Wolfe's Decomposition. A good treatment of the subject can be found in the book *Applied Mathematical Programming* by Bradley, Hax and Magnanti [15].

3.4.1 Step 1: Formulate the Master Problem

For ease of notation, we will focus on the static stochastic programming model for a single class of aircraft presented in Section 3.2. Extending the decomposition procedure to the case of several aircraft classes discussed in Section 3.5, and the dynamic stochastic programming models presented in Chapter 4 is straightforward due to the similarities in the structure of the constraint matrices.

When restating the static GHPP formulation in Section 3.3, we substituted constraints (v) of Section 3.2.4 into the model. If instead, we retain the coupling constraints, the resulting stochastic linear program shown below has primal block angular structure, we refer to it as the "Master problem":

$$\text{Minimize} \quad \sum_{q=1}^{3} p_q \left\{ \sum_{i=1}^{T} \sum_{j=i+1}^{P+1} C_g (j-i) X_{qij} + c_a \sum_{i=1}^{T} W_{qi} \right\} \tag{3.1}$$

Subject to:

- <u>Coupling (Global) Constraints</u>:

(1) $X_{1ij} - X_{2ij} = 0$; $i = 1, \ldots, T$; $i \leq j \leq T + 1$

(2) $X2_{ij} - X_{3ij} = 0$; $i = 1, \ldots, T$; $i \leq j \leq T+1$

- <u>Subproblem Constraints</u>:

For each $q = 1, 2, 3$:

(i) $X_{qii} + X_{qii+1} \ldots X_{qiT+1} = N_i; \quad i = 1, 2, \ldots, T+1$

(ii) $\sum_{i=1}^{T+1} S_{qi} = \sum_{i=1}^{T} K_{qi}$;

(iii) $W_{qi} - \left(W_{qi-1} + \sum_{j=1}^{i} X_{qji} + S_{qi} \right) = -K_{qi}; \quad i = 1, \ldots, T+1$

(with $W_{q0} = W_{qT+1} = 0$)

(iv) $X_{qij}, W_{qi}, S_{qi} \geq 0.$

A feasible solution l to subproblem q is denoted $\{X_q^l, W_q^l, S_q^l\}$, where X_q^l, W_q^l, S_q^l are suitably defined vectors. We see that due to the coupling constrains, l may or may not be feasible for the master problem. Thus we refer to it as a subproblem proposal. In the material that follows we assume that the first proposal from each one of the q subproblems (i.e., $\{X_q^1, W_q^1, S_q^1\}$ for q = 1, 2, 3) is feasible for the master problem. Such first proposals can be generated by finding a basic feasible solution for the master problem using phase one of the simplex method.

3.4.2 Step 2: Formulate and Solve the Restricted Master Problem

The restricted master problem consists in choosing optimal weights for a given set of subproblem proposals within the framework of the master problem objective function and global constraints.

In the restricted master problem, presented below, there are l_q proposals for each q subproblem; denoted $\{X_q^1, W_q^1, S_q^1\}, \ldots, \{X_q^{l_q}, W_q^{l_q}, S_q^{l_q}\}$, for q = 1, 2, 3. Notice that the cardinality of l_q is not necessarily equal for each q. Also, the vector $\lambda_q = \{\lambda_q^1, \ldots, \lambda_q^{l_q}\}$ represents the weights for the proposals from each of the subproblems.

Minimize $\sum_{q=1}^{3}\left\{\sum_{k=1}^{l_q} \lambda_q^k p_q\left(\sum_{i=1}^{T}\sum_{j=i+1}^{T+1} C_g (j-i) X_{qij}^k + c_a \sum_{i=1}^{T} W_{qi}^k\right)\right\}$

by letting $P_q^k = p_q\left(\sum_{i=1}^{T}\sum_{j=i+1}^{T+1} C_g (j-i) X_{qij}^k + c_a \sum_{i=1}^{T} W_{qi}^k\right)$ we obtain:

Minimize $\sum_{q=1}^{3}\sum_{k=1}^{l_q} \lambda_q^k P_{q}^k;$ (3.2)

Subject to:

		Optimal Dual Price:	
(1)	$\sum_{k=1}^{l_1} \lambda_1^k X_{1ij}^k - \sum_{k=1}^{l_2} \lambda_2^k X_{2ij}^k = 0;$	$\pi_{ij}^1;$	$i = 1, \ldots, T;\ i \leq j \leq T+1$
(2)	$\sum_{k=1}^{l_2} \lambda_2^k X_{2ij}^k - \sum_{k=1}^{l_3} \lambda_3^k X_{3ij}^k = 0;$	$\pi_{ij}^2;$	$i = 1, \ldots, T;\ i \leq j \leq T+1$
(3)	$\sum_{k=1}^{l_q} \lambda_q^k = 1;$	$\sigma_q;$	$q = 1, 2, 3.$
(4)	$\lambda_q^k \geq 0$		

Observations on the restricted Master Problem:

1. Notice that λ_q^k is a variable while X_{qij}^k, W_{qi}^k are known data for the restricted master problem. Also, we can interpret P_q^k as the unweighted cost contribution to the objective function of proposal k from subproblem q.

2. We can always obtain the solution to the restricted master problem in terms of the variables in the original problem:

$$(X_q^s, W_q^s, S_q^s) = \sum_{k=1}^{l_q} \lambda_q^k (X_q^k, W_q^k, S_q^k) \text{ for } q = 1, 2, 3.$$

3. Constraints (1) and (2) are satisfied for $\lambda_1^1 = \lambda_2^1 = \lambda_3^1 = 1$ since the first subproblem proposal from each subproblem is feasible for the master problem. Thus, the restricted master problem is feasible. Also, the solution to the restricted master problem satisfies the global constrains in the master problem; and, since this solution is the convex combination of subproblem proposals, it satisfies the subproblem constraints. Thus, the restricted master problem solution is feasible for the master problem.

Having determined optimal weights and optimal dual prices associated with the constraints in the restricted master problem, we consider how to improve the solution by adding new proposals. From linear programming theory, we know that adding a new proposal will improve the solution to the restricted master problem if its reduced cost is strictly negative (since we are dealing with a minimization problem). Let $\{X_q, W_q, S_q\}$ be a new proposal from subproblem q, then the reduced cost for the new proposal is given by[6] :

$$\overline{P_q} = P_q - \sum_{i=1}^{T} \sum_{j=i}^{T+1} \pi_{ij^1}\, r^1_{qij} + \pi_{ij^2}\, r^2_{qij} - \sigma_q \tag{3.3}$$

where:

$$P_q = p_q\left(\sum_{i=1}^{T} \sum_{j=i+1}^{T+1} C_g\,(j-i)\, X_{qij} + c_a \sum_{i=1}^{T} W_{qi} \right);$$

$$r^1_{1ij} = X_{1ij},\quad r^1_{2ij} = -X_{2ij},\quad r^1_{3ij} = 0;\ \text{ and}$$

$$r^2_{1ij} = 0,\quad r^2_{2ij} = X_{2ij},\quad r^2_{3ij} = -X_{3ij}.$$

[6] We recall from linear programming theory that the reduced cost, $\bar{c}_j$, for a new activity j, is given by the equation: $\bar{c}_j = c_j - y^T A_j$; where c_j is the cost coefficient in the objective function, y the optimal dual prices and A_j the "column" of constraint matrix coefficients for the new activity.

Next, we see how to generate new subproblem proposals that improve the solution to the restricted master problem by solving a minimization problem subject to the subproblem constraints.

3.4.3 Step 3: Generate New Subproblem Proposals

To determine whether any new proposal will improve the solution to the restricted master problem, we seek to minimize the reduced cost from equation (3.3) subject to subproblem q constraints. $v_q^{l_q}$ denotes the objective function value for the minimization problem below:

$$v_q^{l_q} = \min \overline{P}_q$$

Subject to:

Constraints for subproblem q from the master formulation.

Notice that the optimization problem above is a minimum cost flow problem in an uncapacitated network. Thus we can solve the problem using faster specialized network algorithms. There are two possible outcomes:

(i) if $v_q^{l_q} < 0$, we can improve the solution to the restricted master problem by adding the optimal solution to the subproblem $\left\{X_q^{l_q+1}, W_q^{l_q+1}, S_q^{l_q+1}\right\}$ as the l_q+1 proposal in the restricted master problem. Then, we solve the restricted master problem with the additional proposal and we repeat the subproblem proposal generation procedure: Step 2.

(ii) If $v_q^{l_q} \geq 0$, for $q = 1, 2, 3$, then no new proposal from the subproblems can improve the solution and the procedure terminates. The optimal solution to the original (i.e., master) problem is then:

$$\left(X_q^*, W_q^*, S_q^*\right) = \sum_{k=1}^{l_q} \lambda_q^{k*} \left(X_q^k, W_q^k, S_q^k\right) \text{ for } q = 1, 2, 3.$$

Where λ_q^{k*} are the optimal weights in the last solution to the restricted master problem.

3.4.4 Computational Considerations

– *Key Decomposition Advantages*

There are several advantages to the decomposition algorithm presented here. By breaking the problem into subproblems the algorithm provides significant computational savings. This is because the computations for linear programs are quite sensitive to the number of constraints, in practice growing proportionally to the cube of the number of constraints. For our problem there is the additional advantage of having subproblems with a network structure which can be solved faster than generic linear programs. Also, since the subproblems can be solved independently, the algorithm lends itself to parallel computation.

– *Approximate Solutions*

In practice, the decomposition algorithm develops a good approximation to the solution relatively quickly and then "tails off" approaching optimality slowly. Thus we can use the following bounds in order to establish termination criteria. At the end of each iteration J we define:

Z^J: optimal objective function value of the restricted master problem at iteration J.

Z^*: optimal objective function value for the master problem.

v_q^J: optimal objective function value of subproblem q at iteration J.

σ_q^J: optimal dual price for "weight" constraint q at iteration J.

We know that $Z^J \geq Z^*$. Also, by weak duality:

$$Z^J - \sum_{q=1}^{3} \sigma_q^J + \sum_{q=1}^{3} v_q^J \leq Z^*.$$

Thus we have:

$$Z^J \geq Z^* \geq Z^J - \sum_{q=1}^{3} \sigma_q^J + \sum_{q=1}^{3} v_q^J \; .$$

These bounds allow us to stop the decomposition procedure when we are reasonably close to optimality.

– *Finite Termination*

We see that the subproblem calculation ensures that the variable introduced into the basis for the restricted master problem has a negative reduced cost. Thus, the optimal solution is reached by solving the restricted master problem a finite number of steps.

– *Resolving The Restricted Master Problem and The Subproblems*

Every time we need to resolve the restricted master problem, after addition of a new proposal, the optimal basis from the previous solution can be used as a starting point. Similarly, the optimal basis for the last time we solved the subproblem can be used to initiate the solution to the subproblem when it is considered next.

– *Dropping Nonbasic Columns*

After many iterations the number of proposals (i.e., columns) in the restricted master problem may become large. Any nonbasic proposal in the current iteration can be dropped for the next iteration. If it is required, it is generated again by the subproblem.

3.5 The Stochastic Programming Model for Three Aircraft Classes

Once we have developed the formulation for the case of a single aircraft class, it is easy to extend the formulation to cover the case of several aircraft classes. The notation for several classes of aircraft was introduced in Section 3.1. Here we present the stochastic programming formulation for the static GHPP with three aircraft classes:

For several aircraft classes, the objective function is to minimize total expected delay costs across aircraft classes thus we have:

1. **Objective Function:**

$$\text{Minimize} \sum_{k=1}^{3} \sum_{i=1}^{T} \sum_{j=i+1}^{T+1} C_g(k, j-i)\, X_{kij} + c_a \left\{ \sum_{q=1}^{3} p_q \sum_{i=1}^{T} W_{qi} \right\}$$

Notice how we now add ground delay costs across aircraft classes. The term accounting for the expected cost of air delays remains unchanged as air delay costs are identical for all aircraft.

2. **Subject to :**

(i) Aircraft of class k scheduled to arrive at Z during period i must be rescheduled to arrive during i, i+1, . . ., T+1:

$$\sum_{j=i}^{T+1} X_{kij} = N_{ki}; \quad k = 1, 2, 3; \quad i = 1, \ldots, T$$

For each q:

(ii) $\sum_{i=1}^{T+1} S_{qij} = \sum_{i=1}^{T} K_{qi};$

(iii) The flow balance at airport Z during period i must account for the different aircraft classes arriving at Z during period i:

$$W_{qi} - W_{qi-1} - \sum_{k=1}^{3} \sum_{j=1}^{i} X_{kji} - S_{qi} = -K_{qi}; \qquad i = 1, \ldots, T+1;$$

(with $W_{qo} = W_{qT+1} = 0$)

(iv) $X_{kij}, W_{qi}, S_{qi} \geq 0$ and integer.

The discussion on modeling constraints of Section 3.3.1 applies here, with the additional possibility of limiting ground-holds on a specific class of aircraft.

Being able to differentiate among aircraft classes in our model allows for a more efficient operation of the ATC system as we take into consideration cost differences among aircraft classes. However, this will result in greater ground delays for aircraft classes with lower costs. Therefore, practical application of our models may be limited to the case of a single aircraft class since currently the FAA has a policy of "equal access" to all users of the ATC network that meet the navigational requirements of any sector in the system.

3.6 Constraint Matrix Size

Here we calculate the size of the constraint matrix for the models of Sections 3.3 (one aircraft class) and 3.5 (three aircraft classes). We assume 15 hours of operations and 15 minute duration for time periods, yielding T=60.

3.6.1 Constraint Matrix Size for One Aircraft Class

1. Number of Variables:

- X_{ij} variables: We have that $i = 1, \ldots, T$ and j is such that $i \leq j \leq T+1$. Thus the number of X_{ij} variables is $(T+2)(T+1)/2 - 1$, yielding 1890 X_{ij} variables.

- S_{qi} variables: $q = 1, 2, 3$ and $i=1, \ldots, T+1$. Thus, the number of S_{qi} variables is 183.

- W_{qi} variables: $q = 1, 2, 3$ and $i=1, \ldots, T$. Thus the number of W_{qi} variables is 180.

From above we see that the total number of variables is 2253.

2. **Number of Constraints:**

- Type (i) constraints: The number of type (i) constraints is T.

- Type (ii) constraints: The number of type (ii) constraints is 3 since there is one type (ii) equation for each q.

- Type (iii) constraints: The number of type (iii) constraints is 3(T+1). Since for each q we perform a flow balance at airport Z for periods $1, \ldots, T+1$.

From above we see that the total number of constraints for T=60 is: 60 + 3 + 3 x 61 = 246. yielding a matrix of size 246 x 2253 for the single aircraft class static problem.

3.6.2 Constraint Matrix Size for Three Classes of Aircraft

1. **Number of Variables**

- Number of X_{kij} variables: The number of X_{kij} variables is $3\{(T+2)(T+1)/2 - 1\}$, since we now have three classes of aircraft.

- Number of S_{qi} and W_{qi} variables: The number of these variables remains at 183 and 180 respectively.

From above we see that for T=60,the number of variables is 1890 x 3 + 183 + 180 = 6033.

2. Number of Constraints:

The number of type (i) constraints is now 3 x T since we now have 3 aircraft classes. The number of type (ii) and type (iii) constraints remains unchanged. Thus the total number of equations is: 60 x 3 + 3 + 183 = 366. The matrix size for the three aircraft classes case is then 366 x 6033.

The constraint matrix size for both, one class and three classes of aircraft models, yields problems that can be solved on a personal computer using available linear programming software packages. Since the number of constraints is large, a modeling language is needed to input the model. In Chapter 5, we give details on the software/hardware combination used to solve problem instances for Boston's Logan Airport.

Worth mentioning, introducing the additional modeling constraints of Section 3.3 results in problems that still can be solved on a personal computer. These formulations have a significantly lower number of ground-hold variables, and at most 180 additional constraints, if we introduce constraints for aircraft air delay limits for each capacity case-period.

CHAPTER 4

4. DYNAMIC STOCHASTIC PROGRAMMING SOLUTION TO THE GHPP

In Chapter 2, we discussed the need of simplifying the model for the GHPP as application of the dynamic programing algorithm developed in Section 2.2 was limited to solution of small problems. Through the simplified model presented in Chapter 3, we were able to provide static solutions to the GHPP for any airport in the US ATC network using stochastic linear programming with one stage. In this chapter, we solve the model of Chapter 3 through a stochastic linear programming with recourse formulation, achieving the goal of solving real instances of the GHPP dynamically.

In Section 4.1 we show that for airport landing capacity forecasts consisting of three scenarios, an optimal dynamic strategy assigns ground-holds at most three times during the time interval of interest (i.e., a day's operation). In Section 4.2, we discuss how the input schedule and decision variables, for the static solution presented in Chapter 3, are modified in the dynamic case. Section 4.3 develops the stochastic programming with recourse formulation for a single class of aircraft, which we extend to cover the case of three classes of aircraft in Section 4.4. Finally, in Section 4.5, we explore the size of the constraint matrix for the stochastic programming with recourse formulation, and verify that the resulting linear programs can still be solved on a personal computer.

4.1 Number of Stages in the Dynamic Solution

Figure 4.1 shows that, for a capacity forecast consisting of three scenarios, the joint PMF of airport Z landing capacities gets updated at most three times during the interval [0, L]. These three instants (denoted t_1, t_2 and t_3 in Figure 4.1) define three stages comprising the time intervals $[t_1, t_2)$, $[t_2, t_3)$, and $[t_3, L]$. We see that within each stage the conditional PMF for future airport landing capacity does not change. Therefore a dynamic solution to the GHPP assigns optimal ground-holds at the beginning of each stage.

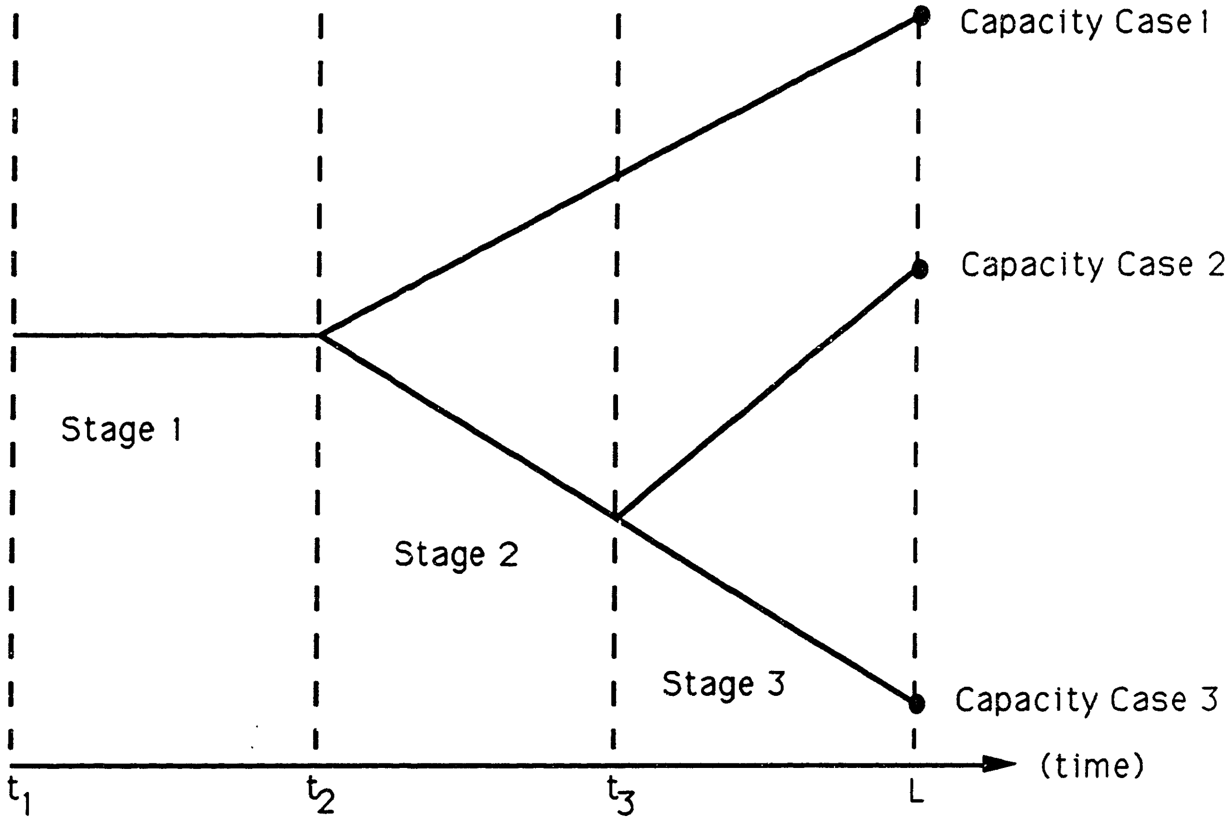


Figure 4.1

Number of Stages Defined by the PMF of Airport Capacities

We recall that in the static solution presented in Chapter 3, the time interval [0, L] comprised a single stage, resulting in a "here-and-now" solution which assigned ground-holds at t_1. In the dynamic case there are up to three stages at which we make ground hold decisions. Thus, we need to incorporate information on aircraft departure times, in addition to arrival times, as described in the next section.

4.2 Modifying the Model for the Dynamic Case

The model for the dynamic case is similar to the one presented in Section 3.1, except for the input aircraft schedule and the ground-hold decision variables (which must be modifyed in order to consider the stage during which aircraft are scheduled to depart as well as the arrival period at airport Z), assumptions for the dynamic model are exactly as described in Sections 3.1.1 through 3.1.6.

In the notation introduced below, t_s refers to the time period at which stage s starts. Notice that t_s is defined by the profile of the capacity forecast, and that in general we can have three or fewer stages[7] .

4.2.1 Input Schedule for Daily Operations

In the dynamic problem we need to consider at which stage the aircraft are scheduled to depart as well as the scheduled arrival period. Thus, we have:

[7] Throughout this chapter we assume that the capacity forecast generates three stages. The case of one stage is equivalent to the static problem discussed in Chapter 3, while the case of two stages is easily derived from the three stage solution.

(i) **For One Aircraft Class:**

N_{si}: The number of aircraft scheduled to depart during stage s and scheduled to arrive at airport Z during period i; $s = 1, 2, 3; i > t_s$.

(ii) **For Three Aircraft Classes:**

N_{ksi}: The number of class k aircraft scheduled to depart during stage s and arrive to airport Z during period i; $k, s = 1, 2, 3; i > t_s$.

Notice that for (i) we have $\sum_{s=1}^{3} \sum_{i=t_{s+1}}^{T} N_{si} = N_{TOT};$

and for (ii) $\sum_{k=1}^{3} \sum_{s=1}^{3} \sum_{i=t_{s+1}}^{T} N_{ksi} = N_{TOT}.$

We see that in order to generate the input schedule defined above; first, we generate the time periods at which each stage starts using the capacity forecast, noticing that for stage one we have $t_1 = 1$ for all capacity forecasts. Then, we use the information on individual flights (i.e., scheduled departure period, arrival period, and aircraft class) in order to assign each flight to its corresponding departing stage, arrival period and class.

4.2.2. Ground-Hold Decision Variables

From the input schedule above, we see that the ground-hold decision variables are:

(i) **For One Aircraft Class:**

X_{qsij}: The number of aircraft originally scheduled to depart during stage s and arrive at airport Z during time period i, which are rescheduled to arrive during period j, under capacity case q;
$q, s = 1, 2, 3;\ i > t_s;\ i \leq j \leq T+1$

(ii) **For Three Aircraft Classes:**

X_{qksij}: The number of aircraft of class k scheduled to depart during stage s and arrive at airport Z during time period i which are rescheduled to arrive during j, under capacity case q.
$q, k, s = 1, 2, 3; \;\; i > t_s; \;\; i \leq j \leq T+1$

We notice that for (i) we have that :

$$\sum_{j=i}^{T+1} X_{qsij} = N_{si},$$

where N_{si} is the input schedule defined in Section 4.2.1, and for (ii):

$$\sum_{j=i}^{T+1} X_{qksij} = N_{ksi},$$

where N_{ski} is the input schedule defined in Section 4.2.1.

Worth mentioning, the set of variables defining the queueing process at airport Z is identical to the one in the static model since our assumption regarding the cost of air delays requires that we only consider the number of aircraft unable to land at the end of any period.

4.3 Stochastic Programming With Recourse Model

In this section we formulate the dynamic GHPP as a stochastic linear programming problem with three stages, in which ground-holds are assigned at the beginning of each stage. We follow the framework presented in Section 3.2, where we developed the stochastic programming with one stage formulation. Initially, we limit our attention to a single class of aircraft.

4.3.1 Formulating the Deterministic Problem

This is the formulation that would result if we knew with certainty that capacity case q, with capacities $K_{q1}, \ldots, K_{qT+1}$; would occur. We see that the following integer programming formulation defines the optimization problem of minimizing total delay costs:

$$\text{Minimize:} \quad \sum_{s=1}^{3} \sum_{i=t_{s+1}}^{T} \sum_{j=i+1}^{T+1} C_g(j-i)\, X_{qsij} + c_a \sum_{i=1}^{T} W_{qi} \tag{4.1}$$

subject to:

(i) All aircraft scheduled to depart during stage s and arrive during period i must be rescheduled to arrive at airport Z during periods i, . . ., T+1:

$$\sum_{j=i}^{T+1} X_{qsij} = N_{si}\,; \quad s = 1, 2, 3; \; i = t_{s+1}, \ldots, T$$

(ii) The flow balance at arrivals airport Z at the end of each period yields:

$$W_{qi} \geq \sum_{\left(\substack{s \\ \text{such that} \\ t_s < i}\right)} \sum_{j=t_{s+1}}^{i} X_{qsji} + W_{qi-1} - K_{qi}; \; i = 1, \ldots T+1$$

(with $W_{q0} = W_{qT+1} = 0$)

(iii) X_{qsij}, $W_i \geq 0$ and integer.

The formulation above is similar to the one presented in Section 3.2.1, and can be transformed to a minimum cost flow problem in an uncapacitated network. Thus, we can relax the integrality constraint. The resulting problem has the same objective function and the following set of constraints.

(i) $$\sum_{j=i}^{T+1} X_{qsij} = N_{si}; \quad s = 1, 2, 3; \quad i = t_{s+1}, \ldots, T$$

(ii) $$\sum_{i=1}^{T+1} S_{qi} = \sum_{i=1}^{T} K_{qi};$$

(iii) $$W_{qi} - W_{qi-1} - \sum_{\left(\substack{s \text{ such that} \\ t_s < i}\right)} \sum_{j=t_{s+1}}^{i} X_{qsji} - S_{qi} = -K_{qi}; \quad i = 1, \ldots, T+1$$

(with $W_{q0} = W_{qT+1} = 0$)

(iv) $X_{qsij}, W_{qi}, S_{qi} \geq 0$

Before we proceed further we make an important observation. The models developed throughout this chapter assume that ground-holds assigned at each stage are final. Therefore, at stage s+1 we do not revise ground-holds for aircraft scheduled to depart during stage s and still waiting to depart at the beginning of stage s+1 due to previous ground-holding. This assumption has only a small impact on the quality of the solution because the number of stages is small compared to the total number of periods. Therefore, the number of aircraft delayed in previous stages and still waiting to depart at a future stage is small compared to the total number of flights. In practice, we are likely to limit the magnitude of ground-holds, further reducing the "overlap" of aircraft between stages. Also, this assumption is in line with ATC practice of trying to assign firm ground-holds whenever possible in order to facilitate schedule adjustment planning by the airlines.

4.3.2 The Distribution Problem

As mentioned in Chapter 3, the distribution problem assumes that we find out the capacities for all the periods before we make the ground-hold decisions. Under this assumption, the solution to the GHPP is a complete policy consisting of optimal ground-holds for each capacity case. We notice that the distribution problem is static in nature. Still, we refer to stages in the formulation below since the distribution formulation is an intermediate step in developing the recourse formulation.

$$\text{Minimize:} \quad \sum_{q=1}^{3} p_q \left\{ \sum_{s=1}^{3} \sum_{i=t_{s+1}}^{T} \sum_{j=i+1}^{T+1} C_g(j-i)\, X_{qsij} + c_a \sum_{i=1}^{T} W_{qi} \right\} \qquad (4.2)$$

subject to:

For each q = 1, 2, 3:

(i) $$\sum_{j=i}^{T+1} X_{qsij} = N_{si}; \quad s = 1, 2, 3; \quad i = t_{s+1}, \ldots, T$$

(ii) $$\sum_{i=1}^{T+1} S_{qi} = \sum_{i=1}^{T} K_{qi};$$

(iii) $$W_{qi} - W_{qi-1} - \sum_{\left(\substack{s \text{ such that} \\ t_s < i}\right)} \sum_{j=t_{s+1}}^{i} X_{qsji} - S_{qi} = -K_{qi}; \quad i = 1, \ldots, T+1;$$

(with $W_{qo} = W_{qT+1} = 0$)

(iv) $X_{qsij}, W_{qi}, S_{qi} \geq 0.$

Notice that the solution to the distribution problem is equivalent to solving three separate minimum cost flow problems, one for each q, since the constraint matrix consists of three separate network components. Unfortunately, when we assign ground-holds at the beginning of stages 1, 2, and 3 we have limited information on future airport capacity (i.e., we only have access to the conditional PMF of future airport landing capacities); thus, we need to modify the distribution problem formulation in order to solve the multistage problem.

4.3.3 Formulating the Three Stage Problem

The dynamic solution to the GHPP, for the capacity forecasts under consideration, reduces to solving a stochastic linear programming problem with three stages. Ground-holds are assigned at the beginning of stages 1, 2 and 3. Referring to Figure 4.1, we see that in stage 1 we define a single set of ground holds thus we need to introduce the following set of constraints to the distribution problem formulation:

(v) $X_{11ij} = X_{21ij} = X_{31ij}; \qquad i = 2, \ldots, T; \quad i \le j \le T+1$

At stage 2, we assign ground-holds conditioned on being on the upper branch or lower branch of the tree in Figure 4.1. If we are on the upper branch we are in capacity case 1 and assign ground-holds X_{12ij}. If we are in the lower branch we assign a single set of ground-holds corresponding to capacity cases 2 and 3. Thus we need to introduce constraints:

(vi) $X_{22ij} = X_{32ij}; \quad i = t_2+1, \ldots, T; \quad i \le j \le T+1$

Finally, at stage 3 all uncertainty has been resolved. We know exactly which capacity case will define future airport capacity. Thus, no further constraints on ground-holds are needed as ground-holds are defined for each of the three capacity cases.

We see that after adding constraints (v) and (vi) to the distribution problem, the constraint matrix becomes block angular as shown in Figure 4.2. This structure is similar to that of the single stage formulation of Chapter 3, making the problem suitable for decomposition techniques.

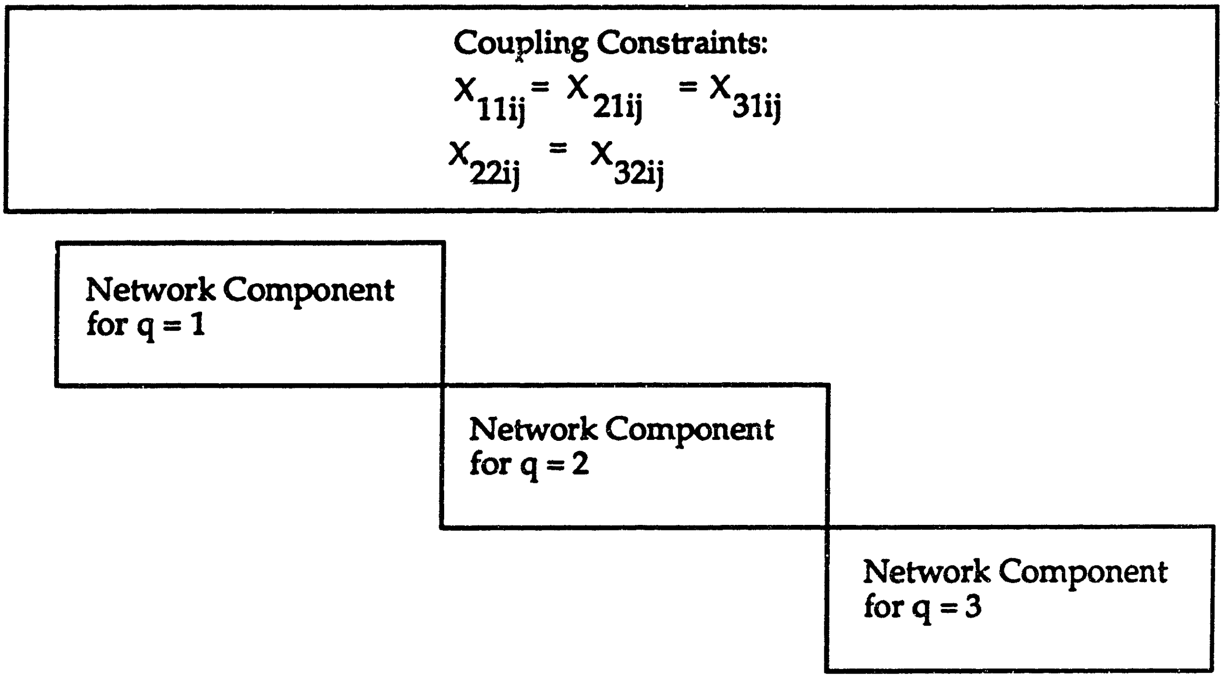

Coupling Constraints:
$X_{11ij} = X_{21ij} = X_{31ij}$
$X_{22ij} = X_{32ij}$
Network Component for q = 1
Network Component for q = 2
Network Component for q = 3

Figure 4.2

Constraint Matrix Structure for the Multistage Problem

The coupling constraints introduced in the multistage formulation break the network structure of the problem. However, as was the case with the single stage formulation, optimal solutions generated for the practical problems in Chapter 5 are integer. After substituting constraints (v) and (vi) into the distribution model: (keeping variables with the lower stage index), we get:

$$\text{Minimize: } \sum_{i=1}^{T} \sum_{j=i+1}^{T+1} C_g(j-i)\, X_{11ij} + p_1 \sum_{i=t_{2+1}}^{T} \sum_{j=i+1}^{T+1} C_g(j-i)\, X_{12ij}$$

$$+ (p_2 + p_3) \sum_{i=t_{2+1}}^{T} \sum_{j=1+1}^{T+1} C_g(j-i)\, X_{22ij} + \sum_{q=1}^{3} p_q \sum_{i=t_{3+1}}^{T} \sum_{j=i+1}^{T+1} C_g(j-i)\, X_{q3ij}$$

$$+ c_a \sum_{q=1}^{3} p_q \sum_{i=1}^{T} W_{qi} \qquad (4.3)$$

subject to:

(i) $$\sum_{j=i}^{T+1} X_{11ij} = N_{1i}; \qquad i = 2, \ldots, T$$

(ii) $$\sum_{j=i}^{T+1} X_{q2ij} = N_{2i}; \qquad i = t_2+1, \ldots, T;\ q = 1, 2$$

(iii) $$\sum_{j=i}^{T+1} X_{q3ij} = N_{3i}; \qquad i = t_3+1, \ldots, T;\ q = 1, 2, 3$$

(iv) $$\sum_{j=1}^{T+1} S_{qi} = \sum_{j=1}^{T} K_{qi}; \qquad q = 1, 2, 3.$$

(v) $$W_{1i} - W_{1i-1} - \sum_{\left(\substack{s\ \text{s.t.} \\ t_{s+1} < i}\right)} \sum_{j=t_{s+1}}^{i} X_{1sji} - S_{1i} = -K_{1i};$$

(vi) $$W_{2i} - W_{2i-1} - \sum_{j=2}^{i} X_{11ji} - \sum_{j=t_{2+1}}^{i} X_{22ji} - \sum_{j=t_{3+1}}^{i} X_{23ji} - S_{2i} = -K_{2i};$$

(vii) $$W_{3i} - W_{3i-1} - \sum_{j=2}^{i} X_{11ji} - \sum_{j=t_{2+1}}^{i} X_{22ji} - \sum_{j=t_{3+1}}^{i} X_{33ji} - S_{3i} = -K_{3i};$$

With equations (v), (vi) and (vii) for $i = 1, \ldots, T+1$; with $W_{q0} = W_{qT+1} = 0$)

(viii) $X_{qsij}, W_{qi}, S_{qi} \geq 0$ and integer.

We see that the formulation above does not have a network structure as variables X_{1sij} appear with a -1 coefficient in three different constraints and variables X_{2sij} appear with a -1 coefficient in two different constraints. In addition to the constraints in the model above, we could introduce constraints that limit congestion at Z by specifying the maximum number of aircraft allowed to queue at the end of any period under any capacity case. Also, we can limit the number of ground-hold periods by limiting the summation on the j index for constraints (i) - (iv).

4.4 Multistage Formulation Covering Three Aircraft Classes

Once we have derived the single aircraft class formulation, we can immediately extend the model to several aircraft classes by using the notation developed for multiple aircraft classes.

In the objective function we minimize expected delay costs across aircraft classes:

Minimize

$$\sum_{k=1}^{3}\Big\{\sum_{i=1}^{T}\sum_{j=i+1}^{T+1} C_g(k,j-i)\,X_{1k1ij}+ p_1\sum_{i=t_2+1}^{T}\sum_{j=i+1}^{T+1} C_g(k,j-i)\,X_{1k2ij}$$

$$+(p_2+p_3)\sum_{i=t_2+1}^{T}\sum_{j=i+1}^{T+1} C_g(k,j-i)\,X_{2k2ij}+\sum_{q=1}^{3}p_q\sum_{i=t_3+1}^{T}\sum_{j=i+1}^{T+1}C_g(k,j-i)\,X_{qk3ij}\Big\}$$

$$+c_a\sum_{q=1}^{3}p_q\sum_{i=1}^{T}W_{qi} \qquad (4.4)$$

subject to:

$$\text{(i)} \qquad \sum_{j=i}^{T+1} X_{1k1ij} = N_{k1i}; \qquad k = 1, 2, 3; \; i = 2, \ldots, T$$

$$\text{(ii)} \qquad \sum_{j=i}^{T+1} X_{qk2ij} = N_{k2i}; \qquad k = 1, 2, 3; \;\; i = t_{2+1}, \ldots, T \;\; q = 1, 2$$

$$\text{(iii)} \qquad \sum_{j=i}^{T+1} X_{qk3ij} = N_{k3i}; \qquad k = 1, 2, 3; \;\; i = t_{3+1}, \ldots, T \;\; q = 1, 2, 3$$

$$\text{(iv)} \qquad \sum_{j=1}^{T+1} S_{qi} = \sum_{j=1}^{T} K_{qi}; \qquad q = 1, 2, 3.$$

$$\text{(v)} \quad W_{1i} - W_{1i-1} - \sum_{k=1}^{3} \sum_{\begin{pmatrix} s \text{ s.t.} \\ t_s < i \end{pmatrix}} \sum_{j=t_{s+1}}^{i} X_{1ksji} - S_{1i} = K_{1i};$$

$$\text{(vi)} \quad W_{2i} - W_{2i-1} - \sum_{k=1}^{3} \left(\sum_{j=2}^{i} X_{1k1ji} + \sum_{j=t_{2+1}}^{i} X_{2k2ji} + \sum_{j=t_{3+1}}^{i} X_{2k3ji} \right) - S_{2i} = -K_{2i}$$

$$\text{(vii)} \quad W_{3i} - W_{3i-1} - \sum_{k=1}^{3} \left(\sum_{j=2}^{i} X_{1k1ji} + \sum_{j=t_{2+1}}^{i} X_{2k2ji} + \sum_{j=t_{3+1}}^{i} X_{3k3ji} \right) - S_{3i} = -K_{3i}$$

Equations (v), (vi), (vii) are for $i = 1, \ldots, T + 1$ with $W_{q0} = W_{qT+1} = 0$).

(viii) $X_{qksij}, W_{qi}, S_{qi} \geq 0$ and integer.

We see that the size of the problem to be solved increases with the number of stages as well as with the number of aircraft classes. In the next section, we explore constraint matrix size for practical problems.

4.5 Constraint Matrix Size

We determine the approximate size of the constraint matrix for the single aircraft class and three aircraft classes models developed in Section 4.3 and 4.4 respectively. In our calculation, we assume forecasts consisting of three capacity scenarios, three aircraft classes, three stages, and sixty time periods (i.e., T = 60).

4.5.1 Constraint Matrix Size for One Aircraft Class

First we determine the total number of constraints in the optimization model (4.3):

Constraint	Number of equations
(i)	60
(ii)	60 x 2 = 120
(iii)	60 x 3 = 180
(iv)	3
(v) - (vi)	(61 x 3) = 183
Total	586

If we introduce limits in the maximum airborne delay (i.e., queue length at the end of every period) for every capacity case as described in Chapter 3, we would increase the total number of equations by 3 x 60 = 180 for a total of 766 equations.

In order to calculate the total number of variables we assume that ground-holds are limited to 20 time periods (i.e., 5 hours). This results in the following approximate number of variables.)

To calculate the number of X_{qsij} variables we assume that stages 2 and 3 start at $t_2 = 20$, and $t_3 = 40$. The approximate number of X_{qsij} variables is then:

$$2 \times \{20 \times 20 + 2 \times 20 \times 20 + 3 \times 20 \times 20\} = 6 \times 800 = 4800.$$

The number of W_{qi} variables and S_{qi} variables is as in the static case: 180 + 183 = 363.

From the analysis above, we see that the constraint matrix size for the three stage problem with a single aircraft class is approximately 766 x 5163.

4.5.2 Constraint Matrix Size for Three Aircraft Classes

The number of equations as well as the number of variables increases significantly vis a vis the single class problem.

The total number of constraints in the optimization model (4.4):

Expression	Number of equations
(i)	3 x 60 = 180
(ii)	3 x 120 = 360
(iii)	3 x 180 = 540
(iv)	3
(v) - (vi)	183
	Total: 1266

Again, introducing limits in the maximum airborne queueing delay for each capacity period we obtain a total number of equations of 1266 + 180 = 1446.

The total number of variables assuming a maximum of 20 periods of ground-hold is:

– number of X_{qksij} variables (3 x 4800) :	14,400
– number of S_{qi} and W_{qi} variables:	360
Total:	14,760

The constraint matrix size for the three stage problem with three aircraft classes is 1446 x 14760.

We see that even for the case of three aircraft classes, the resulting linear program can be solved on a personal computer as will be confirmed by the experimental results of Chapter 5.

The dynamic models developed in this chapter yield a significant improvement in the solution vis. a vis. the static model and current ATC practices. Chapter 5 presents a detailed analysis of experimental results for Boston's Logan Airport.

CHAPTER 5

5. EXPERIMENTAL RESULTS

In this chapter we assess the performance of the static and dynamic stochastic programming models developed in Chapter 3 and 4. The algorithms tested include a very fast heuristic that showed performance comparable to that of the dynamic stochastic programing models. We provide a detailed analysis on how the airport capacity forecasts and air costs affect the relative performance of the algorithms. The chapter is organized as follows:

In Section 5.1 we present the data that define the instances of the GHPP for Logan airport to be solved (i.e., the aircraft schedule data, the airport capacity forecasts, and the ground/air cost functions). In section 5.2 we describe the different algorithms that will be evaluated, and in 5.3 we give an example of a "complete experiment", consisting of an instance of the GHPP for Logan to be solved using the algorithms described in 5.2. In Section 5.4, we discuss performance of the static and dynamic stochastic programming algorithms and of a fast heuristic as compared to a deterministic solution. As a final exercise, in Section 5.5, we compare the performance of the algorithms tested to the "passive"strategy of no ground-holding which minimizes total expected delay.

The choice of algorithms for testing was based on results from preliminary runs for static and dynamic stochastic programming models for a single class of aircraft.

Both models showed an improvement vs. the deterministic solution. However, the dynamic algorithm performed significantly better than the static, suggesting a greater potential for dynamic approaches to solving the GHPP. Thus, in addition to the static and dynamic stochastic programming models for a single class of aircraft we explored two dynamic algorithms: a heuristic based on dynamic use of the deterministic algorithm for a single class of aircraft and the dynamic stochastic programming model for three classes of aircraft . The deterministic dynamic heuristic performed remarkably close to the optimal stochastic dynamic solution in the majority of cases. Discriminating among aircraft classes improved the solution versus the single aircraft class model as ground-holds are assigned to the least expensive aircraft class eligible for ground-holding.

Regarding comparison to the "passive" strategy of no ground-holding, the dynamic algorithms, including the heuristic, performed remarkably better; showing total expected delays within 10% of the passive strategy, with the advantage that over 95% of the delays are on the ground. As well, the overall cost performance of all the algorithms tested, including the deterministic solution, was better than for the "passive" strategy except for cases with low air delay cost premiums versus ground costs.

5.1 Defining the GHPP for Logan Airport

In order to solve realistic problems using the algorithms developed in this thesis, we must obtain the input data and put it into the format described in Sections 3.1 and 4.2 for the static and dynamic models respectively. Next, we provide details on how the instances of the GHPP for Logan airport were generated.

5.1.1 Aircraft Schedule

The aircraft schedule data represent a typical weekday of operations at Boston's Logan Airport during the Fall of 1988 based on information taken from the November 1988 issue of the *Official Airline Guide,* including scheduled direct international flights (a total of 5 flights). Worth mentioning, there are approximately 50 unscheduled daily flights into Logan which are not subject to CFCF ground-holding. This does not affect the quality of our solutions significantly as these unscheduled flights represent less than 10 percent of the total number flights.

Appendix 2a contains information on the airport and time of departure, arrival time and aircraft type for each direct (i.e, last leg) scheduled flight into Logan. There is a total of 551 scheduled flights. The earliest scheduled departure is at 5:45 AM and the latest scheduled arrival at 11:58 PM, yielding 73 fifteen-minute time intervals. Appendix 2b shows the schedule with flights classified according to the three classes described in Section 3.1.3, the departure and the arrival period. Appendix 2c shows the static input schedule for one and three aircraft classes as defined in Section 3.1.4.

The dynamic schedules are determined from the flight schedule in Appendix 2b and the capacity forecasts which define the starting times for the stages as shown in Figure 4.1. Table 5.1 shows flight duration statistics by aircraft type for the Logan schedule data. We see that small and medium aircraft, each with approximately 45% of the scheduled flights, account for about 90% of Logan traffic. Also, we see that over 80% of the flights have duration of two or less hours. This reflects the importance of short range traffic, such as flights from New York and Washington; and Logan's role as "The New England Hub," collecting short range commuter flights and funnelling the passengers from these flights into the main air traffic network.

The high proportion of short range flights highlights the importance of dynamic approaches to solving the GHPP. We also see that if the system was able to differentiate users into three cost classes (i.e., according to aircraft type), substantial savings would be obtained by assigning ground-holds to the lowest cost aircraft eligible for delay.

Flight	Number of Aircraft and Percent (%)			
duration (hrs.)	TYPE 1	TYPE 2	TYPE 3	TOTAL
.0 - 0.5	54 (9.8)	2 (0.4)	——	56 (10.2)
0.5 - 1.0	179 (32.5)	97 (17.6)	6 (1.1)	282 (51.2)
1.0 - 2.0	18 (3.3)	91 (16.5)	6 (1.1)	115 (20.9)
2.0 - 3.0	——	36 (6.5)	8 (1.5)	44 (8.0)
3.0 - 4.0	——	21 (3.7)	19 (3.4)	40 (7.1)
4.0 -	——	2 (0.4)	12 (2.2)	14 (2.6)
	251 (45.6)	249 (45.1)	51 (9.3)	551 (100)

TABLE 5.1

Flight Duration Statistics

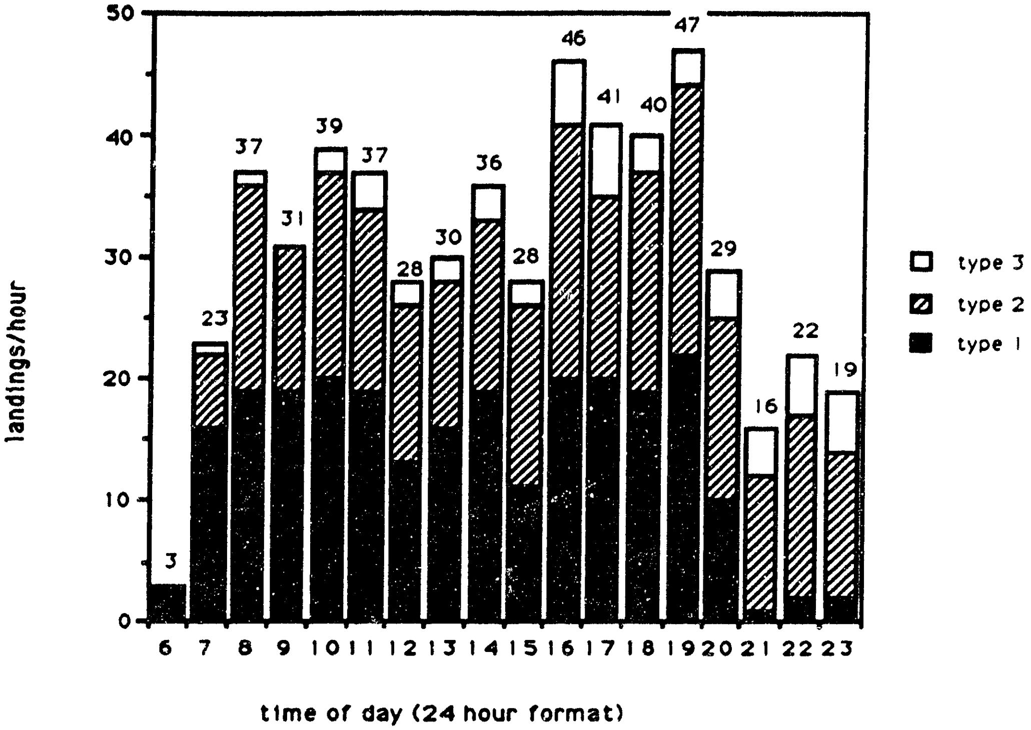


Figure 5.1

Hourly Landings By Aircraft Type for Scheduled Flights

Figure 5.1 shows hourly landings by aircraft type for scheduled flights. We see that during the busiest periods (8 to 11 and 16 to 19 hours) landing demand for scheduled flights averages 36 and 43.5 landings per hour respectively, representing 60% and 74% of the "good weather" maximum landing capacity of 60 aircraft per hour. Thus, in a good weather day this schedule yields little congestion. However, as we will discuss shortly, bad weather can significantly reduce landing capacity, and it is precisely during bad weather days that landing capacity is more uncertain. This is why probabilistic approaches to solving the GHPP yield a significant improvement in the efficiency of operations during periods of restricted airport capacity.

Regarding the landing profile by aircraft type, we see that the most significant deviation versus the average class split occurs for type 1 aircraft, which show significantly reduced activity towards the end of the day; and for type 3 aircraft which show greater activity after 15:00 hours.

5.1.2 Airport Capacity Forecast

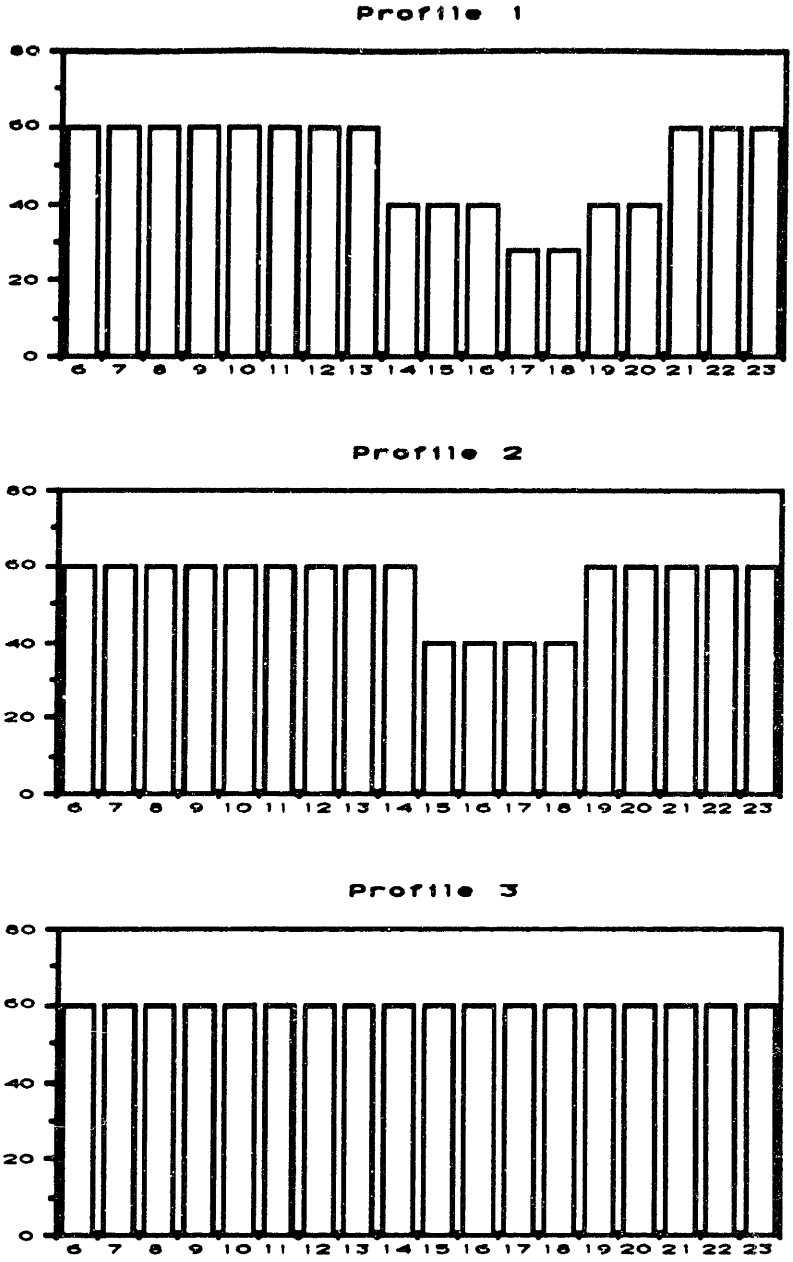
We studied a total of 10 different capacity cases consisting of three capacity profiles each. We explore four different probability scenarios for capacity cases 1-3, and a single probability scenario for capacity cases 4-10, for a total of 19 different capacity forecasts.

The capacity forecasts cover a wide variety of conditions in regard to the levels, timing and duration of restricted capacity periods, and reflect operating conditions prevailing at Logan during bad weather days. Figure 5.2 shows the capacity profiles for capacity case 1 (with stages 1, 2, and 3 starting at 6:00, 14:00, and 15:00 hours; corresponding to times t_1, t_2, and t_3 in Figure 4.1 respectively), and Table 5.2 the corresponding 4 probability scenarios. For example, for capacity case one, under probability scenario 1, profiles 1, 2, and 3 have probabilities of .5, .3, and .2 respectively. Capacity cases 2-10 with the corresponding probability scenarios are shown in Appendix 3.

	Probability Scenario			
Profile #	**1**	**2**	**3**	**4**
1	.5	.3	.3	.34
2	.3	.5	.2	.33
3	.2	.2	.5	.33

Table 5.2

Probability Scenarios for Capacity Case 1



(Vertical axis: landings/hour. Horizontal axis: time of day - 24 hour format)

Figure 5.2

Capacity Profiles for Landing Capacity Case 1

The capacity levels used in preparing the forecasts were 60, 40, and 28 landings per hour, corresponding to VFR1, VFR2/IFR1, and IFR2/IFR3 conditions respectively. VFR stands for visual flying rules and IFR for instrument flying rules. Airport landing capacity under IFR conditions decreases versus VFR conditions as aircraft minimum separation rules are enforced increasing the time between landings. As well, some landing runways available for VFR operations may not be equipped with instrument landing equipment, further reducing capacity during IFR conditions. Worth mentioning, Logan historical data indicate that VFR1 weather conditions prevail about 80% of the time, VFR2/IFR1, 12% of the time, and IFR3 and higher during the remaining 8%. Thus, we have not included extreme congestion cases such as shut down of operations, that are likely to require flight cancelations, due to unacceptable delay levels.

5.1.3 Ground and Air Delay Costs

- *Ground Delay Costs*

In order to assure FCFS within an aircraft class, the ground delay cost functions used were slightly increasing. For the case of a single aircraft class (i.e., the "average" case) ground-hold costs were scaled to $1,000/period for the first period of ground-holding and then increased by $10/period. Since we can multiply the objective function of a linear program by a constant without affecting the solution cost figures can be scaled so that they reflect actual operating costs for aircraft.

In the case of three aircraft classes, two different ground-hold cost functions were used, both yielding the average ground-hold cost, of $1,000 per period, for a single class of aircraft based on a 45%-45%-10% aircraft class split. The cost for the first period of ground delay by aircraft class for each cost function is shown in Table 5.3. The marginal rate of ground-hold cost increase is $10/period. The second function reflects a cost difference among aircraft classes which is closer to what we would expect in reality.

	Aircraft Type		
Cost Function	1 (45%)	2 (45%)	3 (10%)
1	$800	$1,133	$1,300
2	$430	$1,300	$2,225

Table 5.3

First Period Ground-Hold Delay Cost

- *Air Delay Costs*

As discussed in Section 3.3.1 we can adjust the bias of our models towards conservative (liberal) ground-holding strategies by increasing (decreasing) the cost of air delay, c_a. We explore marginal air delay costs of $1,200, $1,600, $2,000, and $3,000, representing cost premiums of approximately 20%, 60%, 100% and 200% vs. the average cost of ground delays. We solved the GHPP for each one of the problems defined in Table 5.4, using the algorithms described in Section 5.2.

Capacity Case	Number of Probability Scenarios	Number of Forecasts	Air Delay Costs	Number of Problems*
1-3	4	12	1200, 1600 2000, 3000	48
4-9	1	6	1600	6
10	1	1	3000	1
		19		55

* A problem is defined as a capacity forecast - air delay cost combination

Table 5.4

Problems Generated by the Different Capacity Forecast- Air Cost Combinations

5.2 The Algorithms

We evaluated the performance of 5 algorithms in the solution of the GHPP's defined above. Algorithms 1-4 are for a single class of aircraft, while 5 is for three classes of aircraft.

1. Deterministic: this algorithm provides a static deterministic solution to the GHPP by taking the most likely capacity profile in the probabilistic forecast as the deterministic capacity profile and disregarding completely all other profiles. Available capacity is then assigned on a FCFS basis with all delays assigned as ground-holds. This algorithm is labeled DETERM.

2. Static: This algorithm is based on the model developed in Chapter 3. It provides the optimal probabilistic static solution for the GHPP using stochastic linear programming with one stage. This algorithm is labeled STATIC.

3. Dynamic: This algorithm is based on the stochastic linear programming model with three stages described in Chapter 4. It provides the optimal dynamic solution to the GHPP under the assumptions described in Section 4.3.1. The algorithm is labeled DYNAMIC.

4. Deterministic Dynamic Heuristic: Based on preliminary results showing a significant advantage for the DYNAMIC vs. the STATIC solutions, we developed a heuristic that combines the speed of DETERM with the good performance of DYNAMIC. The algorithm establishes ground-holds dynamically (i.e., at the beginning of each stage defined by the probabilistic forecast) using DETERM. Worth noticing, some of the information conveyed by the probabilistic forecast is incorporated into the heuristic by utilizing the most likely capacity case at the current stage as the deterministic forecast input

for DETERM. The procedure is described below (please refer to Figure 4.1 which shows the stages defined by the probabilistic forecast) :

At the beginning of stage 1, the capacity case with the highest probability is taken as the deterministic forecast. We then solve the GHPP using DETERM and assign ground-holds for stage 1.

At the beginning of stage 2, we are at either the upper or lower branch of the probabilistic forecast shown in Figure 4.1. If we are at the upper branch, the deterministic forecast to be used as input for DETERM is capacity case 1; if we are at the lower branch, the deterministic forecast is given by the capacity case with the highest probability between capacity cases 2 and 3. Stage 2 ground-holds are assigned solving the GHPP using DETERM, with the ground-holds previously assigned for stage 1 as initial conditions.

Finally, at stage 3 uncertainty has been resolved and we use the capacity case we find ourselves at (i.e., capacity case 1,2 or 3) as the deterministic forecast. Stage 3 ground-holds are assigned using DETERM with the ground-holds determined in stages 1 and 2 above as initial conditions.

The dynamic heuristic described above is denoted DYNAMICH, and we see that its computational complexity is $O(T^2)$ since in order to reassign demand for any given period we need to scan $O(T)$ periods, and there are T periods. Worth mentioning, the modeling constraints described in Section 3.3.1 (e.g., limiting ground-hold durations) can be introduced without affecting the complexity of the heuristic.

5. **The Dynamic Algorithm for Three Aircraft Classes:** This algorithm is based on the stochastic linear programming model with three stages of Chapter 4, with three aircraft classes. This algorithm is denoted DYNAMIC3C for the first ground hold function defined in Section 5.1.3, and DYNAMIC3C2 for the second function.

5.3 Describing a "Complete Experiment"

As shown in Table 5.4, the different capacity forecast - air cost combinations define 55 GHPP instances for Logan airport. Each one of these problems was solved using the following algorithms: DETERM, STATIC, DYNAMICH, DYNAMIC, and DYNAMIC3C. Additionally, problems with air delay cost of $3,000/period were solved using DYNAMIC3C2 for a total of 288 solutions generated for the different algorithms. Next, we present a particular instance of the GHPP for Logan airport:

- The aircraft schedule presented in Section 5.1.1 is fixed for all GHPP instances.

- The capacity forecast is composed of a capacity case (e.g., capacity case 1 composed of three capacity profiles as shown in Figure 5.2) with an associated probability scenario (e.g., the first column of Table 5.2 which assigns probabilities of .5, .3, and .2 to capacity profiles 1,2, and 3 respectively).

- The ground-hold cost function for solutions with a single class of aircraft is as described in Section 5.1.3 and is fixed for all GHPP instances. For cases with three aircraft classes, one of the two cost functions in Table 5.3 (e.g., cost function 1) is specified.

- The marginal cost of air delays (e.g., $1600) is specified out of the four possible values presented in Section 5.1.3.

Once a GHPP instance has been formulated we generate solutions for each of the algorithms described in section 5.2. The algorithms were implemented as follows:

DETERM and DYNAMICH were implemented using C programming language code that utilizes the input schedule and the deterministic landing capacity forecast in order to assign available capacity on a FCFS basis, with delays assigned as ground-holds. However, In order to obtain the expected cost and expected delay statistics necessary to evaluate performance of the algorithms tested, we used static stochastic programming versions of the algorithms to solve the instances of the GHPP in this thesis.

STATIC, DYNAMIC, DYNAMIC3C, and DYNAMIC3C2 are stochastic linear programming algorithms. As discussed in Sections 3.6 and 4.5, the size of the stochastic linear programs we will consider can be solved on a personal computer. We used LINGO on a 386 machine with a 80387 co-processor and 4mb of RAM memory. Our version of LINGO is able to solve systems with a constraint matrix size of up to 5000 x 15,000, for moderately dense matrices.

5.4 Performance Evaluation

As described above, each GHPP instance - algorithm combination generates a solution. We will evaluate performance of the different algorithms based on the following statistics for each solution: total expected costs, expected cost of ground and air delays, expected ground delay and air delay measured in aircraft-periods. Appendix 4 shows these statistics for each solution, including the "passive" strategy of no ground-holding to be discussed in Section 5.5.

5.4.1 Overall Performance

Figures 5.3 and 5.4 show the average performance of the algorithms. Figure 5.3 is on a percentage basis (i.e., equal weight for each solution) while Figure 5.4 is average expected total delay cost.

In Figure 5.3, we see that DYNAMIC provides close to 30% savings in expected total cost vs. DETERM, while STATIC provides only a modest 6.6% savings. The performance of DYNAMICH is remarkably close to that of DYNAMIC showing 25% savings. We also see that DYNAMIC3C and DYNAMIC3C2 show 37% and 62% savings vs. DETERM. These savings are achieved by assigning ground-holds to the lowest cost aircraft eligible for delay.

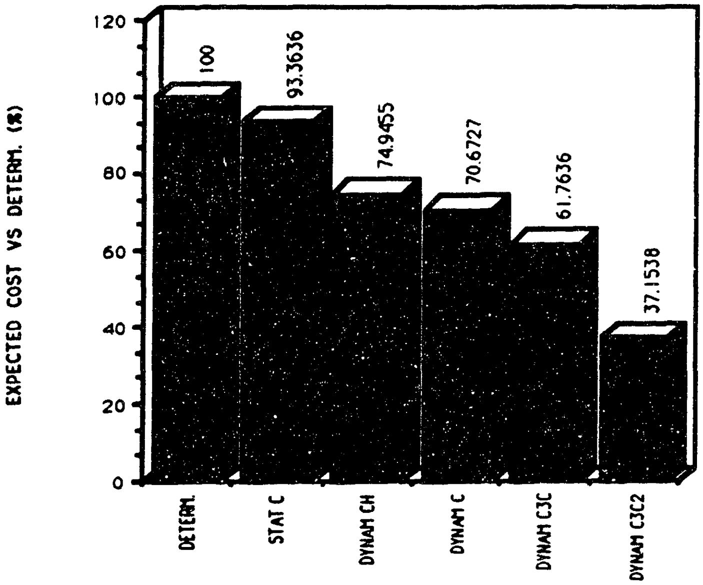


Figure 5.3
Average Cost Performance (% basis)

Figure 5.4 shows that the average cost savings provided by the dynamic algorithms (i.e., DYNAMICH, DYNAMIC, DYNAMIC3C, DYNAMIC3C2) are traceable mainly to significant reductions in the expected cost of air delays. On the other hand, STATIC shows an increase in expected air delay costs which offsets most of the savings in ground-holding cost. The reason for the significant improvement in the expected cost of air delays for the dynamic algorithms is that, by updating the capacity forecast at each stage, these algorithms generate ground-holding policies that reduce expensive air delays significantly.

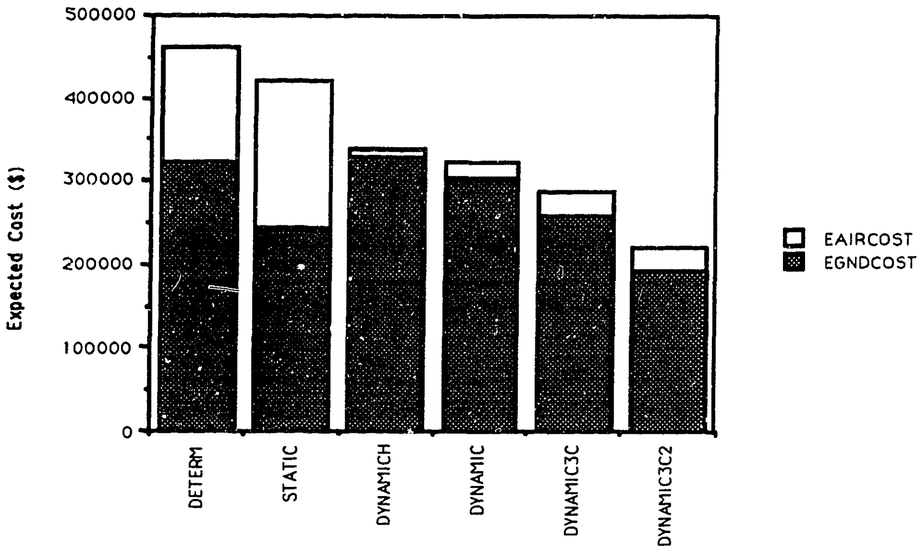
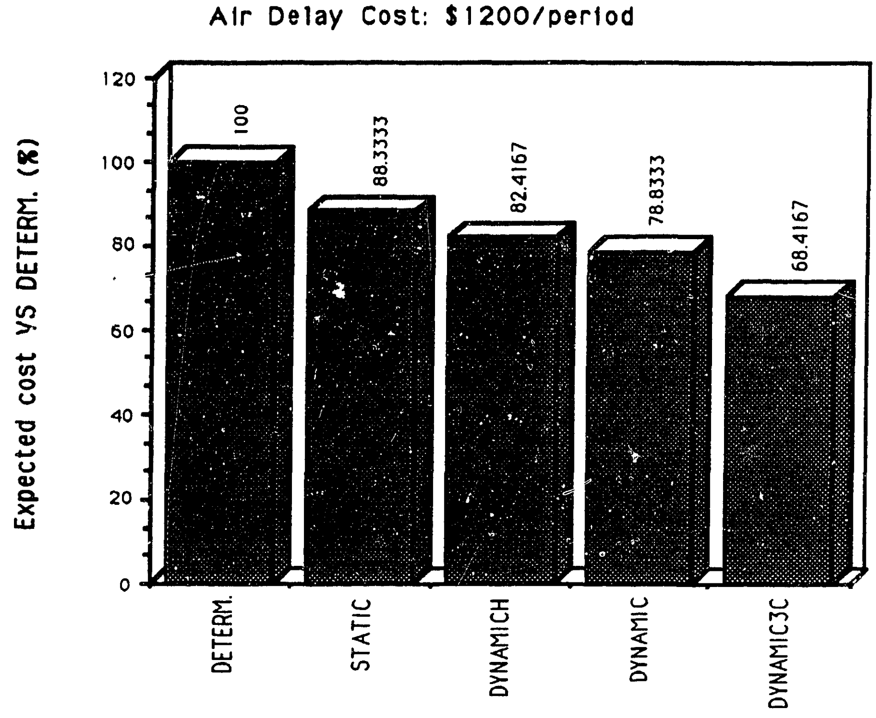


Figure 5.4
Average Cost Performance ($ basis)

5.4.2 Effect of Air Delay Costs

Figure 5.5 shows the average relative performance of the algorithms for each marginal air delay cost value tested. With no exception, the performance of the dynamic algorithms improves as the air cost increases. For example, DYNAMICH and DYNAMIC show 24% and 27% savings for marginal air delay cost of $1,600/period; while for marginal air delay cost of $3,000/period the savings are 33% and 39%.

On the other hand, the performance of STATIC seems to deteriorate as the cost of air delays increases. We see that for air delay cost of $1,200/period STATIC provides 12% savings while for $3,000/period savings are reduced to 5%. This is



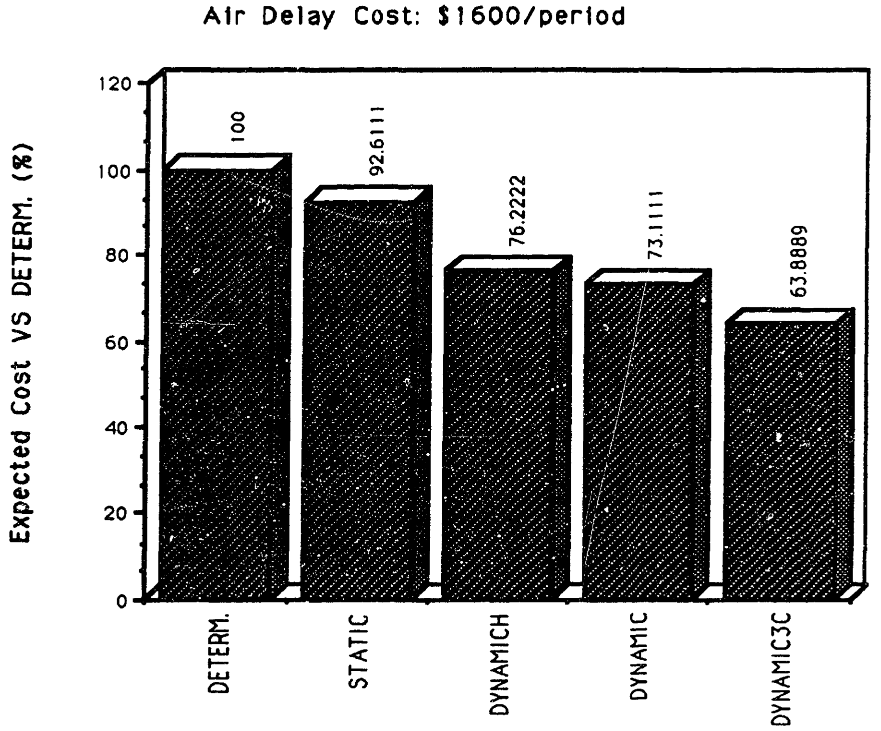
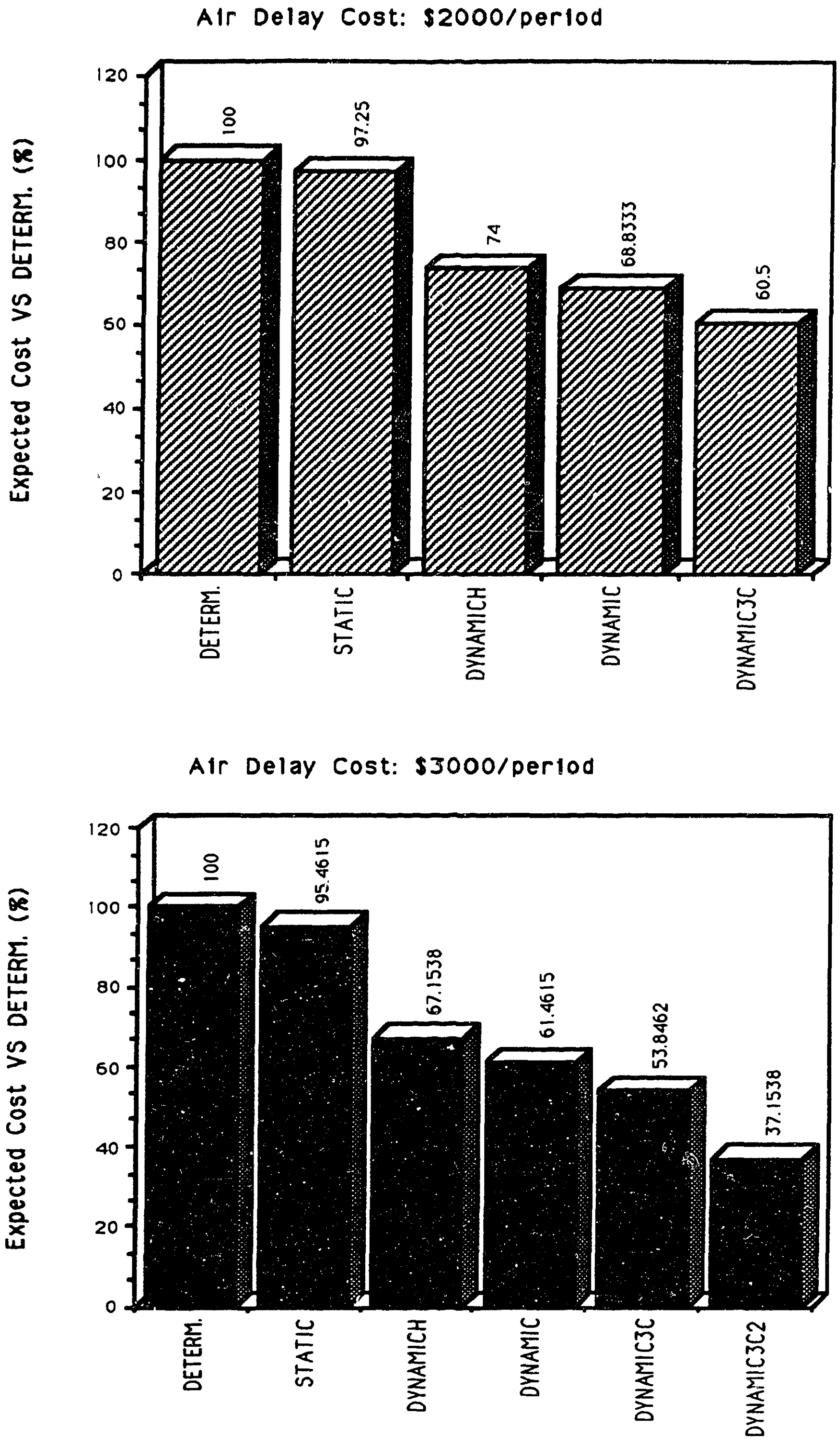


Figure 5.5a

Average Cost Performance - Air Delay Cost: $1200 & $1600



Air Delay Cost: $2000/period

Air Delay Cost: $3000/period

Figure 5.5b

Average Cost Performance -Air Delay Cost: $2000 & $3000

an important observation as the advantage of STATIC over DETERM diminishes within the range of air cost premiums we are likely to find in real problems. This is because as we increase the marginal air delay cost - particularly for cases with the most pessimistic capacity profile as the most likely profile - the solution for DETERM improves (we recall that for high enough marginal air delay cost the optimal static solution is the strategy of assigning available capacity for the most pessimistic capacity profile on a FCFS basis with all delays taken in the form of ground-holds).

We reach similar conclusions by looking at individual solutions rather than averages. Figure 5.6 shows solutions for each of the four probability scenarios in capacity case 1, for air delay cost of $1,200, $1,600, $2,000, and $3,000/period (from left to right, solutions within a given air delay cost correspond to probability scenarios 1,2,3, and 4 respectively). Again, the improvement for the dynamic solutions as the air delay cost increases is evident.

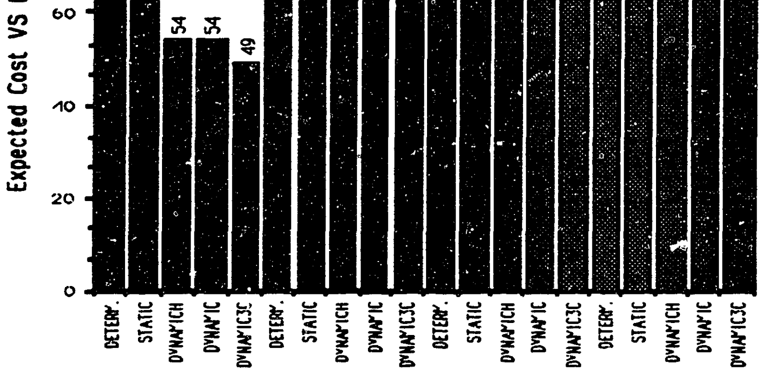


Figure 5.6a

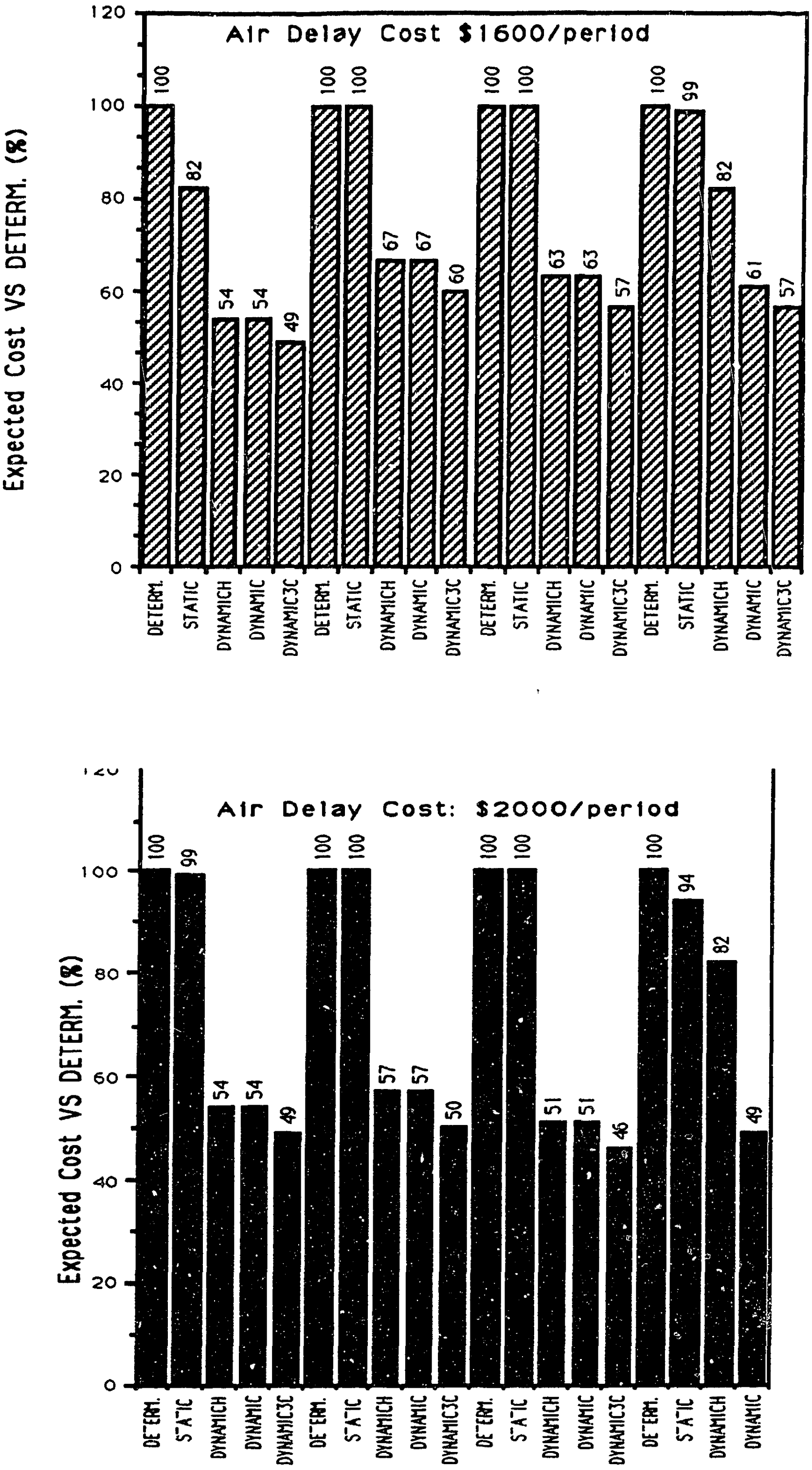


Figure 5.6b
Capacity Case 1 Cost Performance (Air Delay Cost $1600 & $2000)

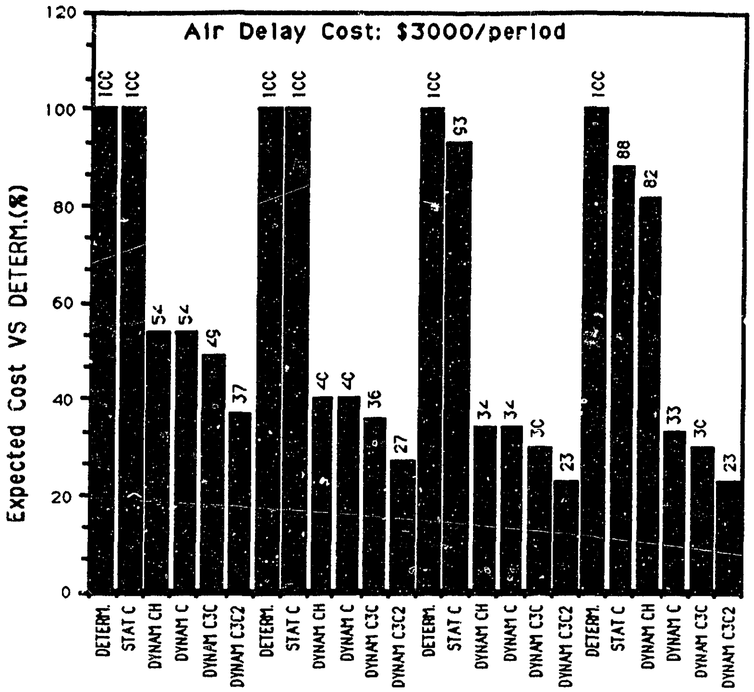


Figure 5.6c
Capacity Case 1 Cost Performance (Air Delay Cost $3000)

5.4.3 Effect of Uncertainty

Observing the capacity profiles for capacity cases 1, 2, and 3 in Appendix 3, we see that the main congestion periods occur during stages 3, 2, and 1 respectively. Figure 5.7 shows the average relative performance of the algorithms for the problems in capacity cases 1,2, and 3. We see that while the performance of STATIC is comparable for the three capacity cases, the performance of the dynamic algorithms improves as the uncertainty on congestion increases. For capacity case 1, for which congestion occurs mainly during stage 3, the stage of greatest uncertainty, we see the best performance as the dynamic algorithms assign ground-holds at the beginning of stage 3, when uncertainty in the forecast is resolved. On the other hand, capacity case 3, for which congestion occurs mainly during stage 1, the stage with the least uncertainty, we observe the worst relative performance.

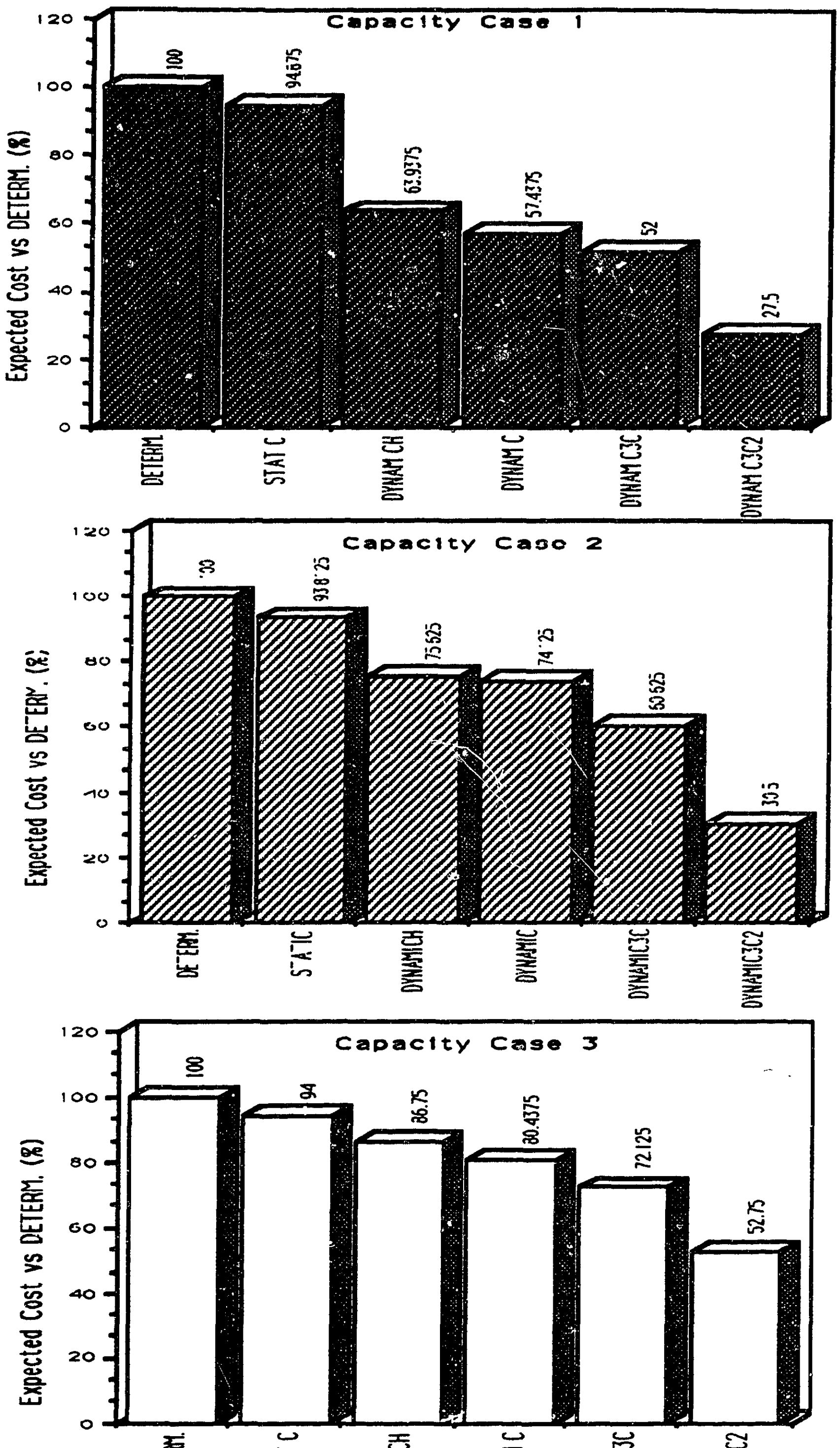


Figure 5.7

Average Cost Performance by Capacity Case (Capacity Cases 1-3)

5.4.4 Effect of Marginal Air Delay Costs on Ground and Air Delays

Figure 5.8 shows the average expected number of aircraft-periods of air and ground delay for all problems; while Figure 5.9 shows expected air and ground delay for each marginal air delay cost value tested.

From Figure 5.8 we see that the key advantage of the dynamic algorithms is a significant reduction on the magnitude of expected air delays with no increase on ground-holds compared to DETERM. Expected air delays for DYNAMIC and DYNAMIC3C represent only 16% and 23% of air delays for DETERM respectively, with the added advantage of having average ground-holds slightly below those for DETERM. Notice that DYNAMIC3C2 cannot be compared to DETERM in Figure 5.8 since DYNAMIC3C2 applies only to problems with air delay cost of $3,000/period.

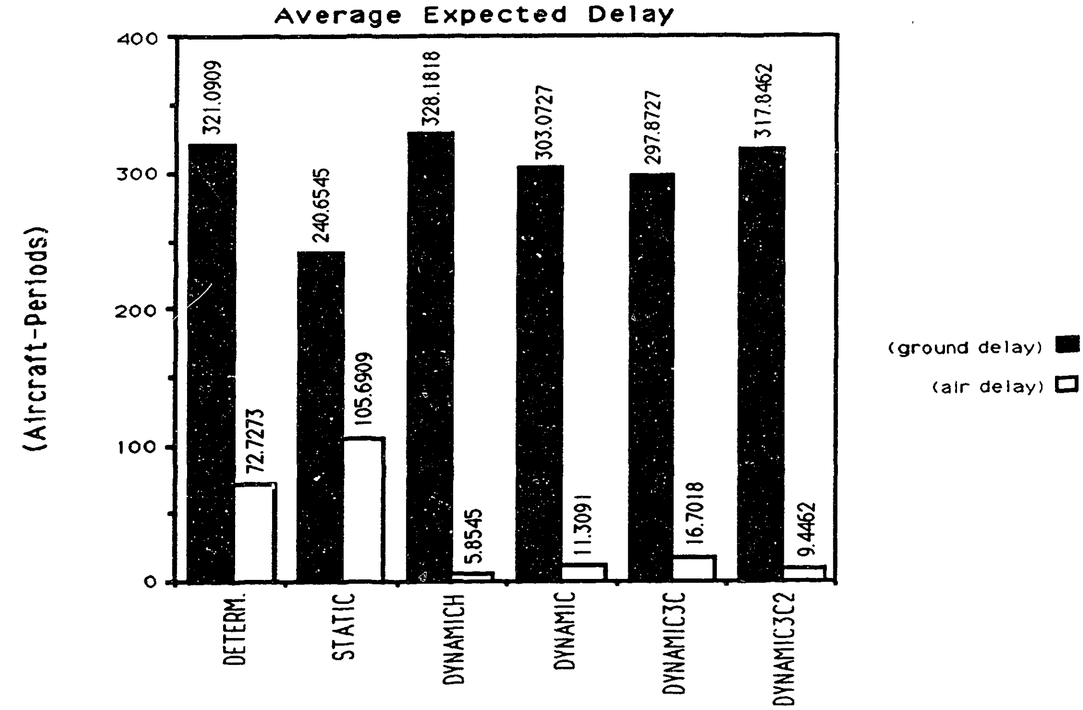


Figure 5.8

Average Expected Ground and Air Delay

In Figure 5.9 we see that, as we had anticipated, average expected ground-hold and air delays for STATIC, DYNAMIC and DYNAMIC3C show declining expected air delays as air delay costs increase; while DETERM and DYNAMICH are insensitive to the cost of air delays since these algorithms assign available landing capacity on a FCFS basis regardless of the cost of air delays.

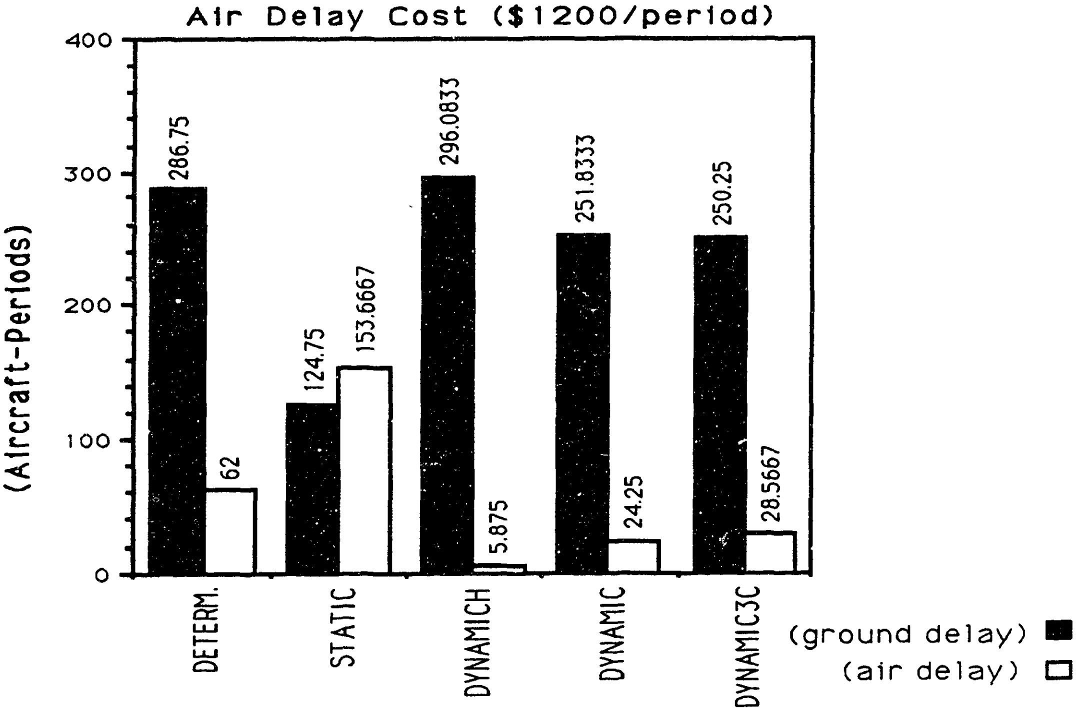


Figure 5.9a

Average Ground and Air Delays - Air Delay Cost $1200

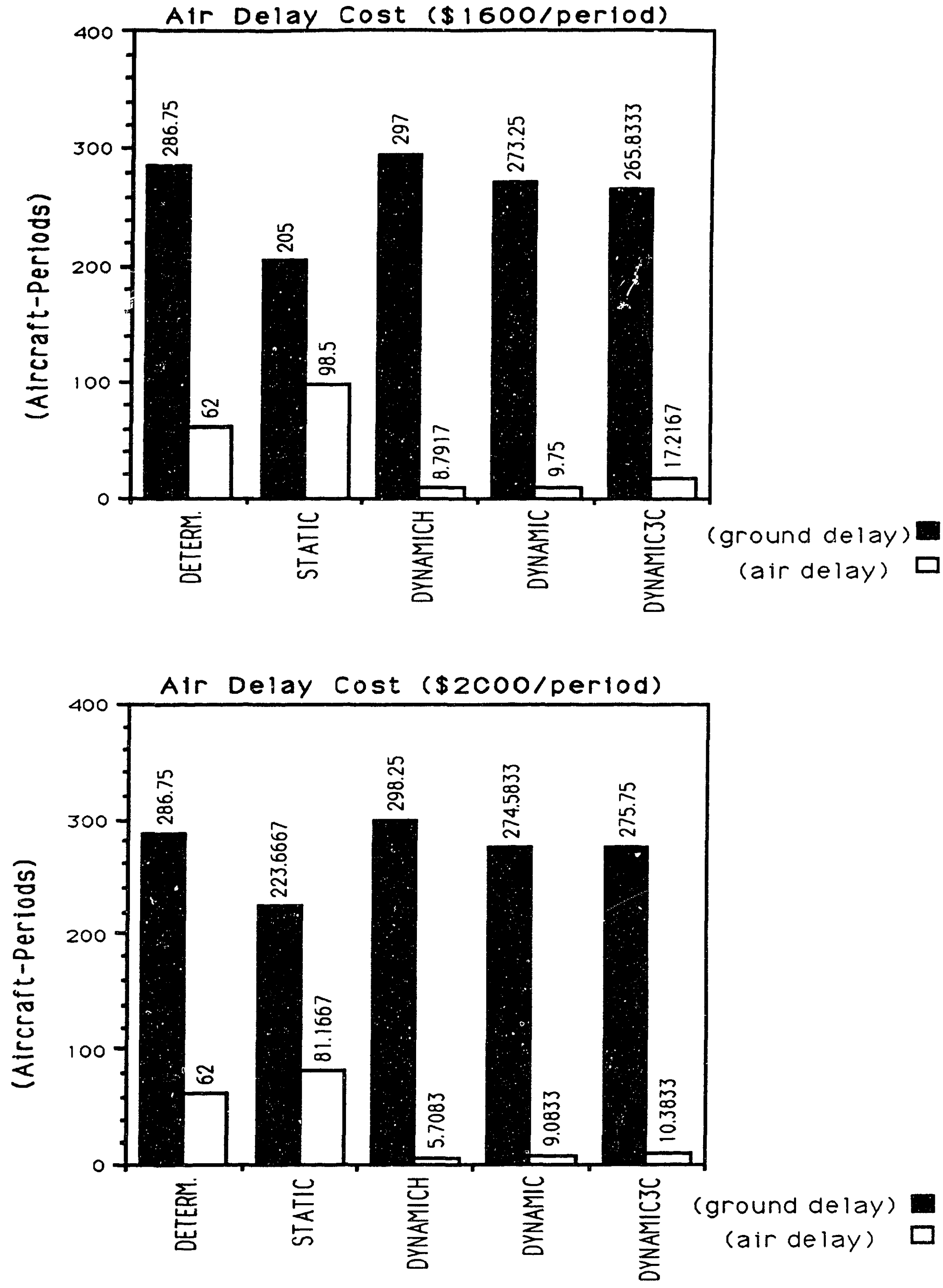


Figure 5.9b

Average Ground and Air Delays - Air Delay Cost $1600 & $2000

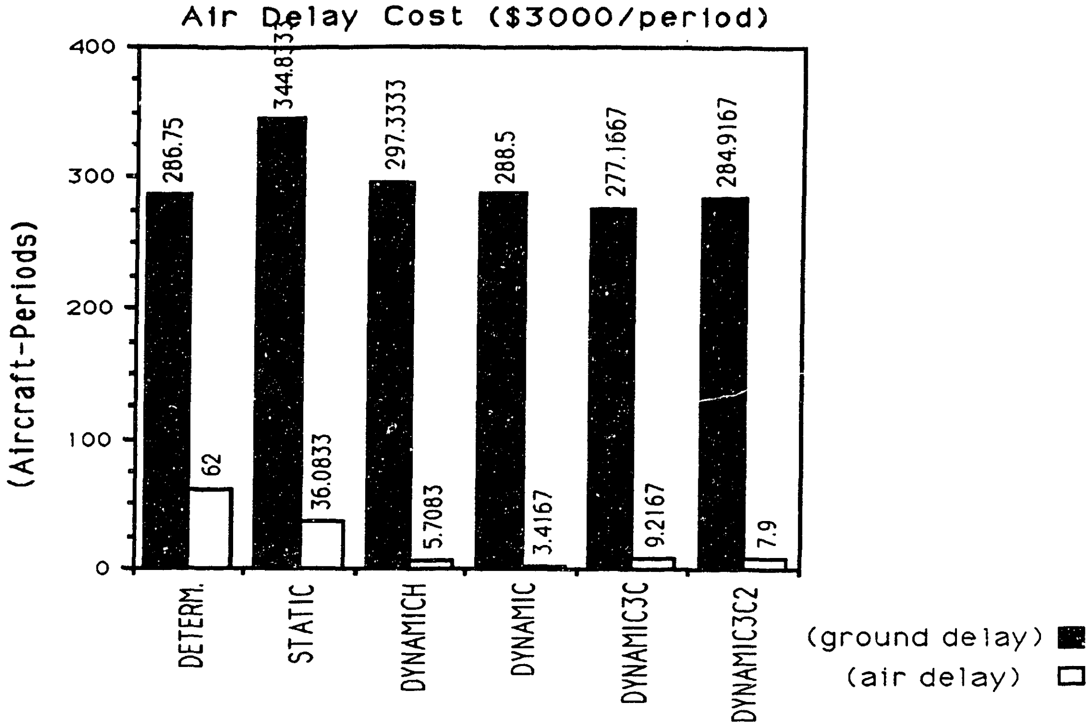


Figure 5.9c

Average Ground and Air Delays - Air Delay Cost $3000

5.4.5 Effect of Ground-Hold Cost Function For Three Aircaft Classes

Figure 5.10 shows the average expected ground delay for each aircraft class for DYNAMIC3C and DYNAMIC3C2 for problems with air cost $3,000 within capacity cases 1-3. Unlike algorithms for a single aircraft class, those that make distinctions among aircraft classes result in ground-holds that assign the lower cost classes ground-holds which are substantially higher than the 45%-45%-10% split for aircraft classes 1,2, and 3 respectively. Specifically, for DYNAMIC3C and DYNAMIC3C2 class 1 aircraft account for 80% of total ground-holds, while classes 2 and 3 - which comprise 55% of the total number of aircraft - account for only 20% of total ground-holds, well bellow their proportion in the schedule. Notice also how the lower ground-hold cost for class 1 aircraft in the second ground-hold cost function ($430 vs. $800 for the first cost function) results in higher expected average ground-holds for DYNAMIC3C2 (284.9) vs. DYNAMIC3C (277.2).

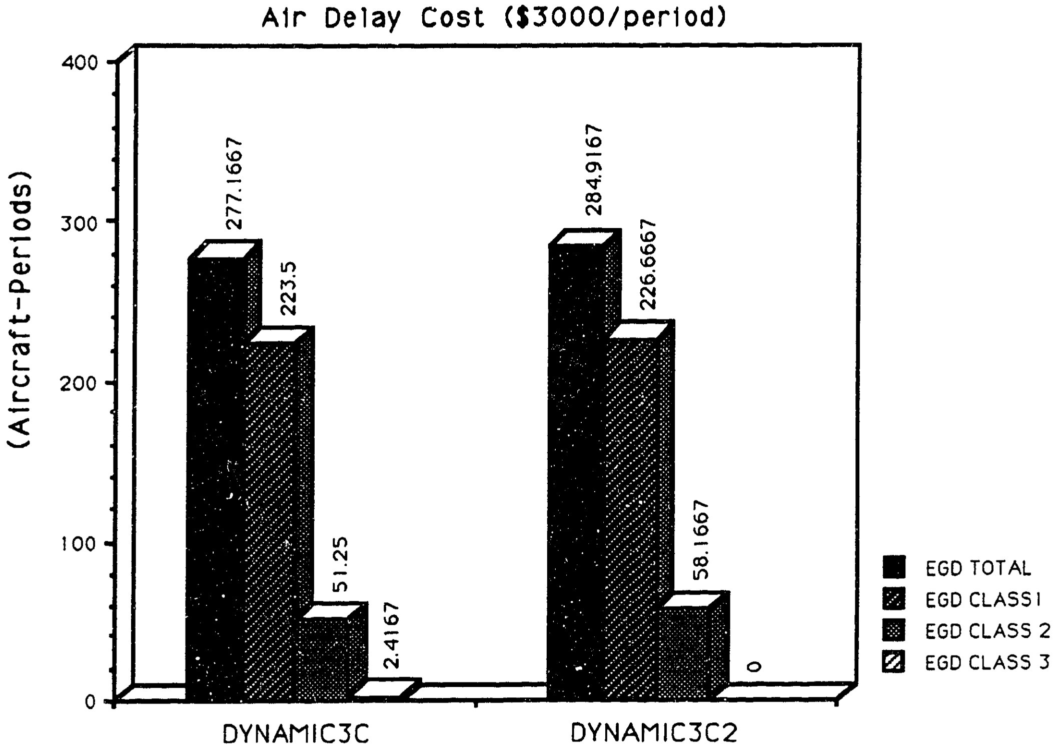


Figure 5.10

Average Expected Ground Delay By Aircraft Class

5.5 Comparing the Algorithms to the "Passive" Strategy

Total expected delay is minimized if all aircraft are allowed to depart as scheduled. This is because under this strategy there are no ground-holds and airport landing capacity utilization is maximized by letting aircraft arrive as early as possible (i.e., according to the original schedule). As a last exercise we compare the algorithms tested to the "passive" strategy of no ground-holds. The reason why ground-holding makes sense is that the cost of air delays is significantly higher than the cost of ground delays. Figure 5.11 shows average total expected delay (ground plus air) for the algorithms tested and the passive strategy. We see that the passive strategy indeed has the lowest expected delay. Unfortunately, these delays are all in the form of air delays.

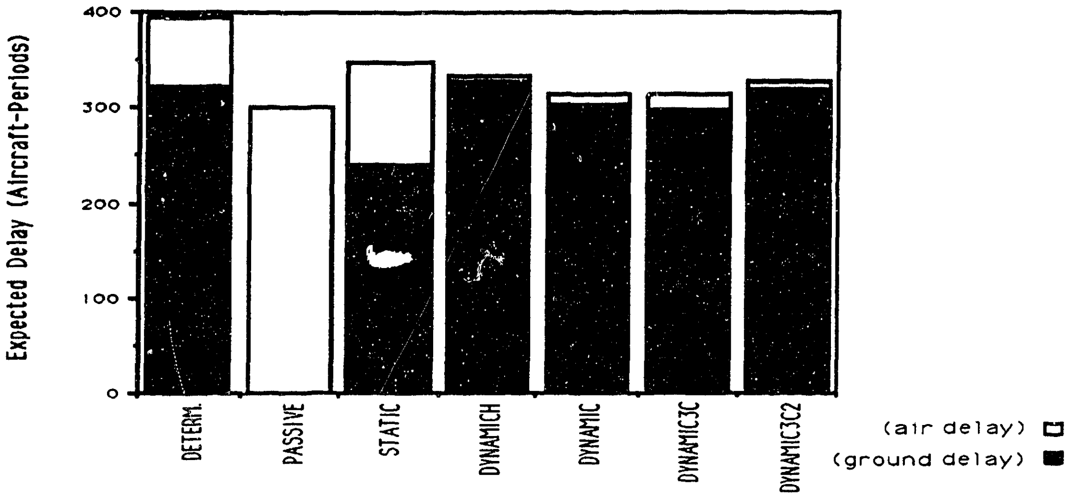


Figure 5.11
Average Expected Ground Plus Air Delay Including "Passive" Strategy

Figure 5.12 provides a breakdown of expected ground and air delay. In this figure it is evident that the dynamic algorithms have the advantage of producing remarkably low expected air delay while maintaining total delays within 10% of the minimum.

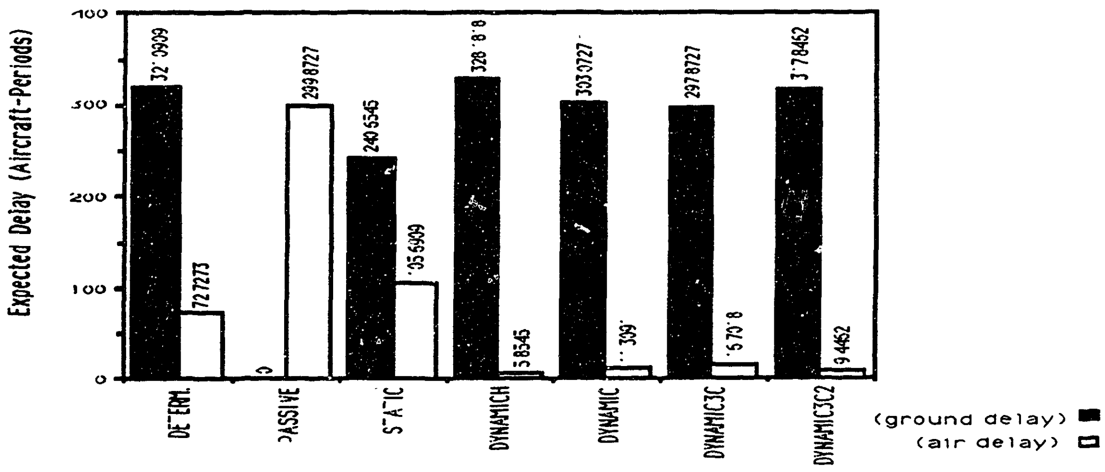


Figure 5.12
Average Expected Delay (Ground-Air Delay Breakdown)

Regarding cost performance, Figure 5.13 shows that, except for the case of a marginal air delay cost of $1200, the lowest air delay cost tested, the passive strategy performs worse than all the algorithms tested. This indicates that even simple ground-holding practices improve the efficiency of operations vis-a-vis a strategy that uses no ground-holds.

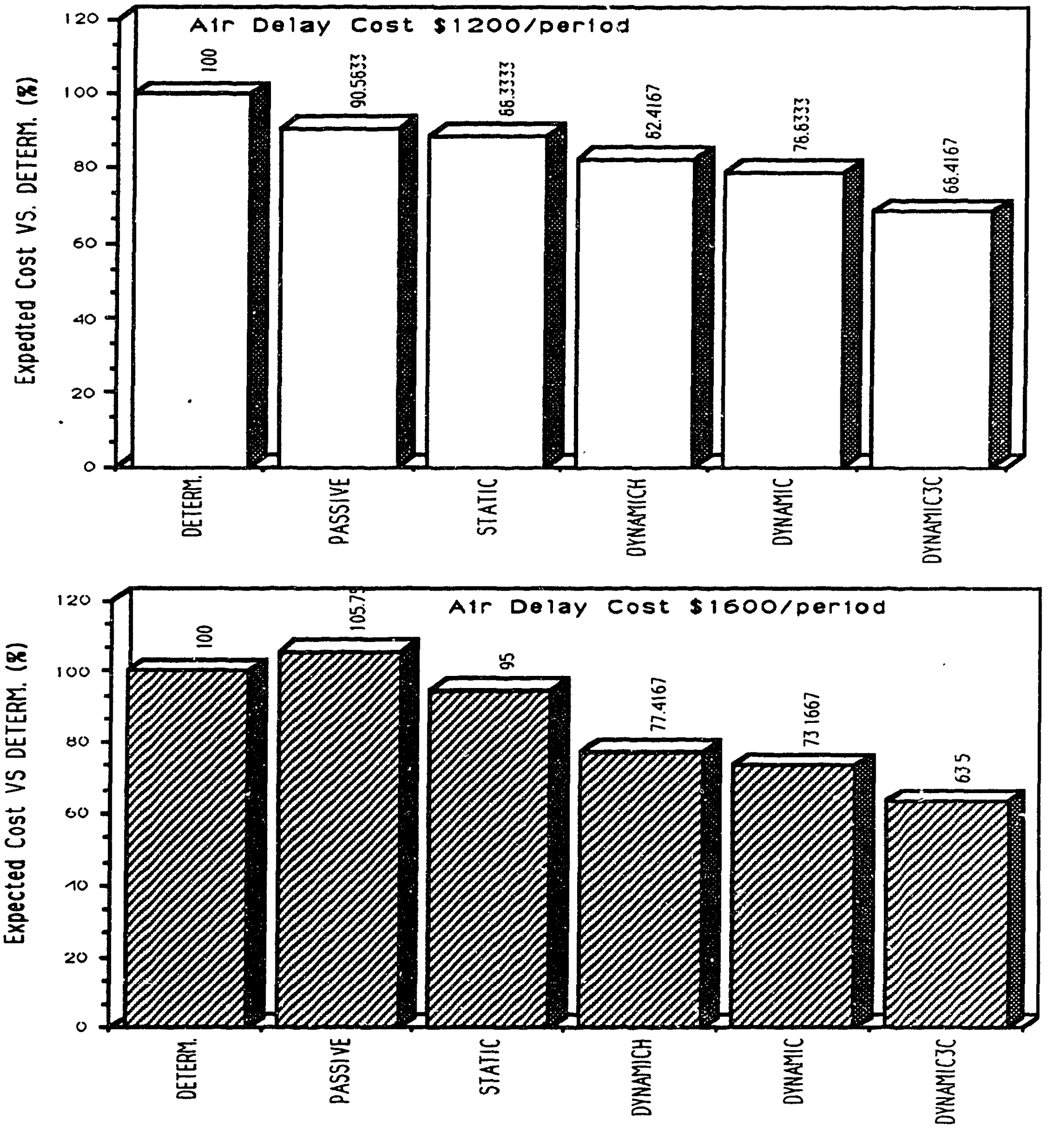


Figure 5.13a
Average Expected Delay Cost Including Passive - Air Cost: $1200 & $1600

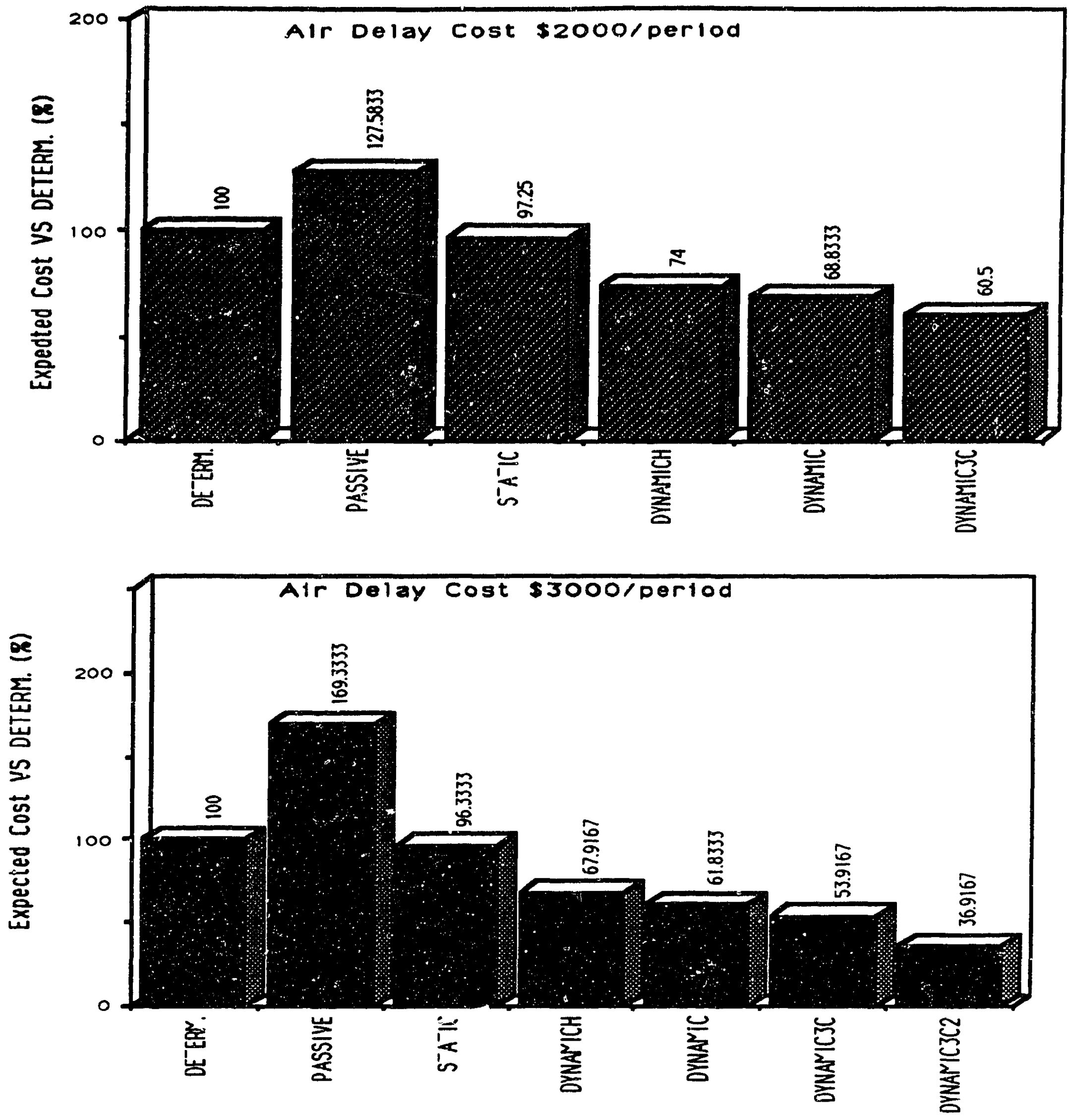


Figure 5.13b
Average Expected Delay Cost Including Passive - Air Cost: $2000 & $3000

Key conclusions from the model development and experimental work performed in this thesis are summarized in the next chapter. We will also discuss possible practical solution approaches to the "complete network" GHPP.

CHAPTER 6

6. CONCLUSION

The primary objective of this thesis was to develop static and dynamic algorithms that provide high quality solutions to practical instances of the probabilistic GHPP for the largest airports in the US ATC network. In this chapter we summarize key results in light of our primary objective, and bring these results into perspective by discussing conditions under which solutions based on the "star configuration" network are adequate when considering the complete air traffic network. Following this exercise, we address, in a preliminary way, the GHPP for the complete network by presenting a possible solution approach that uses one of the dynamic algorithms developed in this thesis as the key "building block". Finally, we suggest directions for future research.

The chapter is structured as follows: Section 6.1 summarizes key results obtained through model development and experimentation and discusses issues related to implementation of the algorithms developed. In 6.2, we discuss important characteristics of the Network GHPP that need to be considered when solving the network-wide problem. Section 6.3 presents a procedure to solve the GHPP for the complete network using the deterministic dynamic heuristic developed in Chapter 5. Finally, in Section 6.4 we propose future research steps.

6.1 Key Results

The principal contribution of this thesis is the development of algorithms that consider the probabilistic nature of airport landing capacity when solving the "star configuration" GHPP in air traffic control. Specifically, we developed static and dynamic algorithms capable of providing solutions to the GHPP, even for the largest airports in the US air traffic network. As mentioned in the early stages of this thesis the ground-holding strategies resulting from ground-holding algorithms are strategic in nature. An efficient ATC system should also utilize real-time, "fine tuning" ATC tools (such as the En Route Spacing , Departure Spacing and Arrival Spacing Programs, described in Chapter 1) as complements to an effective ground-holding program. In this section we start by providing a summary of the key results from the modelling and experimental work performed in this thesis. Then, we assess the practical limitations of the algorithms developed and discuss implementation issues.

6.1.1 Summary of Modelling and Experimental Results

In the early stages of model building we realized the advantages of dynamic solutions to the GHPP. The dynamic programming algorithm presented in Chapter 2 provides the exact solution to the dynamic probabilistic GHPP by exercising ground-hold control on individual aircraft at the beginning of each time period. The practicality of the algorithm was limited to small problems due to the exponentiality of the control variable (i.e., if n flights are eligible for delay, 2^n ground-hold strategies must be considered) and of the sample space size for the joint PMF of airport capacities. However, the exact modeling approach provided valuable insights, leading to a simplified model that captures the key elements of the real system.

The static and dynamic stochastic linear programming models developed in Chapters 3 and 4 are the counterparts to Terrab's static dynamic programming algorithm and the dynamic solution of Chapter 2 respectively. The main feature of the stochastic programming models is that they simplify the structure of the control mechanism and the airport capacity forecast by making ground-hold decisions on groups of aircraft (i.e., on aircraft classified according to cost class, and schedule) rather than individual flights; and considering few rather than many landing capacity profiles, in line with current weather forecasting technology. These models are able to provide solutions for realistic instances of the GHPP using just a personal computer. Another important aspect of these models is that important constraints, such as limiting maximum ground-holds and airborne queueing delays, are easily introduced.

The experimental work of this thesis yielded interesting findings. Overall, the probabilistic algorithms tested performed better than a deterministic algorithm roughly approximating current practice (Figures 5.4 and 5.8 provide overall cost and delay performance of the algorithms tested). However, the dynamic stochastic programming algorithms performed significantly better than static algorithms, highlighting the importance of dynamic approaches to solving the GHPP . This finding led to the development and testing of a "fast" heuristic (i.e., $O(T^2)$, where T is the total number of periods) which performed remarkably close to the dynamic stochastic programming solution, yielding over 20% savings by comparison to the deterministic algorithm.

Another important finding came from comparing the algorithms tested to the policy of no ground-holds, which minimizes total delay. Interestingly, the dynamic algorithms tested, including the heuristic, performed remarkably well compared to the "passive" strategy of no ground-holds. Total expected delays for these algorithms were within 10% of the minimum expected delay, with the advantage that over 95% of the delays are on the ground.

6.1.2 Practical Limitations and Implementation Issues

- *Dealing with the GHPP for the US Air Traffic Network*

When trying to use the algorithms developed in this thesis in the context of the complete air traffic system we need to consider the effect of network dependencies. These dependencies affect the aircraft schedule for daily operations, and - by implication - the quality of solutions to the GHPP for a "star configuration" network. However, under certain conditions, solutions provided by the algorithms for the single congested airport model are robust.

First, consider the case of a single congested airport in the network. A practical example of this case would be a day during which bad weather is expected at Chicago's O'Hare Airport, while relatively good weather is expected in the rest of the country. We see that since only a quite small fraction of the connecting flights through O'Hare will return back to O'Hare the same day, the solutions provided by the single airport model applied to O'Hare are adequate.

Suppose now that there are several congested airports in the network but only a relatively minor number of connecting flights between any pair of these airports. We see that in this case single airport models can again be used to solve the problems for each congested airport independently. However, when there is significant connecting flight traffic between congested airports, we need to consider network dependencies. In the Section 6.2 we will discuss important characteristics of the network-wide problem that must be considered in the development of exact and approximate solutions.

- *Assigning Ground-Holds to Specific Flights*

The algorithms for the single airport GHPP developed in this thesis assign ground-holds to groups of aircraft classified according to cost class and "last leg" flight schedule (i.e., the schedule for the flight segments into the congested airport). In order to assign ground-holds to specific flights we must refer back to the original schedule data for individual flights. In the absence of network dependencies, ground-holds are assigned randomly among the group of aircraft eligible for ground delay. However, when network dependencies are significant we need to consider these dependencies when assigning ground-holds. In Section 6.3 we present an approximate solution to the network GHPP that assigns ground-holds from the single airport solutions to specific flights by considering expected queueing delays at the congested airports within the ATC network.

- *Updating Solutions to the Dynamic GHPP*

When solving the GHPP dynamically, ground-holds are implemented several times throughout the day. Due to the probabilistic nature of airport landing capacities, the actual history of airport capacities will not coincide exactly with the airport capacities in the initial forecast. The forecast of airport landing capacities may also have been updated by the time we need to assign new ground-holds. Since the ground-holds resulting from the dynamic algorithms we have developed need to consider the current state of the ATC system and landing capacity forecast, we may be required to resolve the GHPP before implementing ground-holds for the next stage.

Worth noticing, taking a dynamic approach to ground-holding does not preclude the use of tactical ATC since, due to the stochastic environment, "fine tuning" ATC tools further improve the efficiency of operations.

- *Real-time Requirements*

Since ground-holding decisions must be implemented in real-time the speed of solution to the GHPP is critical. Running times for STATIC and DYNAMIC (the optimal static and dynamic algorithms for a single class of aircraft) were for most cases under 10 minutes and 70 minutes respectively. This is remarkable considering that we made no attempt to optimize the software/hardware combination used in solving the problems. Even with LINGO, the generic linear programming software we used, running times can be improved by at least 50% to 60% by using a Weitek co-processor and a 486 machine.

The deterministic dynamic heuristic should exhibit running times below one minute when implemented in the real ATC system[8].

The running times for DYNAMIC3C and DYNAMIC3C2, the optimal dynamic algorithms for three aircraft classes, were substantially higher than for the single class models, averaging 3.8 hours. However, the FAA's policy of "equal access" to all users of the ATC network that meet the navigational requirements of any sector in the system may limit application of our models to the faster single aircraft class algorithms.

- *Information Requirements*

Fortunately most of the information required by the algorithms developed in this thesis (i.e., input data) is available within the current US ATC system. We can readily access information on air traffic demand since updated airline schedule data is already being used by the CFCF in their EQF program. Updating the state of the

[8] In order to obtain the expected cost and expected delay statistics necessary to evaluate performance of the algorithms tested, the versions of the deterministic algorithm and dynamic heuristic implemented had running times comparable to STATIC (i.e, under 10 minutes).

system is also possible since real time information on the location of every aircraft within the ATC system is available.

Regarding the probabilistic landing capacity forecasts, these could be constructed using weather forecast information provided by the National Weather Service facilities at key ATC centers. Currently, weather forecast information is translated into a deterministic landing capacity forecast for key airports in the network. However, in order to incorporate probabilistic elements into the ground-hold assignment process, ATC management needs to be convinced about the advantages of a probabilistic approach to the problem. Assuming this is can be achieved, we believe that, in principle, the traffic management units within the ATC system should be able to adopt the simple forecasting system required by the probabilistic algorithms we have developed.

6.2 Important Features of the Network GHPP

- *Combinatorial Nature of the Network GHPP*

One of the complications arising in trying to build a model of the GHPP for the complete air traffic network is that we cannot group aircraft into broad cost classes when establishing ground-holds. We need to keep track of the identity of individual aircraft as they travel through the network. The following example illustrates this point.

Suppose two Boeing 747 aircraft are scheduled to arrive at Logan Airport at 2:00 PM and the next destinations in their schedules are Miami and Los Angeles, respectively. Suppose operating conditions at Logan Airport require that one of the two aircraft be delayed on the ground prior to its departure for Logan. In order to decide which aircraft is delayed into Logan, we need to consider expected air delays at Miami and Los Angeles as we should favor delaying the aircraft bound for the most congested terminal area. We see that in order to build a model for the GHPP that reflects network dependencies we must consider the combinatorial nature of the problem.

- *The Cost of Air Delay*

Since aircraft land in a fashion that approximates FCFS, the discussion of Chapter 3 regarding the cost of air delays holds for the network GHPP. Thus the marginal cost of air delay can be considered a constant - not necessarily identical for each congested airport.

- *The Effect of Time Horizon on Network Dependencies*

The effect of network dependencies diminishes as the time horizon for the GHPP is shortened. This is because the number of connections per aircraft declines as we reduce the time span for the problem. We see that for a one hour time horizon, network dependencies are practically non-existent. While for time horizons of two to three hours network dependencies are probably limited to airport pairs, depending on the proximity and traffic patterns between congested airports.

From the discussion above we see that solutions to exact models for the network GHPP have a combinatorial nature. However, by periodically updating the state of the system in regards to aircraft schedule, we could use one of the dynamic algorithms developed in this thesis in order to produce reasonably good solutions for the network GHPP. In the next section we present a possible procedure to solve the network GHPP using the "fast" dynamic heuristic developed in this thesis.

6.3 Approximate Solution to the Network GHPP

The four step procedure discussed here is based on the deterministic dynamic heuristic developed in Chapter 5. It is applicable to instances of the GHPP that cannot be modeled by a "star configuration" network. The rationale for the procedure is provided by the discussion in Sections 6.2. This first proposal for solving the network GHPP is a simple heuristic procedure. As is the case for all heuristics, it might be improved by introducing refinements following experimentation.

Step 1: We identify a set of airports for which weather-related congestion is anticipated for the day of operations and which exhibit a significant degree of connecting flight dependency. A probabilistic capacity forecast of the form described in Chapter 4 is provided independently for each congested airport. We also define stage durations. For example we may decide on three-hour stages which would result in 5 to 7 instances during the day at which ground-holds are determined. The rationale behind choosing a relatively short stage duration is trying to limit network dependencies within any given stage to airport pairs.

Step 2: We calculate expected air queueing delay for every time period of operations for every congested airport independently. This is easily done using the capacity forecast and the original landing schedule (i.e., assuming no ground-holds) for each congested airport. The information on expected air delays at congested airports will help in the assignment of ground-holds by the deterministic heuristic to specific flights by considering expected congestion on the next leg of a flight eligible for ground-holding (i.e., ground delays are assigned to flights expected to encounter the greatest air queueing delays in their next leg).

Step 3: At each stage:

- we generate a solution to the GHPP for every congested airport in the network independently, using the deterministic dynamic heuristic of Chapter 5. Inputs to the algorithm are based on the current state of the system (i.e., an updated demand profile for each congested airport derived from information on current aircraft current location in the network, past ground hold decisions and the original schedule), and an updated landing capacity forecast.

- we update expected air queueing delay at each of the congested airports for the remaining time periods using an updated demand profile for each congested airport - derived from information on aircraft current location in the network, ground hold decisions for previous stages and the original schedule - and an updated landing capacity forecast.

Step 4: The deterministic dynamic heuristic determines ground-holds for groups of aircraft. Therefore, at each stage we need to assign ground- holds to specific aircraft. For this we use the information on expected air queueing delay at congested airports found in step 3. Due to the relatively short stage duration, we can limit our horizon to the next leg of a connecting flight eligible for ground-holding as determined by the deterministic dynamic heuristic solution. Once ground-holds for the current stage are assigned to specific flights we proceed with step 3 for the next stage. The example below illustrates this procedure:

If in the two Boeing 747 example mentioned in Section 6.2, we assume that Logan, Miami, and Los Angeles are the congested airports in the network which exhibit connecting flight dependencies. The deterministic dynamic heuristic, applied to Logan, would only indicate that one of the two aircraft into Logan must be subjected to ground delay (e.g., 30 minutes of ground delay prior to departure for Logan). We then use the information on expected air queueing delay at the arrival airport in the next leg of these flights in order to assign the ground delay to a specific aircraft. For example, suppose that for the aircraft continuing to Miami, and for the originally scheduled arrival time, the expected air queueing delay in Miami is 30 minutes; while for the Los Angeles flight there is not expected air queueing delay at the scheduled arrival time. Then, the aircraft going to Miami in the next leg would be the one subject to the 30 minute ground-holding prior to its departure for Logan.

We notice that limiting the time horizon for the network GHPP may significantly affect the performance of the proposed heuristic as we only consider the next leg of connecting flights when assigning ground-holds to specific aircraft. Therefore, we should not rule out the development of exact models for the network GHPP. Even if these exact models can not be implemented in practice, they can help in the assessment of heuristics based on single airport models. Furthermore, heuristics derived from exact network algorithms may show better performance than heuristics based on single airport models by providing a better model of the ATC network. In the next section, we propose possible research directions for the GHPP.

6.4 Future Research

Future research on the GHPP can be classified as ongoing research on the "star configuration" problem and initial modelling/experimentation on the network-wide problem.

- *Continued Research on the Single Airport GHPP*

We have made substantial progress on solving probabilistic static and dynamic versions of the single airport GHPP for practical size problems. The experimental results in this thesis indicate that, if implemented, these algorithms could generate substantial savings vis-a-vis current ground-holding practices. However, the algorithms need to be validated through testing in the actual US ATC system. In addition, the implementation issues discussed in Section 6.1 must be assessed. An experimental program at one or several airports in the US ATC network can provide a test bed for the single airport probabilistic algorithms we developed. Notice that since the forecasts of airport landing capacity for each airport are provided independently, we can use probabilistic ground-holding algorithms for the airports in the test while maintaining the current CFCF ground-holding program at other airports.

- *Possible Research Directions for the Network GHPP*

Future research on the network GHPP can be classified according to the approach to modelling. As discussed in section 6.3, one possibility is the development and testing of heuristics that use the single airport probabilistic algorithms as "building blocks". The advantage of this approach is that it provides simple solutions to the network problem capitalizing on previous research results. However, these heuristics may not provide an adequate model for the network problem since they are based on algorithms that ignore network dependencies.

On the other hand, we can try to develop exact models of the network GHPP that fully reflect the combinatorial nature of the problem. and then develop simplified versions of these algorithms that can solve problems of realistic size. Once exact models are developed, research should concentrate on producing algorithms that can be used in the solution of practical instances of the network GHPP.

Given the complexity of the network problem, initial efforts should probably concentrate on solving deterministic versions of the problem - similar to the research program for the "star configuration" network that led to the successful solution to the fully dynamic probabilistic solution to the problem presented in Chapter 2 and the development of algorithms capable of solving large problems.

We believe that future research efforts should not rule out any of the directions mentioned above as all of them have the potential of providing significant contributions in solving the GHPP for the complete ATC network. From the discussion in this chapter we conclude that in order to implement effective ground-holding policies for the complete ATC network future research should: (i) develop algorithms that address probabilistic aspects of the problem, have a dynamic nature, and consider network dependencies; (ii) consider issues such as ease of implementation, information requirements and speed of solution; and (iii) integrate decision support/data management systems that complement any optimization algorithms implemented within the ATC system.

References

1. Gish, J. "FAA Hopes Technology Can Increase Capacity." Airline Executive International, March 1990.

2. Kulikowski, A. J. "Traffic Flow Management in Today's ATC System." *Journal of ATC*, January-March 1987.

3. Reilly, D. J. Presentation During Second Working Session at the 27th ICAA World Congress, *Airport Forum* 5, 1987.

4. Donoghue, J. A. "A Numbers Game." Editorial published in *Air Traffic World*, December 1986.

5. Del Balzo, J. M. "The US National Airspace System for the Year 2010." *Journal of ATC*, July-September 1989.

6. Mornberger, M. "Exit: A Decade of Aircraft Purchases Enter: The Decade of the Airport." Editorial published in *Airport Forum*, January 1990.

7. Wolley D. "Pressure Building To Free Eurocontrol From Political Constraints." Airline Executive International, March 1990.

8. Bjorek, A. Presentation during the October 1989 European Civil Austrian Conference (ECAC) Symposium, *Airport Forum*, January 1990.

9. Odoni, A. R. "The Flow Management Problem in Air Traffic Control." In *Flow Control of Congested Networks*, A. R. Odoni, L. Bianco and G. Szego (eds.), Springer Verlag, New York, pp. 269-288, 1987.

10. Cohen, D. and Odoni, A.R. "A Survey of Approaches to the Airport Slot Allocation Problem." Flight Transportation Laboratory Report 85-3, MIT, Cambridge, MA, 1985.

11. Terrab, M. "Ground Holding Strategies in Air Traffic Control." Operations Research Center Technical Report No. 196, MIT, Cambridge, MA, February 1990.

12. Andreatta, G., and Romanin-Jacur, G. "Airport Flow Management Under Congestion." *Transportation Science* 21:249-253, 1987.

13. Bertsekas, D.P. *Dynamic Programming,* Prentice-Hall Inc., NJ, 1987.

14. Wagner, H. M. *Principles of Operations Research,* Prentice Hall, Inc., NJ, 1975.

15. Bradley, S. P., Hax, A. C., and Magnanti, T. L. *Applied Mathematical Programming,* Addison-Wesley Publishing Company, Inc., 1977

APPENDIX 1

DYNAMIC PROGRAMMING ALGORITHM:

1A. INPUT FILE FORMAT:

-Start with four integers:
-Period 0 capacity k_0 (0 to 5)
-Number of planes N (0 to 5)
-Maximum capacity M (0 to 5)
-Number of periods P+1 (0 to 5)

-Conditional PMF's (float numbers. Need a period after integer entries):

for i =0,1,...,M write:

$p(0)/_i$ $p(1)/_i$ ... $p(M)/_i$

-Departure and arrival times for each flight (integers):

for i=1,2,...,N write:

departure time of i   arrival time of i

-Ground and air delay costs for each flight (integers):

for i=1,2,....N write:

$cg_i(1)$ $ca_i(1)$ $cg_i(2)$ $ca_i(2)$ ...... $cg_i(P)$ $ca_i(P)$

1B. C CODE:

```
/*Dynamic DP algorithm for  the FMP size 5 problem*/
/******************************Global section*******************/
#include <stdio.h>
#include <stdlib.h>
/*************************Global Macros****************************/
#define MALLOC(x) ((x*)malloc(sizeof(x))) /*use:ptr-MALLOC(type)*/
/************************Global Constant declarations*****************/
#define MAXN 5 /*number of planes 0,1,2,3,4.(3 has highest priority)*/
#define MAXM 5 /* max airport capacity*/
#define MAXP 5 /* number of finite capacity time periods*/
/*********************Global Type Declarations**********************/
```

```
typedef struct state
(char cap[MAXP];
char g_hold[MAXN];
char feasible[MAXN];
float opt_cost;
struct state * next_stage;
struct state * next_state;
)
state;
typedef struct queue
(char f_number;
struct queue * next;
)
queue;
typedef struct feas_cap
(char cap;
struct feas_cap * next;
)
feas_cap;

/************************Global variable declarations******************/
long int states_n-0;                          /*number of states*/
int first_cap,N,M,P;                          /*period 0 capacity, #planes, max capacity,#
periods*/
float p[MAXM+1][MAXM+1];                      /*conditional PMF's. p[i][j]-p(i)/j*/
int L[MAXN], A[MAXN];                         /*departure and arrival times*/
int cg[MAXN][MAXP],ca[MAXN][MAXP];            /*ground and air marginal costs*/
state first_state[MAXP];            /*first state in each stage*/
feas_cap *head_ptr[MAXM+1];         /*pointers to nonzero prob. capacities list*/
char counter[MAXM+1];                         /*number of non-zero prob. capacities*/
queue * head-NULL;                            /*ptr to first in queue to land*/
/************************Funtions Section******************************/

FILE * openfile (filename, type, defaultfile, defaultname)
/*opens file, performs validation*/
char *filename, *type, *defaultname;
FILE *defaultfile;
(FILE *temp;
if (lstrcmp(filename,defaultname))
temp-defaultfile;
else
temp-fopen(filename, type);
if (temp--NULL)
(printf("\nError - can't open \"%s\"", filename);
printf("\nUsing default:\"%s\"", defaultname);
temp-defaultfile;
)
return(temp);
)
void create_state_space(void)
(char i,j,k;
state *current, *current_next, *prev_next;
feas_cap *cap_ptr, *current_cap_ptr;
```

```
/*generate linked list of non_zero capacities*/
for (i-0; i<-M;i++)
(counter[i]-0;
for (j-0;j<-M;j++)
if (p[i][j]>0)
(counter[i]++;
head_ptr[i]-MALLOC(feas_cap);
if(head_ptr[i]--NULL)
(printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
)
head_ptr[i]->cap-j;
head_ptr[i]->next-NULL;
break;
)
current_cap_ptr-head_ptr[i];
for (j-j+1;j<-M;j++)
if (p[i][j]>0)
(counter[i]++;
current_cap_ptr->next-MALLOC(feas_cap);
if(current_cap_ptr->next--NULL)
(printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
)
current_cap_ptr-current_cap_ptr->next;
current_cap_ptr->cap-j;
current_cap_ptr->next-NULL;
)
)

/*start with first_state[0]*/
first_state[0].cap[0]-first_cap;
for(i-0;i<-N-1;i++)
(first_state[0].g_hold[i]-0;
if(L[i]--0)
first_state[0].feasible[i]-1;
else
first_state[0].feasible[i]-0;
)
first_state[0].next_state-NULL;
for (i-0; i<-P-2;i++) /*generate states for stage i+1*/
(/*start at top of stage*/
current-&first_state[i];
current_next-&first_state[i+1];

/*Create states for each feasible strategy. One for each nonzero capacity*/
while(current!-NULL)
(/*start with strategy 0.handle first state separately*/
if (current!-&first_state[i])
current_next-prev_next->next_state-MALLOC(state);
if(current_next--NULL)
(printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
)
```

```
states_n++;
current->next_stage=current_next;
cap_ptr=head_ptr[current->cap[i]];
for(j=0;j<=i;j++)
current_next->cap[j]=current->cap[j];
current_next->cap[i+1]=cap_ptr->cap;
current_cap_ptr=cap_ptr->next;

for(k=0;k<=N-1;k++)
{current_next->g_hold[k]=current->g_hold[k];
if(L[k]==i+1)
current_next->feasible[k]=1;
else
current_next->feasible[k]=0;
}
current_next->next_state=NULL;
prev_next=current_next;


while(current_cap_ptr!=NULL) /*do all nonzero caps*/
{current_next=prev_next->next_state=MALLOC(state);

if(current_next==NULL)
{printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
}
states_n++;
for(j=0;j<=i;j++)
current_next->cap[j]=current->cap[j];
current_next->cap[i+1]=current_cap_ptr->cap;
for(k=0;k<=N-1;k++)
{current_next->g_hold[k]=prev_next->g_hold[k];
current_next->feasible[k]=prev_next->feasible[k];
}
current_next->next_state=NULL;
prev_next=current_next;
current_cap_ptr=current_cap_ptr->next;
}

/*strategy 00001*/
if(current->feasible[0]==1&&((current->g_hold[0]+A[0])<=P-1))
{current_cap_ptr=cap_ptr; /*move current cap ptr back*/
current_next=prev_next->next_state=MALLOC(state);
if(current_next==NULL)
{printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
}
states_n++;
for(j=0;j<=i;j++)
current_next->cap[j]=current->cap[j];
current_next->cap[i+1]=current_cap_ptr->cap;
for(k=0;k<=N-1;k++)
{if(k==0)
{current_next->g_hold[k]=current->g_hold[k]+1;
current_next->feasible[k]=1;
}
else
{current_next->g_hold[k]=current->g_hold[k];
if(L[k]==i+1)
current_next->feasible[k]=1;
else
current_next->feasible[k]=0;
}
}
current_next->next_state=NULL;
prev_next=current_next;
current_cap_ptr=current_cap_ptr->next;
while(current_cap_ptr!=NULL) /*do all nonzero caps*/
{current_next=prev_next->next_state=MALLOC(state);
if(current_next==NULL)
{printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
}
states_n++;
for(j=0;j<=i;j++)
current_next->cap[j]=current->cap[j];
current_next->cap[i+1]=current_cap_ptr->cap;
for(k=0;k<=N-1;k++)
{current_next->g_hold[k]=prev_next->g_hold[k];
current_next->feasible[k]=prev_next->feasible[k];
}
current_next->next_state=NULL;
prev_next=current_next;
current_cap_ptr=current_cap_ptr->next;
}
}

if(N>=2)
{
/*strategies : 00010, 00011, 00100,  00110, 00111, 01000, 01001, 01010, 01011, 01100,
01101, 01110, 01111, 10000, 10001, 10010, 10011, 10100, 10101, 10110, 10111,
11000,11001, 11010, 11011, 11100,11101, 11110, 11111 have been ommited as they have a
similar structure*/
}
}
void delete_queue( char cap)
{queue * temp;
char i;
for (i=1; i<=cap; i++)
{if (head !=NULL)
{
temp=head->next;
free((void*)head);
head=temp;
}
else
return ;
}
```

```
return ;
}
void add_queue( char j)
{queue * current,* current_next, * temp;
if (head--NULL)                    /*an empty queue*/
{head-MALLOC(queue);
if(head--NULL)
{printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
}
head->f_number-j;
head->next-NULL;
}
else                           /*not an empty queue*/
{current-head;
while(current->next!-NULL&&current->f_number>j)
{
current-current->next;
}
if (current->next--NULL)
{
if (current->f_number<j)
{temp-current;
current-MALLOC(queue);
if(current--NULL)
{
printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
}
if(temp--head)
head-current;
current->next-temp;
current->f_number-j;
}
else
{
temp-current->next;
current->next-MALLOC(queue);
if(current->next--NULL)
{
printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
}
current->next->next-temp;
current->next->f_number-j;
}
}
else
{
temp-current;
current-MALLOC(queue);
if(current--NULL)
{
printf("run out of memory. Exit program\n");
scanf("%d",first_cap);
}
if(temp==head)
head=current;
current->next=temp;
current->f_number=j;
}
}
}

int find_air_cost (int period, state * current)
{
int cost=0;
char i,j;
queue *q_current;
for (i=1;i<=period;i++)
{
for (j=N-1; j>=0;j--)
if (A[j]+current->g_hold[j]==i)
add_queue (j);
if (current->cap[i]>0)
delete_queue(current->cap[i]);
}
q_current=head;
while(q_current!=NULL)
{
cost+=ca[q_current->f_number][period + 1 - (A[q_current->f_number]+current-
>g_hold[q_current->f_number])];
q_current=q_current->next;
delete_queue(1);
}
return cost;
}
void find_optimal_cost(void)
{
char i,j;
state *start, *current, *current_next, *first_next,*min;
int air_cost;
float min_cost,cost;

/*calculate opt_cost for last stage*/

current=&first_state[P-1];

while (current!=NULL)
{
current->opt_cost=find_air_cost(P-1,current);
current=current->next_state;
}

/* find opt_cost for other stages*/

for( i=P-2; i>=0;i--)
{
```

```
current=&first_state[i];
current_next=current->next_stage;
if(current->next_state==NULL)
first_next=NULL;
else
first_next=current->next_state->next_stage;
while(current!=NULL)
{
/*air cost same for all strategies*/
air_cost=find_air_cost(i,current);
min_cost=air_cost;
min=current_next;

/*look strategy 0 set it to min. notice ground cost is zero*/

for (j=1; j<=counter[current->cap[i]];j++)
{
min_cost+=current_next->opt_cost*p[current_next->cap[i]][current_next->cap[i+1]];
current_next=current_next->next_state;
}

while(current_next!=NULL&&current_next!=first_next)
{

cost=air_cost;

/*add ground cost*/
for(j=0;j<=N-1;j++)
if(current->feasible[j]*current_next->feasible[j]==1)
cost+=cg[j][current_next->g_hold[j]];

/*add exp opt cost all cap cases*/

start=current_next;
for (j=1; j<=counter[current->cap[i]];j++)
{
cost+=current_next->opt_cost*p[current_next->cap[i]][current_next->cap[i+1]];
current_next=current_next->next_state;
}
if(cost<min_cost)
{
min_cost=cost;
min=start;
}
}
current->opt_cost=min_cost;
current->next_stage=min;
current=current->next_state;
current_next=current->next_stage;
if(current->next_state==NULL)
first_next=NULL;
else
first_next=current->next_state->next_stage;
}
```

```
)
)

main()
(
char in_file[20],out_file[20];
int i,j,k,l,period-0,cap, count;
FILE * fpo;
FILE * fpi;
state * current, * temp, *prev;
printf("\n input filenames for inputs and outputs respectively\n");
scanf("%s%s",in_file,out_file);
fpi-openfile(in_file, "r", stdin, "stdin");
fpo-openfile(out_file,"w", stdout,"stdout");

/*read input data*/
fscanf(fpi, "%d%d%d%d",&first_cap,&N,&M,&P);
printf("Inputs are:\nPeriod 0 capacity is: %d.\nThe number of planes is: %d"
"\nThe maximum capacity is: %d \nThe number of periods is: %d",first_cap,N,M,P);

fprintf(fpo,"Inputs are:\nPeriod 0 capacity is: %d.\nThe number of planes is: %d"
"\nThe maximum capacity is: %d \nThe number of periods is: %d",first_cap,N,M,P);

printf("\n\nThe conditional PMF'S are:");
fprintf(fpo,"\n\nThe conditional PMF'S are:\n");

for(i-0;i<-M;i++)
(
printf("\n");
fprintf(fpo,"\n");
for(j-0;j<-M;j++)
(
fscanf(fpi,"%f",&p[i][j]);
printf("\t p(%d/%d)-%.2f",j,i,p[i][j]);
fprintf(fpo,"\t p(%d/%d)-%.2f",j,i,p[i][j]);
)

)

printf("\n\n\nThe scheduled departure and arrival times are:\n\n");
fprintf(fpo,"\n\n\nThe scheduled departure and arrival times are:\n\n");
for(i-0; i<-N-1;i++)
(
fscanf(fpi,"%d%d",&L[i],&A[i]);
printf("Plane %d: departure time is: %d.\t Arrival time is: %d\n\n",i+1,L[i],A[i]);
fprintf(fpo,"Plane %d: departure time is: %d.\t Arrival time is: %d\n\n",i+1,L[i],A[i]);
)

printf("\n\nThe ground  and air marginal costs for each plane are (plane, # of periods):\n");
fprintf(fpo,"\n\nThe ground  and air marginal costs for each plane are(plane, # of
periods):\n");


for(i-0; i<-N-1;i++)
```

```
{
printf("\n");
fprintf(fpo,"\n");
for(j=1;j<=P-1;j++)
{
fscanf(fpi,"%d%d",&cg[i][j],&ca[i][j]);
printf("ground cost(%d, %d)=%6d\t air cost(%d, %d)=%6d\n\n",i+1,j,cg[i][j],i+1,j,ca[i][j]);
fprintf(fpo,"ground cost(%d, %d)=%6d\t air cost(%d, %d)=%6d\n\n",i+1,j,cg[i][j],i+1,j,ca[i][j]);
}
}
fclose(fpi);
create_state_space();
find_optimal_cost();

/*retrieve solution*/

fprintf(fpo,"\n\nSolution:\n\nThe number of states is: %ld\n\nThe exp. cost for opt. dynamic
strategy is: $%.2f\n\n",states_n,first_state[0].opt_cost);
printf("\n\nSolution:\n\nThe number of states is: %ld\n\nThe exp. cost for opt. dynamic
strategy is: $%.2f\n\n",states_n,first_state[0].opt_cost);
fprintf(fpo,"The optimal delay strategy for period 0 is(0 is do not delay, 1 is delay):\n\n");
printf("The optimal delay strategy for period 0 is(0 is do not delay 1 is delay):\n\n");

for(i=0;i<=N-1;i++)
{
fprintf(fpo,"\nPlane %d strategy:
%d.\n",i+1,first_state[0].feasible[i]*first_state[0].next_stage->feasible[i]);
printf("\nPlane %d strategy: %d.\n",i+1,first_state[0].feasible[i]*first_state[0].next_stage-
>feasible[i]);
}
fprintf(fpo,"\n\n");
printf("\n\n");
l=1;
printf("\n do you want to get results for next time period? (enter 1 if yes 0 if no).\n");
scanf("%d",&j);
while(l==1)
{
prev=&first_state[0];
current=first_state[0].next_stage;
period=0;

while(j==1)
{
if(period==P-1)
{
printf("\n sorry previous period was the final period\n");
break;}

period++;
k=1;
temp=current;
while(k==1)
{current=temp;
count=0;
printf("\n enter capacity for period %d:",period);
scanf("%d",&cap);
while(current->cap[period]!=cap)
{current=current->next_state;
count++;
if(count>=counter[prev->cap[period-1]])
{current=temp;
printf("\nSorry. Capacity of %d is not  valid here (i.e. prob 0 event). Try again.\n", cap);
break;}
}
if(count<counter[prev->cap[period-1]])
{fprintf(fpo,"\n\nThe exp. cost for opt. dynamic strategy for period %d capacity of %d is:
$%.2f\n\n",period,cap,current->opt_cost);
printf("\n\nThe exp. cost for opt. dynamic strategy for period %d capacity of %d is:
$%.2f\n\n",period,cap,current->opt_cost);
fprintf(fpo,"The optimal delay strategy is: \n\n",period);
printf("The optimal delay strategy is:\n\n");
for(i=0;i<=N-1;i++)
{if(period==P-1)
{fprintf(fpo,"\nPlane %d strategy:  0.\n",i+1);
printf("\nPlane %d strategy:  0.\n",i+1);
}
else
{fprintf(fpo,"\nPlane %d strategy:  %d.\n",i,current->feasible[i]*current->next_stage-
>feasible[i]);
printf("\nPlane %d strategy:  %d.\n",i,current->feasible[i]*current->next_stage->feasible[i]);
}
}
printf("do you want to try another capacity for period %d? (enter 1 if yes 0 if
not):\n",period);
scanf("%d",&k);
}
}
prev=current;
current=current->next_stage;
fprintf(fpo,"\n\n");
printf("\n\n");
printf("\n\n do you want to get results for next time period? (enter 1 if yes 0 if no.\n");
scanf("%d",&j);
}
printf("\nDo you want to try another capacity sample path starting with period 1?. (enter 1 if
yes 0 if not)\n");

scanf("%d",&l);
j=1;
}
}
```

1C. "SIZE 5" EXAMPLE: LARGEST SOLVED ON CRAY 2

Inputs are:
Period 0 capacity is: 4.
The number of planes is: 5
The maximum capacity is: 4

The number of periods is: 5

The conditional PMF'S are:

p(0/0)-0.60	p(1/0)-0.10	p(2/0)-0.10	p(3/0)-0.10	p(4/0)-0.10
p(0/1)-0.20	p(1/1)-0.50	p(2/1)-0.10	p(3/1)-0.10	p(4/1)-0.10
p(0/2)-0.10	p(1/2)-0.10	p(2/2)-0.60	p(3/2)-0.10	p(4/2)-0.10
p(0/3)-0.10	p(1/3)-0.10	p(2/3)-0.20	p(3/3)-0.50	p(4/3)-0.10
p(0/4)-0.10	p(1/4)-0.10	p(2/4)-0.10	p(3/4)-0.20	p(4/4)-0.50

The scheduled departure and arrival times are:

Plane 1: departure time is: 0. Arrival time is: 1
Plane 2: departure time is: 0. Arrival time is: 1
Plane 3: departure time is: 0. Arrival time is: 1
Plane 4: departure time is: 0. Arrival time is: 1
Plane 5: departure time is: 0. Arrival time is: 1

The ground and air marginal costs for each plane are(plane, # of periods):

ground cost(1, 1)-	1	air cost(1, 1)-	1000
ground cost(1, 2)-	2	air cost(1, 2)-	1100
ground cost(1, 3)-	3	air cost(1, 3)-	1200
ground cost(1, 4)-	4	air cost(1, 4)-	1300
ground cost(2, 1)-	3	air cost(2, 1)-	2000
ground cost(2, 2)-	4	air cost(2, 2)-	2100
ground cost(2, 3)-	5	air cost(2, 3)-	2200
ground cost(2, 4)-	6	air cost(2, 4)-	2300
ground cost(3, 1)-	6	air cost(3, 1)-	3000
ground cost(3, 2)-	7	air cost(3, 2)-	3100
ground cost(3, 3)-	8	air cost(3, 3)-	3200
ground cost(3, 4)-	9	air cost(3, 4)-	3300
ground cost(4, 1)-	9	air cost(4, 1)-	4000
ground cost(4, 2)-	10	air cost(4, 2)-	4100
ground cost(4, 3)-	11	air cost(4, 3)-	4200
ground cost(4, 4)-	12	air cost(4, 4)-	4300
ground cost(5, 1)-	13	air cost(5, 1)-	5000
ground cost(5, 2)-	14	air cost(5, 2)-	5100
ground cost(5, 3)-	15	air cost(5, 3)-	5200
ground cost(5, 4)-	16	air cost(5, 4)-	5300

Solution:

The number of states is: 2087360
The exp. cost for opt. dynamic strategy is: $158.00
The optimal delay strategy for period 0 is(0 is do not delay, 1 is delay):
Plane 1 strategy: 1.
Plane 2 strategy: 1.
Plane 3 strategy: 1.
Plane 4 strategy: 1.
Plane 5 strategy: 1.

The exp. cost for opt. dynamic strategy for period 1 capacity of 2 is: $126.00

```
The optimal delay strategy is:
Plane 0 strategy:  1.
Plane 1 strategy:  1.
Plane 2 strategy:  1.
Plane 3 strategy:  1.
Plane 4 strategy:  1.

The exp. cost for opt. dynamic strategy for period 2 capacity of 2 is:  $89.00
The optimal delay strategy is:
Plane 0 strategy:  1.
Plane 1 strategy:  1.
Plane 2 strategy:  1.
Plane 3 strategy:  1.
Plane 4 strategy:  1.

The exp. cost for opt. dynamic strategy for period 3 capacity of 2 is:  $47.00
The optimal delay strategy is:
Plane 0 strategy:  1.
Plane 1 strategy:  1.
Plane 2 strategy:  1.
Plane 3 strategy:  1.
Plane 4 strategy:  1.

The exp. cost for opt. dynamic strategy for period 3 capacity of 4 is:  $47.00
The optimal delay strategy is:
Plane 0 strategy:  1.
Plane 1 strategy:  1.
Plane 2 strategy:  1.
Plane 3 strategy:  1.
Plane 4 strategy:  1.

The exp. cost for opt. dynamic strategy for period 4 capacity of 2 is:  $0.00
The optimal delay strategy is:
Plane 1 strategy:  0.
Plane 2 strategy:  0.
Plane 3 strategy:  0.
Plane 4 strategy:  0.
Plane 5 strategy:  0.
```

APPENDIX 2

PROBLEM DATA

2A. OFICIAL AIRLINE GUIDE DATA FOR LOGAN (NOVEMBER' 88):

COLUMN 1: EQUIPMENT;
COLUMN 2: AIRCRAFT TYPE;
COLUMN 3: DEPARTURE TIME;
COLUMN 4: DURATION;
COLUMN 5: ARRIVAL TIME;
COLUMN 6: AIRPORT OF DEPARTURE;

1	2	3	4	5	6
DO8	1	5.75	0.5	6.25	MHT
DO8	1	6	0.67	6.67	LEB
DO8	1	6	0.67	6.67	PWM
BE1	1	6.36	0.67	7.03	LEB
DO8	1	6.75	0.5	7.25	MHT
BE1	1	6.82	0.5	7.32	MVY
D9S	2	6.38	1	7.38	BTV
SH6	1	6.75	0.67	7.42	HYA
PAG	1	6.92	0.5	7.42	PVC
BE1	1	6.95	0.5	7.45	MHT
BE1	1	6.62	0.88	7.5	BDR
DO8	1	6.67	0.83	7.5	ALB
BE1	1	6.83	0.67	7.5	HYA
CNA	1	6.83	0.67	7.5	LCI
SH6	1	6.77	0.75	7.52	ACK
72S	2	6.52	1	7.52	BGR
72S	2	6.52	1	7.52	LGA
SF3	1	6.07	1.5	7.57	PQI
72S	2	6.9	0.67	7.57	PWM
BEC	1	7.08	0.5	7.58	MVY
D9S	2	6.65	1	7.65	EWR
DO8	1	6.67	1	7.67	ISP
72S	2	6.5	1.17	7.67	PHL
SF3	1	6.73	1	7.73	BTV
SH6	1	6.73	1	7.73	HPN
L10	3	6.98	0.75	7.73	BDL
BEC	1	6.75	1	7.75	AUG
SF3	1	6.75	1	7.75	BTV
CNJ	1	7	0.75	7.75	BDL
D9S	2	6.61	1.17	7.78	BUF
72S	2	6.7	1.08	7.78	YUL

BEC	1	6.92	1	7.92	RKD
BEC	1	7.09	0.83	7.92	ALB
BEC	1	7.17	0.75	7.92	ACK
SF3	1	6.48	1.5	7.98	PQI
SH6	1	7	1	8	ISP
SF3	1	7.33	0.67	8	PWM
72S	2	7.07	1	8.07	LGA
BE9	1	7.08	1	8.08	UCA
L10	3	7.08	1	8.08	JFK
733	2	6.93	1.17	8.1	PHL
D9S	2	7.13	1	8.13	ROC
M80	2	7.15	1	8.15	EWR
ATR	1	7.17	1	8.17	HPN
BE1	1	7.34	0.83	8.17	ALB
72S	2	7.03	1.17	8.2	DCA
DH8	1	7	1.25	8.25	ABE
BE1	1	7.25	1	8.25	SYR
F28	2	7.25	1	8.25	SYR
DH8	1	7.08	1.25	8.33	ABE
DO8	1	7.66	0.67	8.33	PWM
73S	2	7.16	1.17	8.33	IAD
73S	2	7.16	1.17	8.33	BWI
HS7	2	7.16	1.17	8.33	YOW
73S	2	5.92	2.5	8.42	ORF
M80	2	7.28	1.17	8.45	DCA
CNA	1	7.83	0.67	8.5	LCI
M80	2	7.41	1.17	8.58	DCA
72S	2	7.65	1	8.65	LGA
SF3	1	7.34	1.33	8.67	BGM
BE1	1	7.8	0.88	8.68	BDR
733	2	7.2	1.5	8.7	CLE
733	2	7.7	1	8.7	EWR
BE1	1	7.42	1.33	8.75	BGM
SH6	1	7.75	1	8.75	ISP
F28	2	7.36	1.42	8.78	RIC
733	2	7.4	1.42	8.82	PIT
72S	2	7	1.83	8.83	CVG
ATR	1	7.92	1	8.92	JFK
BE9	1	7.92	1	8.92	SYR
SF3	1	8.09	0.83	8.92	ALB
D9S	2	7.42	1.5	8.92	YYZ
BE1	1	8.31	0.67	8.98	PWM
SF3	1	8	1	9	BGR
DO8	1	8.17	0.83	9	ALB
757	2	6.67	2.33	9	ATL
757	2	7.25	1.75	9	DTW
72S	2	8.13	1	9.13	LGA
BE1	1	8.51	0.67	9.18	PWM
733	2	8.2	1	9.2	EWR
D9S	2	7.75	1.5	9.25	YYZ
SH6	1	8.42	1	9.42	HPN
DO8	1	8.75	0.67	9.42	LEB
BE1	1	8.95	0.5	9.45	MHT
BE1	1	8.83	0.67	9.5	LEB

733	2	8.46	1.17	9.63	BWI
72S	2	8.65	1	9.65	LGA
ATR	1	8.67	1	9.67	JFK
BE1	1	8.92	0.75	9.67	HVN
PAG	1	9.17	0.5	9.67	PVC
M80	2	8.7	1	9.7	EWR
BE1	1	8.72	1	9.72	HPN
BE1	1	8.72	1	9.72	HPN
DH8	1	9.17	0.58	9.75	YSJ
D9S	2	8.61	1.17	9.78	DCA
767	3	6.47	3.33	9.8	ORD
D9S	2	7.32	2.5	9.82	YHZ
D08	1	9	0.83	9.83	ALB
BEC	1	9.16	0.67	9.83	PWM
BE1	1	8.97	0.88	9.85	BDR
BE1	1	9.12	0.83	9.95	ALB
D9S	2	8.47	1.5	9.97	CLE
DH8	1	8.92	1.08	10	YUL
72S	2	8.03	2	10.03	CLT
73S	2	9.07	1	10.07	SYR
72S	2	8.91	1.17	10.08	PHL
BE1	1	9.63	0.5	10.13	MVY
72S	2	9.13	1	10.13	LGA
BE1	1	9.15	1	10.15	BGR
D9S	2	6.84	3.33	10.17	MDW
D10	3	6.87	3.33	10.2	ORD
733	2	6.9	3.33	10.23	DAY
D08	1	9.75	0.5	10.25	MHT
73S	2	8.25	2	10.25	CLT
D08	1	9.33	1	10.33	ISP
BE1	1	9.5	0.83	10.33	ALB
SH6	1	9.66	0.67	10.33	PWM
757	2	8.58	1.75	10.33	DTW
BEC	1	9.42	1	10.42	BHB
BEC	1	9.92	0.5	10.42	MVY
BEC	1	9.5	1	10.5	AUG
SF3	1	9.5	1	10.5	ROC
D9S	2	9.11	1.42	10.53	PIT
SH6	1	9.58	1	10.58	HPN
SH6	1	9.91	0.67	10.58	HYA
D9S	2	9.41	1.17	10.58	DCA
72S	2	9.6	1	10.6	JFK
SF3	1	9.65	1	10.65	BTV
72S	2	9.65	1	10.65	LGA
D08	1	9.84	0.83	10.67	ALB
D9S	2	9.7	1	10.7	EWR
D9S	2	9.56	1.17	10.73	PHL
SF3	1	10	0.75	10.75	ACK
CNA	1	10.08	0.67	10.75	PWM
D08	1	10.08	0.67	10.75	LEB
DC9	2	8.75	2	10.75	MKE
SF3	1	10.08	0.75	10.83	BDL
D9S	2	8.83	2	10.83	MKE
72S	2	10.21	0.67	10.88	PWM

SH6	1	9.92	1	10.92	ISP
DH7	1	9.98	1	10.98	JFK
ATR	1	10	1	11	HPN
DC9	2	9	2	11	MKE
BE1	1	10.4	0.67	11.07	LEB
73S	2	9.07	2	11.07	CLT
72S	2	10.07	1	11.07	LGA
DO8	1	10.41	0.67	11.08	PWM
72S	2	10.12	1	11.12	BGR
72S	2	9.98	1.17	11.15	IAD
BEC	1	10.5	0.67	11.17	HYA
757	2	10.09	1.08	11.17	YUL
SF3	1	9.7	1.5	11.2	PQI
767	3	8.75	2.5	11.25	ORL
72S	2	10.13	1.17	11.3	BWI
CNJ	1	10.33	1	11.33	JFK
SH6	1	10.66	0.67	11.33	HYA
D9S	2	10.21	1.17	11.38	MDT
BE1	1	10.92	0.5	11.42	MHT
M80	2	10.26	1.17	11.43	DCA
DO8	1	10.5	1	11.5	ISP
767	3	8.22	3.33	11.55	DFW
D9S	2	9.07	2.5	11.57	YHZ
SH6	1	10.58	1	11.58	ISP
BEC	1	10.91	0.67	11.58	PWM
72S	2	10.58	1	11.58	LGA
D9S	2	10.63	1	11.63	EWR
BE1	1	10.67	1	11.67	BHB
PAG	1	11.17	0.5	11.67	PVC
D10	3	8.35	3.33	11.68	ORD
M80	2	9.08	2.67	11.75	STL
737	3	10.28	1.5	11.78	CLE
D9S	2	10.7	1.17	11.87	PHL
M80	2	10.13	1.75	11.88	RDU
72S	2	10.93	1	11.93	LGA
72S	2	10.98	1	11.98	LGA
M80	2	9.69	2.33	12.02	BNA
D9S	2	9.75	2.33	12.08	ATL
BE1	1	11.48	0.67	12.15	PWM
72S	2	10.4	1.83	12.23	CVG
DO8	1	11.42	0.83	12.25	ALB
72S	2	8.5	3.75	12.25	MCI
L10	3	9.6	2.67	12.27	STL
D9S	2	11.3	1	12.3	ROC
757	2	9.99	2.33	12.32	ATL
BE1	1	11.5	0.83	12.33	ALB
BEC	1	11.42	1	12.42	RKD
BE1	1	11.42	1	12.42	JFK
BE1	1	11.45	1	12.45	BTV
SF3	1	11	1.5	12.5	PQI
SF3	1	11.17	1.33	12.5	AIV
BE1	1	11.57	1	12.57	BGR
72S	2	11.58	1	12.58	LGA
DH8	1	11.35	1.25	12.6	YQI

733	2	11.63	1	12.63	EWR
BE9	1	11.67	1	12.67	UCA
BE1	1	12	0.67	12.67	PWM
CNA	1	12	0.67	12.67	LCI
BEC	1	12	0.75	12.75	ACK
73S	2	11.76	1.17	12.93	DCA
72S	2	11.98	1	12.98	LGA
SF3	1	12	1	13	BGR
SF3	1	12	1	13	ISP
D9S	2	11.58	1.5	13.08	YYZ
M80	2	12.41	0.67	13.08	PWM
BE1	1	11.84	1.33	13.17	BGM
BE9	1	12.17	1	13.17	SYR
BEC	1	12.5	0.67	13.17	HYA
D1M	3	3.87	9.3	13.17	BRU
ATR	1	12.25	1	13.25	JFK
L10	3	6	7.25	13.25	LHR
BE1	1	12.73	0.67	13.4	LEB
SH6	1	12.9	0.5	13.4	MVY
D9S	2	11.98	1.42	13.4	PIT
DH7	1	12.42	1	13.42	JFK
BE1	1	11.98	1.5	13.48	PQI
D10	3	10.15	3.33	13.48	ORD
BE1	1	12.5	1	13.5	BTV
D08	1	12.5	1	13.5	ISP
734	2	12.38	1.17	13.55	BWI
BE1	1	12.57	1	13.57	BGR
SF3	1	13.07	0.5	13.57	MHT
72S	2	11.08	2.5	13.58	ORL
72S	2	12.58	1	13.58	LGA
72S	2	12.52	1.08	13.6	YUL
M80	2	12.63	1	13.63	EWR
PAG	1	13.17	0.5	13.67	PVC
733	2	12.53	1.17	13.7	PHL
D9S	2	12.23	1.5	13.73	YYZ
D9S	2	10.42	3.33	13.75	MDW
757	2	12.05	1.75	13.8	DTW
D08	1	13	0.83	13.83	ALB
D08	1	13.16	0.67	13.83	PWM
D08	1	13.25	0.67	13.92	LEB
72S	2	12.8	1.17	13.97	PHL
733	2	12.98	1	13.98	SYR
72S	2	12.98	1	13.98	LGA
SH6	1	13.33	0.67	14	HYA
D08	1	13.5	0.5	14	MHT
733	2	12.83	1.17	14	IAD
SF3	1	13.33	0.75	14.08	ACK
SH6	1	13.33	0.75	14.08	BDL
AB3	3	10.17	4	14.17	SJU
767	3	11.72	2.5	14.22	ORL
SH6	1	13.25	1	14.25	HPN
BE1	1	13.28	1	14.28	BTV
757	2	11.58	2.75	14.33	MEM
747	3	7.08	7.25	14.33	LHR

BE1	1	13.6	0.75	14.35	ACK
733	2	13.18	1.17	14.35	PHL
BE1	1	13.59	0.83	14.42	ALB
BEC	1	13.5	1	14.5	BHB
DH7	1	13.5	1	14.5	JFK
SH6	1	13.5	1	14.5	ISP
733	2	12.53	2	14.53	CLT
D9S	2	13.4	1.17	14.57	DCA
BE1	1	13.58	1	14.58	BTV
M80	2	11.25	3.33	14.58	DFW
72S	2	13.58	1	14.58	LGA
M80	2	13.62		14.62	EWR
BE1	1	14	0.67	14.67	PWM
BEC	1	14	0.67	14.67	HYA
BEC	1	14	0.67	14.67	PWM
D08	1	14	0.67	14.67	PWM
72S	2	12	2.67	14.67	STL
SF3	1	13.92	0.83	14.75	ALB
D10	3	11.45	3.33	14.78	ORD
757	2	11.83	3	14.83	TPA
M80	2	13.48	1.42	14.9	PIT
72S	2	12.18	2.75	14.93	PBI
757	2	14.3	0.67	14.97	PWM
72S	2	13.98	1	14.98	LGA
D08	1	14	1	15	ISP
BE1	1	14.25	0.75	15	BDL
72S	2	12.05	3	15.05	MIA
72S	2	14.08	1	15.08	BGR
SH3	1	14.25	1	15.25	HPN
BE9	1	14.5	0.75	15.25	ACK
CNA	1	14.58	0.67	15.25	PWM
D08	1	14.66	0.67	15.33	LEB
D08	1	14.83	0.5	15.33	MHT
D9S	2	14.16	1.17	15.33	PHL
BE1	1	14.9	0.5	15.4	MVY
D9S	2	12.9	2.5	15.4	YHZ
BEC	1	14.92	0.5	15.42	MVY
757	2	12.59	2.83	15.42	FLL
D9S	2	14.31	1.17	15.48	DCA
72S	2	14.33	1.17	15.5	BDA
D9S	2	14.33	1.17	15.5	DCA
D9S	2	14.03	1.5	15.53	CLE
72S	2	14.58	1	15.58	LGA
L10	3	7.98	7.6	15.58	PAR
PAG	1	15.17	0.5	15.67	PVC
BEC	1	14.75	1	15.75	AUG
D9S	2	14.75	1	15.75	EWR
767	3	13.42	2.33	15.75	ATL
734	2	14.63	1.17	15.8	BWI
BE1	1	15.16	0.67	15.83	LEB
M80	2	14.41	1.42	15.83	PIT
BE1	1	15.09	0.83	15.92	ALB
DH8	1	15.34	0.58	15.92	YSJ
D9S	2	14.97	1	15.97	ROC

72S	2	14.98	1	15.98	LGA
CNA	1	15.33	0.67	16	PWM
DO8	1	15.33	0.67	16	PWM
757	2	11.67	4.33	16	SLC
D9S	2	15.05	1	16.05	BTV
72S	2	13.07	3	16.07	TPA
SH6	1	15.08	1	16.08	ISP
M80	2	12.27	3.83	16.1	IAH
M80	2	14.35	1.75	16.1	RDU
D9S	2	14.95	1.17	16.12	BUF
SF3	1	15.17	1	16.17	BGR
SH6	1	15.17	1	16.17	HPN
BE1	1	15.29	0.88	16.17	BDR
SF3	1	15.67	0.5	16.17	MVY
D10	3	8.17	8	16.17	FRA
ATR	1	15.23	1	16.23	JFK
SF3	1	15.23	1	16.23	BTV
BEC	1	15.5	0.75	16.25	ACK
757	2	14.5	1.75	16.25	DTW
D9S	2	15.25	1	16.25	EWR
DO8	1	15.83	0.5	16.33	MHT
D9S	2	14.83	1.5	16.33	YYZ
D9S	2	13.88	2.5	16.38	ORL
72S	2	14.38	2	16.38	CLT
BE1	1	15.73	0.67	16.4	PWM
733	2	15.3	1.17	16.47	BWI
DO8	1	15.5	1	16.5	ISP
BE1	1	15.57	1	16.57	BGR
72S	2	13.57	3	16.57	MIA
767	3	11.35	5.25	16.6	LAX
72S	2	15.63	1	16.63	LGA
D10	3	12.88	3.75	16.63	DEN
D8S	3	12.15	4.5	16.65	SFO
DO8	1	15.84	0.83	16.67	ALB
BEC	1	16	0.67	16.67	PWM
D9S	2	13.34	3.33	16.67	MDW
72S	2	14.85	1.83	16.68	CVG
BEC	1	15.75	1	16.75	RKD
SF3	1	15.75	1	16.75	BTV
BE1	1	15.92	0.83	16.75	ALB
767	3	11.52	5.25	16.77	LAX
M80	2	15.8	1	16.8	EWR
72S	2	15.77	1.08	16.85	YUL
72S	2	14.54	2.33	16.87	ATL
D10	3	13.54	3.33	16.87	ORD
L10	3	12.42	4.5	16.92	SFO
BEC	1	16	1	17	AUG
DO8	1	16.17	0.83	17	ALB
SH6	1	16.08	1	17.08	HPN
M80	2	13.33	3.75	17.08	DEN
M80	2	14.75	2.33	17.08	BNA
SH6	1	16.65	0.5	17.15	MVY
72S	2	16.15	1	17.15	LGA
BE1	1	16.5	0.67	17.17	HYA

73S	2	16	1.17	17.17	PHL
L10	3	14.5	2.67	17.17	STL
BE1	1	16.53	0.67	17.2	HYA
72S	2	13.89	3.33	17.22	ORD
D8S	3	13.89	3.33	17.22	ORD
DH7	1	16.25	1	17.25	JFK
SH6	1	16.25	1	17.25	HPN
SH6	1	16.52	0.75	17.27	ACK
733	2	16.3	1	17.3	EWR
BE1	1	16.33	1	17.33	BTV
SH6	1	16.33	1	17.33	ISP
BE1	1	16.49	0.88	17.37	BDR
F28	2	14.54	2.83	17.37	GSO
72S	2	14.63	2.75	17.38	MEM
CNA	1	16.75	0.67	17.42	LCI
D9S	2	15.92	1.5	17.42	YYZ
BE1	1	16.75	0.75	17.5	HVN
D08	1	17	0.5	17.5	MHT
72S	2	16.77	0.75	17.52	BDL
D9S	2	16.03	1.5	17.53	CLE
BE1	1	16.57	1	17.57	BGR
747	3	16.65	1	17.65	JFK
PAG	1	17.17	0.5	17.67	LGA
SF3	1	16.42	1.33	17.75	BGM
D08	1	17.08	0.67	17.75	DFW
M80	2	16.8	1	17.8	EWR
SF3	1	17	0.83	17.83	ALB
733	2	16.68	1.17	17.85	IAD
BE9	1	16.92	1	17.92	UCA
SF3	1	17.25	0.67	17.92	PWM
757	2	14.59	3.33	17.92	DFW
72S	2	16.75	1.17	17.92	PHL
DH8	1	16.92	1.08	18	YUL
D08	1	17	1	18	ISP
72S	2	16.27	1.75	18.02	DTW
L10	3	16.85	1.17	18.02	BDA
BE1	1	17.53	0.5	18.03	MHT
72S	2	17.03	1	18.03	SYR
BEC	1	17.08	1	18.08	BHB
SF3	1	17.41	0.67	18.08	HYA
72S	2	17.12	1	18.12	LGA
SF3	1	17.48	0.67	18.15	PWM
D9S	2	16.7	1.5	18.2	CLE
733	2	16.78	1.42	18.2	PIT
SH3	1	17.25	1	18.25	HPN
DC9	2	16.25	2	18.25	MKE
D9S	2	17.11	1.17	18.28	PHL
ATR	1	17.33	1	18.33	JFK
D08	1	17.5	0.83	18.33	ALB
BE1	1	17.09	1.33	18.42	BGM
D08	1	18	0.5	18.5	MHT
D10	3	16.78	1.75	18.53	DTW
733	2	16.57	2	18.57	CLT
BE1	1	17.75	0.83	18.58	ALB

HS7	2	17.41	1.17	18.58	YOW
D9S	2	17.46	1.17	18.63	MDT
757	2	17.63	1	18.63	LGA
BE9	1	17.67	1	18.67	SYR
DO8	1	18	0.67	18.67	PWM
73S	2	16.59	2.08	18.67	DAY
72S	2	17.55	1.17	18.72	DCA
72S	2	17.73	1	18.73	LGA
72S	2	15.42	3.33	18.75	ORD
D9S	2	17.25	1.5	18.75	YYZ
DH7	1	17.83	1	18.83	JFK
SF3	1	17.83	1	18.83	BGR
BE1	1	17.95	0.88	18.83	BDR
BEC	1	18	0.83	18.83	ALB
M80	2	17.87	1	18.87	EWR
SH6	1	17.92	1	18.92	ISP
72S	2	18.17	0.75	18.92	BDL
D9S	2	17.76	1.17	18.93	PHL
BEC	1	18.25	0.75	19	ACK
CNA	1	18.33	0.67	19	LCI
DO8	1	18.33	0.67	19	LEB
73S	2	17.83	1.17	19	BWI
BE1	1	18.55	0.5	19.05	MVY
DH8	1	17.83	1.25	19.08	YQI
SH6	1	18.08	1	19.08	HPN
72S	2	17.96	1.17	19.13	DCA
D9S	2	18.15	1	19.15	LGA
72S	2	15.84	3.33	19.17	ORD
D10	3	13.92	5.25	19.17	LAX
F28	2	17.86	1.42	19.28	RIC
DO8	1	18.33	1	19.33	ISP
DO8	1	18.5	0.83	19.33	ALB
SH6	1	18.66	0.67	19.33	DTW
72S	2	16.54	2.83	19.37	FLL
767	3	16.1	3.33	19.43	DFW
M80	2	18.28	1.17	19.45	DCA
M80	2	18.45	1	19.45	EWR
D9S	2	17.47	2	19.47	MKE
73S	2	18.31	1.17	19.48	IAD
BEC	1	18.5	1	19.5	RKD
72S	2	16.17	3.33	19.5	ORD
L10	3	17.17	2.33	19.5	ATL
BEC	1	18.58	1	19.58	BHB
BE1	1	18.75	0.83	19.58	ALB
D9S	2	18.6	1	19.6	ROC
BE1	1	18.9	0.75	19.65	BDL
SF3	1	18.67	1	19.67	BTV
SH6	1	18.92	0.75	19.67	ACK
PAG	1	19.17	0.5	19.67	PVC
72S	2	16.67	3	19.67	MIA
72S	2	18.67	1	19.67	BGR
BE1	1	19.2	0.5	19.7	MHT
733	2	15.95	3.75	19.7	DEN
72S	2	18.7	1	19.7	LGA

CNJ	1	19.08	0.75	19.83	BDL
767	3	18.77	1.08	19.85	YUL
72S	2	18.9	1	19.9	JFK
SF3	1	19.09	0.83	19.92	ALB
733	2	18.95	1	19.95	EWR
733	2	16.15	3.83	19.98	IAH
733	2	18.98	1	19.98	SYR
72S	2	19.31	0.67	19.98	PWM
SH6	1	19.05	1	20.05	HPN
SH6	1	19.08	1	20.08	ISP
SF3	1	19.33	0.75	20.08	ACK
72S	2	18.91	1.17	20.08	DCA
72S	2	19.15	1	20.15	LGA
ATR	1	19.25	1	20.25	JFK
BEC	1	19.75	0.5	20.25	MVY
D9S	2	19.1	1.17	20.27	DCA
73S	2	19.11	1.17	20.28	BUF
F28	2	17.8	2.5	20.3	ORF
D10	3	17	3.33	20.33	ORD
BE1	1	19.5	1	20.5	BGR
SH6	1	19.58	1	20.58	HPN
D9S	2	19.08	1.5	20.58	YYZ
D9S	2	19.46	1.17	20.63	PHL
D8S	3	15.38	5.25	20.63	LAX
DH7	1	19.67	1	20.67	JFK
D9S	2	19.17	1.5	20.67	YYZ
72S	2	19.67	1	20.67	LGA
D10	3	17.4	3.33	20.73	ORD
D9S	2	18.42	2.33	20.75	ATL
D9S	2	19.77	1	20.77	EWR
D08	1	19.83	1	20.83	ISP
767	3	18.43	2.5	20.93	ORL
D9S	2	19.61	1.42	21.03	PIT
D8S	3	17.3	3.75	21.05	DEN
72S	2	20.08	1	21.08	LGA
D9S	2	19.68	1.5	21.18	CLE
72S	2	20.25	1	21.25	JFK
72S	2	19	2.33	21.33	ATL
M80	2	18.86	2.67	21.53	STL
AB3	3	17.55	4	21.55	SJU
733	2	19.57	2	21.57	CLT
72S	2	18.25	3.33	21.58	DFW
72S	2	20.63	1	21.63	LGA
767	3	16.48	5.25	21.73	LAX
ATR	1	20.75	1	21.75	JFK
73S	3	20.58	1.17	21.75	BWI
72S	2	18.02	3.75	21.77	MCI
M80	2	20.8	1	21.8	EWR
AB3	3	18.82	3	21.82	MIA
D10	3	18.54	3.33	21.87	DFW
757	2	18.92	3	21.92	TPA
72S	2	20.95	1	21.95	LGA
ATR	1	21	1	22	JFK
733	2	18.72	3.33	22.05	ORD

D10	3	17.73	4.5	22.23	SFO
D9S	2	18.92	3.33	22.25	MDW
72S	2	19.75	2.5	22.25	ORL
72S	2	21.08	1.17	22.25	DCA
733	2	20.78	1.5	22.28	CLE
734	2	21.13	1.17	22.3	BWI
D9S	2	20.17	2.33	22.5	ATL
M80	2	20.75	1.75	22.5	DTW
72S	2	21.53	1	22.53	LGA
72S	2	19.83	2.75	22.58	PBI
73S	2	20.57	2.08	22.65	DAY
767	3	20.84	1.83	22.67	CVG
M80	2	21.07	1.75	22.82	RDU
D9S	2	21.66	1.17	22.83	DCA
733	2	21.73	1.17	22.9	IAD
757	2	18.59	4.33	22.92	SLC
72S	2	21.92	1	22.92	LGA
M80	2	20.62	2.33	22.95	BNA
757	3	21.95	1	22.95	LGA
SF3	1	21.92	1.33	23.25	AIV
757	2	20.5	2.83	23.33	FLL
757	2	21.05	2.33	23.38	ATL
D10	3	20.1	3.33	23.43	ORD
733	2	22.06	1.42	23.48	PIT
D10	3	21.88	1.75	23.63	DTW
AB3	3	22.73	1	23.73	JFK
733	2	19.97	3.83	23.8	IAH
ATR	1	22.83	1	23.83	JFK
733	2	21.87	2	23.87	CLT
M80	2	22.95	1	23.95	EWR
767	3	20.64	3.33	23.97	ORD

2B. DATA BASE SHOWING AIRCRAFT TYPE, DEPARTURE, AND ARRIVAL PERIOD :

TYPE	DEPART	ARRIVES
1	1	3
1	2	4
1	2	4
1	3	6
1	5	7
1	5	7
1	5	7
1	5	7
1	5	7
2	3	7
1	2	8
1	4	8
1	4	8
1	4	8
1	4	8
1	4	8
1	5	8
1	5	8
1	5	8
1	6	8
2	4	8
2	4	8
2	4	8
2	4	8
2	5	8
3	5	8
1	3	9
1	5	9
1	5	9
1	5	9
1	6	9
1	6	9
1	6	9
2	4	9
2	4	9
1	6	10
1	6	10
1	6	10
1	7	10
1	7	10
2	5	10
2	6	10
2	6	10
2	6	10
2	6	10
3	6	10
1	6	11
1	6	11
1	7	11

1	8	11
2	1	11
2	6	11
2	6	11
2	6	11
2	7	11
2	7	11
1	7	12
1	9	12
1	9	12
2	6	12
2	7	12
2	8	12
2	8	12
1	7	13
1	9	13
1	9	13
1	9	13
1	9	13
1	10	13
1	11	13
2	6	13
2	7	13
2	7	13
2	7	13
1	10	14
1	10	14
1	12	14
2	4	14
2	7	14
2	10	14
2	10	14
1	11	15
1	13	15
1	13	15
2	9	15
1	12	16
1	12	16
1	12	16
1	13	16
1	13	16
1	14	16
2	11	16
2	12	16
2	12	16
1	13	17
1	14	17
1	14	17
1	14	17
1	14	17
2	7	17
2	11	17
2	12	17
3	3	17

1	13	18
1	14	18
1	16	18
2	5	18
2	5	18
2	10	18
2	13	18
2	14	18
2	14	18
3	5	18
1	15	19
1	15	19
1	16	19
1	16	19
1	17	19
1	17	19
2	11	19
2	12	19
1	16	20
1	16	20
1	16	20
1	16	20
1	17	20
1	17	20
2	14	20
2	15	20
2	16	20
2	16	20
2	16	20
2	16	20
1	17	21
1	17	21
1	18	21
1	18	21
1	18	21
1	18	21
2	13	21
2	13	21
2	18	21
1	16	22
1	18	22
1	19	22
1	19	22
1	20	22
2	14	22
2	14	22
2	17	22
2	18	22
2	18	22
2	18	22
1	19	23
1	20	23
1	21	23
2	18	23

2	18	23
2	19	23
3	13	23
1	20	24
1	20	24
1	20	24
1	21	24
1	22	24
2	14	24
2	20	24
2	20	24
3	10	24
3	11	24
2	14	25
2	18	25
2	20	25
2	21	25
2	21	25
3	19	25
1	23	26
2	16	26
2	17	26
2	19	26
1	23	27
1	23	27
1	23	27
1	23	27
1	24	27
2	12	27
2	17	27
2	23	27
3	16	27
1	22	28
1	22	28
1	23	28
1	24	28
1	24	28
1	26	28
1	26	28
2	24	28
2	24	28
1	26	29
2	25	29
2	25	29
1	25	30
1	26	30
1	26	30
1	26	30
1	28	30
2	24	30
2	27	30
1	25	31
1	27	31
1	27	31

1	28	31
1	29	31
2	25	31
3	2	31
3	18	31
1	28	32
1	28	32
1	28	32
1	30	32
1	30	32
2	22	32
2	26	32
2	27	32
2	28	32
2	28	32
2	28	32
2	28	32
1	30	33
1	30	33
1	31	33
2	19	33
2	26	33
2	29	33
2	29	33
2	29	33
1	31	34
1	31	34
1	31	34
1	32	34
2	29	34
3	18	34
3	24	34
1	31	35
1	31	35
1	32	35
1	32	35
2	24	35
2	30	35
3	6	35
1	32	36
1	32	36
1	32	36
1	32	36
1	34	36
1	34	36
1	34	36
1	34	36
2	23	36
2	26	36
2	28	36
2	31	36
2	32	36
2	32	36
1	33	37

2	25	37
2	26	37
2	31	37
2	33	37
2	35	37
3	23	37
1	34	38
1	35	38
2	26	38
2	34	38
1	35	39
1	36	39
1	36	39
1	36	39
1	37	39
1	37	39
1	37	39
2	28	39
2	29	39
2	34	39
2	35	39
1	38	40
2	34	40
2	35	40
2	35	40
2	36	40
3	9	40
1	37	41
1	38	41
1	38	41
1	39	41
2	35	41
2	36	41
2	37	41
2	37	41
2	37	41
3	31	41
1	38	42
1	38	42
1	38	42
1	38	42
1	38	42
1	39	42
1	39	42
1	39	42
1	40	42
2	24	42
2	27	42
2	30	42
2	35	42
2	37	42
2	38	42
3	10	42
1	40	43

1	40	43
1	41	43
2	33	43
2	35	43
2	36	43
2	37	43
2	39	43
2	39	43
1	40	44
1	40	44
1	41	44
1	42	44
2	31	44
2	32	44
2	37	44
2	40	44
3	23	44
3	26	44
3	29	44
1	41	45
1	41	45
1	41	45
2	36	45
2	41	45
2	41	45
3	24	45
3	27	45
3	32	45
1	42	46
1	42	46
1	42	46
1	44	46
1	44	46
1	44	46
2	31	46
2	33	46
2	37	46
2	42	46
2	42	46
3	33	46
3	36	46
1	43	47
1	43	47
1	43	47
1	43	47
1	43	47
1	44	47
1	45	47
2	36	47
2	36	47
2	41	47
2	43	47
1	44	48
1	45	48

1	46	48
1	46	48
2	42	48
2	44	48
2	45	48
3	44	48
1	43	49
1	45	49
1	46	49
1	46	49
1	47	49
2	36	49
2	44	49
2	45	49
2	45	49
3	35	49
1	45	50
1	46	50
1	46	50
1	47	50
1	47	50
1	48	50
2	43	50
2	44	50
2	45	50
2	46	50
2	46	50
3	45	50
1	46	51
1	47	51
1	47	51
1	48	51
2	43	51
2	46	51
1	48	52
1	49	52
1	50	52
1	50	52
2	44	52
2	44	52
2	47	52
2	47	52
2	48	52
2	48	52
2	48	52
3	45	52
1	49	53
1	49	53
1	49	53
1	49	53
1	50	53
2	39	53
2	47	53
2	49	53

2	49	53
2	50	53
1	49	54
1	50	54
1	51	54
1	51	54
1	51	54
1	52	54
2	41	54
2	49	54
2	49	54
2	50	54
3	33	54
1	51	55
1	52	55
1	52	55
2	44	55
2	47	55
2	48	55
2	49	55
2	51	55
2	51	55
2	51	55
3	42	55
1	52	56
1	52	56
1	52	56
1	53	56
1	53	56
1	53	56
1	54	56
1	54	56
2	41	56
2	42	56
2	44	56
2	52	56
2	52	56
2	52	56
3	46	56
1	54	57
1	54	57
2	42	57
2	53	57
2	53	57
2	53	57
2	55	57
3	53	57
1	54	58
1	54	58
1	55	58
2	53	58
2	54	58
1	55	59
1	57	59

2	49	59
2	54	59
2	54	59
3	46	59
1	56	60
1	56	60
1	56	60
2	54	60
2	54	60
2	55	60
2	56	60
3	39	60
3	47	60
1	57	61
2	51	61
2	57	61
3	51	61
2	56	62
2	56	62
2	58	62
3	47	62
2	54	63
2	59	63
2	51	64
2	53	64
2	56	64
2	60	64
3	43	64
3	48	64
1	61	65
2	50	65
2	53	65
2	61	65
2	61	65
3	52	65
3	53	65
3	60	65
1	62	66
2	52	66
3	48	66
2	53	67
2	57	67
2	61	67
2	62	67
2	62	67
2	57	68
2	58	68
2	60	68
2	61	68
2	64	68
3	61	68
2	52	69
2	60	69
2	62	69

2	64	69
2	64	69
2	65	69
3	65	69
1	65	71
2	60	71
2	62	71
2	66	71
3	58	71
3	65	72
3	68	72
1	69	73
2	57	73
2	65	73
2	69	73
3	60	73

2C. STATIC SCHEDULE FOR ONE AND THREE AIRCRAFT CLASSES

The static schedule for 1 aircraft class, period 1 through T+1=74 is :

N=0,0,1,2,0,1,6,16,9,11,10,7,11,7,4,9,9,10,8,12,9,11,7,10,6,4,9,9,3,7, 8,12,8,7,7,14,7,4,11,6,10,16,9,11,9,13,11,8,10,12,6,12,10,11,11,15,8,5,6,9, 4,4,2,6,8,3,5,6,7,0,5,2,5,0.

The static schedulefor 3 aircraft classes, period 1 through T+1:

N_1=0,0,1,2,0,1,5,10,7,5,4,3,7,3,3,6,5,3,6,6,6,5,3,5,0,1,5,7,1,5,5,5,3,4, 4,8,1,2,7,1,4,9,3,4,3,6,7,4,5,6,4,4,5,6,3,8,2,3,2,3,1,0,0,0,1,1,0,0,0,0,1,0,1,0.

N_2=0,0,0,0,0,0,1,5,2,5,6,4,4,4,1,3,3,6,2,6,3,6,3,3,5,3,3,2,2,2,1,7,5,1,2,6,5,2,4 ,4,5,6,6,4,3,5,4,3,4,5,2,7,5,4,7,6,5,2,3,4,2,3,2,4,4,1,5,5,6,0,3,0,3,0.

N_3=0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,1,0,0,0,0,1,2,1,0,1,0,0,0,2,0,0,2,1, 0,1,0,0,1,1,1,0,3,3,2,0,1,1,1,0,1,0,1,1,1,1,0,1,2,1,1,0,2,3,1,0,1,1,0,1,2,1,0.

APPENDIX 3

CAPACITY CASES

Capacity Case 1:

		Scenario			
Until (hour)	capacity (landings/15 min)	1	2	3	4
12.0	15				
16.0	10	.5	.3	.3	.34
18.0	7				
20.0	10				
24.0	15				
12.0	15				
16.0	10	.3	.5	.2	.33
24.0	15				
24.0	15	.2	.2	.5	.33

Capacity Case 2:

		Scenario			
Until (hour)	capacity (landings/15 min)	1	2	3	4
24.0	10	.6	.1	.1	.34
18.0	10	.3	.6	.3	.33
24.0	15				
14.0	10	.1	.3	.6	.33
24.0	15				

Capacity Case 3:

		Scenario			
Until (hour)	capacity (landings/15 min)	1	2	3	4
10.0	7				
15.0	15	.6	.1	.33	.3
18.0	10				
24.0	15				

11.0		7				
15.0	10		.3	.3	.33	.6
24.0	15					
12.0		7				
16.0	10		.1	.6	.34	.1
24.0	15					

Capacity Case 4:

			Scenario	
Until (hour)	capacity (landings/15 min)		1	
10.0	15			
18.0	10		.3	
24.0	7			
12.0	15			
24.0		10		.3
12.0	15			
16.0	10			
18.0	7		.4	
20.0	10			
24.0	15			

Capacity Case 5:

			Scenario	
Until (hour)	capacity (landings/15 min)		1	
9.0	15			
12.0	10			
16.0	15		.3	
24.0	10			
16.0	15			
24.0		10		.6
12.0	15			
16.0	10			
18.0	15		.1	
20.0	10			
24.0	15			

Capacity Case 6:

		Scenario
Until (hour)	capacity (landings/15 min)	1
9.0	15	
12.0	10	
16.0	7	.6
24.0	10	
16.0	15	
24.0	10	.3
11.0	15	
15.0	10	
18.0	15	.1
20.0	10	
24.0	7	

Capacity Case 7:

Until (hour)	capacity (landings/15 min)	1
12.0	10	
14.0	15	
18.0	10	.3
24.0	7	
10.0	10	
13.0	15	
15.0	10	.2
18.0	7	
24.0	10	
15.0	10	
18.0	15	
20.0	10	.5
24.0	7	

Capacity Case 8:

Until (hour)	capacity (landings/15 min)	1
12.0	7	
14.0	10	.3
24.0	15	
11.0	7	

13.0	10	
18.0	15	.2
24.0	10	
13.0	7	
16.0	10	
20.0	15	.5
24.0	10	

Capacity Case 9:

		Scenario
Until (hour)	capacity (landings/15 min)	1
10.0	10	
15.0	7	
18.0	10	.3
24.0	15	
7.0	10	
12.0	15	
14.0	10	
18.0	7	.2
20.0	10	
24.0	15	
12.0	10	
16.0	7	.5
24.0	10	

Capacity Case 10:

Until (hour)	capacity (landings/15 min)	1
9.0	7	
12.0	10	
14.0	7	.1
24.0	10	
10.0	7	
i5.0	10	
18.0	15	.3
24.0	10	
12.0	7	
18.0	10	.6
24.0	15	

APPENDIX 4

SPREADSHEET FOR SOLUTION STATISTICS

HEADING:	DESCRIPTION:
CAP.CASE	Capacity case number
PROBS.	Probability scenario number for given capacity case.
AIRCOST	Marginal cost of air delays ($/period)
SOLUTION	Name of algorithm used to solve problem defined by CAP.CASE, PROBS., AIRCOST above.
EXPCOST	Expected total delay costs (ground +air) ($)
EGNDCOST	Expected ground delay cost ($)
EAIRCOST	Expected air delay cost ($)
EGD1	Expected ground delay for type 1 aircraft (aircraft-periods)
EGD2	Expected ground delay for type 2 aircraft (aircraft-periods)
EGD3	Expected ground delay for type 2 aircraft (aircraft-periods)
EGDT	Expected ground delay for all aircraft (aircraft-periods)
EAD	Expected air delay for all aircraft (aircraft-periods)
%VSDETERM	% of total expected cost for DETERM. solution

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
1	1	1200	DETERM.	491430	491430	0				491	0	100
1	1	1200	PASSIVE	315720	0	315720				0	263	64
1	1	1200	STATIC	315320	2000	313320				2	261	64
1	1	1200	DYNAMICH	263890	263890	0				264	0	54
1	1	1200	DYNAMIC	263890	263890	0				264	0	54
1	1	1200	DYNAMIC3C	238559	237359	1200	191	74	1	266	1	49
1	1	1600	DETERM.	491430	491430	0				491	0	100
1	1	1600	PASSIVE	420960	0	420960				0	263	86
1	1	1600	STATIC	402400	65600	336800				66	211	82
1	1	1600	DYNAMICH	263890	263890	0				264	0	54
1	1	1600	DYNAMIC	263890	263890	0				264	0	54
1	1	1600	DYNAMIC3C	238659	238659	0	191	75	1	267	0	49
1	1	2000	DETERM.	491430	491430	0				491	0	100
1	1	2000	PASSIVE	526200	0	526200				0	263	107
1	1	2000	STATIC	487000	107000	380000				107	190	99
1	1	2000	DYNAMICH	263890	263890	0				264	0	54
1	1	2000	DYNAMIC	263890	263890	0				264	0	54
1	1	2000	DYNAMIC3C	238659	238659	0	191	75	1	267	0	49
1	1	3000	DETERM.	491430	491430	0				491	0	100
1	1	3000	PASSIVE	789300	0	789300				0	263	161
1	1	3000	STATIC	491430	491430	0				491	0	100
1	1	3000	DYNAMICH	263890	263890	0				264	0	54
1	1	3000	DYNAMIC	263890	263890	0				264	0	54
1	1	3000	DYNAMIC3C	238659	238659	0	191	75	1	267	0	49
1	1	3000	DYNAMIC3C2	182246	182246	0	193	76	0	269	0	37
1	2	1200	DETERM.	216280	64000	152280				64	127	100
1	2	1200	PASSIVE	214200	0	214200				0	179	99
1	2	1200	STATIC	213800	2000	211800				2	177	99
1	2	1200	DYNAMICH	179838	179838	0				180	0	83
1	2	1200	DYNAMIC	179838	179838	0				180	0	83
1	2	1200	DYNAMIC3C	159699	158979	720	135	45	0	180	0.6	74
1	2	1600	DETERM.	267040	64000	203040				64	127	100
1	2	1600	PASSIVE	285600	0	285600				0	179	107

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
1	2	1600	STATIC	267040	64000	203040				64	127	100
1	2	1600	DYNAMICH	179838	179838	0				180	0	67
1	2	1600	DYNAMIC	179838	179838	0				180	0	67
1	2	1600	DYNAMIC3C	159759	159759	0	135	45	1	181	0	60
1	2	2000	DETERM.	317800	64000	253800				64	127	100
1	2	2000	PASSIVE	357000	0	357000				0	179	112
1	2	2000	STATIC	317800	64000	253800				64	127	100
1	2	2000	DYNAMICH	179838	179838	0				180	0	57
1	2	2000	DYNAMIC	179838	179838	0				180	0	57
1	2	2000	DYNAMIC3C	159759	159759	0	135	45	1	181	0	50
1	2	3000	DETERM.	444700	64000	380700				64	127	100
1	2	3000	PASSIVE	535500	0	535500				0	179	120
1	2	3000	STATIC	444700	68500	376200				69	125	100
1	2	3000	DYNAMICH	179838	179838	0				180	0	40
1	2	3000	DYNAMIC	179838	179838	0				180	0	40
1	2	3000	DYNAMIC3C	159759	159759	0	135	45	1	181	0	36
1	2	3000	DYNAMIC3C2	118274	118274	0	136	46	0	182	0	27
1	3	1200	DETERM.	191480	2000	189480				2	158	100
1	3	1200	PASSIVE	191880	0	191880				0	160	100
1	3	1200	STATIC	191480	2000	189480				2	158	100
1	3	1200	DYNAMICH	161238	161238	0				161	0	84
1	3	1200	DYNAMIC	161238	161238	0				161	0	84
1	3	1200	DYNAMIC3C	144771	144051	720	117	44	0	161	0.6	76
1	3	1600	DETERM.	254640	2000	252640				2	158	100
1	3	1600	PASSIVE	255840	0	255840				0	160	100
1	3	1600	STATIC	254640	2000	252640				2	158	100
1	3	1600	DYNAMICH	161238	161238	0				161	0	63
1	3	1600	DYNAMIC	161238	161238	0				161	0	63
1	3	1600	DYNAMIC3C	144830	144830	0	117	44	1	162	0	57
1	3	2000	DETERM.	317800	2000	315800				2	158	100
1	3	2000	PASSIVE	319800	0	319800				0	160	101
1	3	2000	STATIC	317800	2000	315800				2	158	100
1	3	2000	DYNAMICH	161238	161238	0				161	0	51

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
1	3	2000	DYNAMIC	161238	161238	0				161	0	51
1	3	2000	DYNAMIC3C	144830	144830	0	117	44	1	162	0	46
1	3	3000	DETERM.	475700	2000	473700				2	158	100
1	3	3000	PASSIVE	479700	0	479700				0	160	101
1	3	3000	STATIC	444700	50000	394700				50	132	93
1	3	3000	DYNAMICH	161238	161238	0				161	0	34
1	3	3000	DYNAMIC	161238	161238	0				161	0	34
1	3	3000	DYNAMIC3C	144830	144830	0	117	44	1	162	0	30
1	3	3000	DYNAMIC3C2	110228	110228	0	117	46	0	163	0	23
1	4	1200	DETERM.	224432	2000	222432				2	185	100
1	4	1200	PASSIVE	224880	0	224880				0	187	100
1	4	1200	STATIC	224432	2000	222432				2	185	100
1	4	1200	DYNAMICH	183982	183982	0				184	0	82
1	4	1200	DYNAMIC	182092	182092	0				182	0	81
1	4	1200	DYNAMIC3C	169002	168186	816	138	51	0	189	0.7	75
1	4	1600	DETERM.	298576	2000	296576				2	185	100
1	4	1600	PASSIVE	299840	0	299840				0	187	100
1	4	1600	STATIC	294112	64000	230112				64	144	99
1	4	1600	DYNAMICH	183982	183982	0				184	0	82
1	4	1600	DYNAMIC	182092	182092	0				182	0	61
1	4	1600	DYNAMIC3C	169070	169070	0	138	51	1	190	0	57
1	4	2000	DETERM.	372720	2000	370720				2	185	100
1	4	2000	PASSIVE	374800	0	374800				0	187	101
1	4	2000	STATIC	351640	64000	287640				64	144	94
1	4	2000	DYNAMICH	183982	183982	0				184	0	82
1	4	2000	DYNAMIC	182092	182092	0				182	0	49
1	4	2000	DYNAMIC3C	169070	169070	0	138	51	1	190	0	45
1	4	3000	DETERM.	558080	2000	556080				2	185	100
1	4	3000	PASSIVE	562200	0	562200				0	160	101
1	4	3000	STATIC	491290	395410	95880				395	32	88
1	4	3000	DYNAMICH	183982	183982	0				184	0	82
1	4	3000	DYNAMIC	182092	182092	0				182	0	33
1	4	3000	DYNAMIC3C	169070	169070	0	138	51	1	190	0	30

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
1	4	3000	DYNAMIC3C2	127582	127582	0	139	52	0	191	0	23
2	1	1200	DETERM.	180060	180060	0				180	0	100
2	1	1200	PASSIVE	170640	0	170640				0	142	95
2	1	1200	STATIC	157880	101000	56880				101	47	88
2	1	1200	DYNAMICH	150936	150936	0				151	11	84
2	1	1200	DYNAMIC	144462	131502	12960				132	11	80
2	1	1200	DYNAMIC3C	118706	110066	8640	137	0	0	137	7.2	66
2	1	1600	DETERM.	180060	180060	0				180	0	100
2	1	1600	PASSIVE	227520	0	227520				0	142	126
2	1	1600	STATIC	176840	101000	75840				101	47	98
2	1	1600	DYNAMICH	150936	150936	0				151	11	84
2	1	1600	DYNAMIC	147702	147702	0				148	0	82
2	1	1600	DYNAMIC3C	118769	118769	0	148	0	0	148	0	66
2	1	2000	DETERM.	180060	180060	0				180	0	100
2	1	2000	PASSIVE	284400	0	284400				0	142	158
2	1	2000	STATIC	180060	180060	0				180	0	100
2	1	2000	DYNAMICH	150936	150936	0				151	11	84
2	1	2000	DYNAMIC	147702	147702	0				148	0	82
2	1	2000	DYNAMIC3C	118769	118769	0	148	0	0	148	0	66
2	1	3000	DETERM.	180060	180060	0				180	0	100
2	1	3000	PASSIVE	426600	0	426600				0	142	237
2	1	3000	STATIC	180060	180060	0				180	0	100
2	1	3000	DYNAMICH	150936	150936	0				151	11	84
2	1	3000	DYNAMIC	147702	147702	0				148	0	82
2	1	3000	DYNAMIC3C	118769	118769	0	148	0	0	148	0	66
2	1	3000	DYNAMIC3C2	64086	64086	0	149	0	0	149	0	36
2	2	1200	DETERM.	110480	101000	9480				101	8	100
2	2	1200	PASSIVE	108360	0	108360				0	90	98
2	2	1200	STATIC	100560	39000	61560				39	51	91
2	2	1200	DYNAMICH	90817	87817	3000				88	2.5	82
2	2	1200	DYNAMIC	90677	88517	2160				89	2	82
2	2	1200	DYNAMIC3C	73177	71017	2160	89	0	0	89	1.8	65
2	2	1600	DETERM.	113640	101000	12640				101	8	100

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
2	2	1600	PASSIVE	108360	0	108360				0	90	95
2	2	1600	STATIC	113640	101000	12640				101	8	100
2	2	1600	DYNAMICH	91817	87817	4000				88	2.5	80
2	2	1600	DYNAMIC	91397	88517	2880				89	2	80
2	2	1600	DYNAMIC3C	73897	71017	2880	89	0	0	89	1.8	65
2	2	2000	DETERM.	116800	101000	15800				101	8	100
2	2	2000	PASSIVE	180600	0	180600				0	90	155
2	2	2000	STATIC	116800	101000	15800				101	8	100
2	2	2000	DYNAMICH	92817	87817	5000				88	2.5	80
2	2	2000	DYNAMIC	92117	88517	3600				89	2	79
2	2	2000	DYNAMIC3C	74617	71017	3600	89	0	0	89	1.8	64
2	2	3000	DETERM.	124700	101000	23700				101	8	100
2	2	3000	PASSIVE	270900	0	270900				0	90	217
2	2	3000	STATIC	124700	101000	23700				101	8	100
2	2	3000	DYNAMICH	95317	87817	7500				88	2.5	76
2	2	3000	DYNAMIC	93917	88517	5400				89	2	75
2	2	3000	DYNAMIC3C	76417	71017	5400	89	0	0	89	1.8	61
2	2	3000	DYNAMIC3C2	43726	38326	5400	89	0	0	89	1.8	35
2	3	1200	DETERM.	78240	39000	39240				39	33	100
2	3	1200	PASSIVE	58200	0	58200				0	49	74
2	3	1200	STATIC	78240	39000	39240				39	33	100
2	3	1200	DYNAMICH	72077	69917	2160				70	2	92
2	3	1200	DYNAMIC	72077	69917	2160				70	2	92
2	3	1200	DYNAMIC3C	58249	56089	2160	70	0	0	70	1.8	74
2	3	1600	DETERM.	91320	39000	52320				39	33	100
2	3	1600	PASSIVE	77600	0	77600				0	49	74
2	3	1600	STATIC	91320	39000	52320				39	33	100
2	3	1600	DYNAMICH	72797	69917	2880				70	2	80
2	3	1600	DYNAMIC	72797	69917	2880				70	2	80
2	3	1600	DYNAMIC3C	58969	56089	2880	70	0	0	70	1.8	65
2	3	2000	DETERM.	104400	39000	65400				39	33	100
2	3	2000	PASSIVE	97000	0	97000				0	49	93
2	3	2000	STATIC	104400	39000	65400				39	33	100

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
2	3	2000	DYNAMICH	73517	73517	0				74	0	70
2	3	2000	DYNAMIC	73517	73517	0				74	0	70
2	3	2000	DYNAMIC3C	59689	56089	3600	70	0	0	70	1.8	65
2	3	3000	DETERM.	137100	39000	98100				39	33	100
2	3	3000	PASSIVE	145500	0	145500				0	49	106
2	3	3000	STATIC	124700	101000	23700				101	8	91
2	3	3000	DYNAMICH	73517	73517	0				74	0	54
2	3	3000	DYNAMIC	73517	73517	0				74	0	54
2	3	3000	DYNAMIC3C	61489	56089	5400	70	0	0	70	1.8	45
2	3	3000	DYNAMIC3C2	33404	33404	0	77	0	0	77	0	24
2	4	1200	DETERM.	180060	180060	0				180	0	100
2	4	1200	PASSIVE	128800	0	128800				0	107	71
2	4	1200	STATIC	121080	39000	82080				39	68	67
2	4	1200	DYNAMICH	116990	116990	0				117	0	65
2	4	1200	DYNAMIC	108682	101338	7344				101	6	60
2	4	1200	DYNAMIC3C	88761	81417	7344	102	0	0	102	6.1	49
2	4	1600	DETERM.	180060	180060	0				180	0	100
2	4	1600	PASSIVE	171840	0	171840				0	107	95
2	4	1600	STATIC	143976	101000	42976				101	27	80
2	4	1600	DYNAMICH	116990	116990	0				117	0	65
2	4	1600	DYNAMIC	111130	101338	9792				101	6	62
2	4	1600	DYNAMIC3C	91112	91112	0	114	0	0	114	0	51
2	4	2000	DETERM.	180060	180060	0				180	0	100
2	4	2000	PASSIVE	214800	0	214800				0	107	119
2	4	2000	STATIC	154720	101000	53720				101	27	86
2	4	2000	DYNAMICH	116990	116990	0				117	0	65
2	4	2000	DYNAMIC	113398	113398	0				113	0	63
2	4	2000	DYNAMIC3C	91112	91112	0	114	0	0	114	0	51
2	4	3000	DETERM.	180060	180060	0				180	0	100
2	4	3000	PASSIVE	322200	0	322200				0	107	179
2	4	3000	STATIC	180060	180060	0				180	0	100
2	4	3000	DYNAMICH	116990	116990	0				117	0	65
2	4	3000	DYNAMIC	113398	113398	0				113	0	63

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
2	4	3000	DYNAMIC3C	91112	91112	0	114	0	0	114	0	51
2	4	3000	DYNAMIC3C2	49205	49205	0	114	0	0	114	0	27
3	1	1200	DETERM.	461750	276350	185400				276	155	100
3	1	1200	PASSIVE	488880	0	488880				0	407	109
3	1	1200	STATIC	446750	203510	243240				204	202	96
3	1	1200	DYNAMICH	420556	357556	63000				358	53	91
3	1	1200	DYNAMIC	420556	357556	63000				358	53	91
3	1	1200	DYNAMIC3C	367992	296712	71280	307	44	0	351	59	80
3	1	1600	DETERM.	523550	276350	247200				276	155	100
3	1	1600	PASSIVE	651840	0	651840				0	407	124
3	1	1600	STATIC	521550	234990	286560				235	179	100
3	1	1600	DYNAMICH	441556	357556	84000				358	53	84
3	1	1600	DYNAMIC	441556	357556	84000				358	53	84
3	1	1600	DYNAMIC3C	391067	296835	94232	308	45	2	355	59	75
3	1	2000	DETERM.	585350	276350	309000				276	155	100
3	1	2000	PASSIVE	814800	0	814800				0	407	139
3	1	2000	STATIC	585350	294750	290600				155	145	100
3	1	2000	DYNAMICH	462556	357556	105000				358	53	79
3	1	2000	DYNAMIC	462556	357556	105000				358	53	79
3	1	2000	DYNAMIC3C	413899	300699	113200	309	44	3	356	57	70
3	1	3000	DETERM.	739850	276350	463500				276	155	100
3	1	3000	PASSIVE	1222200	0	1222200				0	407	165
3	1	3000	STATIC	676490	616490	60000				616	20	91
3	1	3000	DYNAMICH	515056	357556	157500				358	53	70
3	1	3000	DYNAMIC	494836	475636	19200				476	6	67
3	1	3000	DYNAMIC3C	443681	408581	35100	373	94	3	470	12	60
3	1	3000	DYNAMIC3C2	320131	226831	93300	377	48	1	426	31	43
3	2	1200	DETERM.	786020	773540	12480				774	10	100
3	2	1200	PASSIVE	782880	0	782880				0	652	100
3	2	1200	STATIC	717470	564470	153000				564	128	91
3	2	1200	DYNAMICH	727043	727043	0				727	0	92
3	2	1200	DYNAMIC	677127	641127	3600				641	30	86
3	2	1200	DYNAMIC3C	602428	482068	120360	482	85	0	567	100	77

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
3	2	1600	DETERM.	790180	773540	16640				774	10	100
3	2	1600	PASSIVE	1043840	0	1043840				0	652	132
3	2	1600	STATIC	769790	564990	204800				564	128	97
3	2	1600	DYNAMICH	727043	727043	0				727	0	92
3	2	1600	DYNAMIC	688019	647699	40320				648	25	87
3	2	1600	DYNAMIC3C	622973	574013	48960	478	164	5	647	31	79
3	2	2000	DETERM.	794340	773540	20800				774	10	100
3	2	2000	PASSIVE	1304800	0	1304800				0	652	164
3	2	2000	STATIC	793040	748440	44600				748	22	100
3	2	2000	DYNAMICH	727043	727043	0				727	0	92
3	2	2000	DYNAMIC	698099	648099	50000				648	25	87
3	2	2000	DYNAMIC3C	634817	577217	57600	477	164	8	649	29	80
3	2	3000	DETERM.	804740	773540	31200				774	10	100
3	2	3000	PASSIVE	1957200	0	1957200				0	652	243
3	2	3000	STATIC	796000	778000	18000				778	6	99
3	2	3000	DYNAMICH	727043	727043	0				727	0	90
3	2	3000	DYNAMIC	716413	682213	34200				682	11	89
3	2	3000	DYNAMIC3C	661657	596856	64800	477	168	16	661	22	82
3	2	3000	DYNAMIC3C2	522109	424909	97200	484	167	0	651	32	65
3	3	1200	DETERM.	814724	773540	41184				774	34	100
3	3	1200	PASSIVE	640008	0	640008				0	533	79
3	3	1200	STATIC	597758	236486	361272				236	301	73
3	3	1200	DYNAMICH	652634	652634	0				652	0	80
3	3	1200	DYNAMIC	556420	440752	115668				441	96	68
3	3	1200	DYNAMIC3C	497265	364989	132276	363	65	0	428	110	61
3	3	1600	DETERM.	828452	773540	54912				774	34	100
3	3	1600	PASSIVE	853344	0	853344				0	533	103
3	3	1600	STATIC	702910	564990	137920				565	86	84
3	3	1600	DYNAMICH	652634	652634	0				652	0	79
3	3	1600	DYNAMIC	588771	554627	34144				555	21	71
3	3	1600	DYNAMIC3C	524757	422900	101857	429	67	3	499	64	63
3	3	2000	DETERM.	842180	773540	68640				774	34	100
3	3	2000	PASSIVE	1066680	0	1066680				0	533	127

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
3	3	2000	STATIC	737390	564990	172400				565	86	88
3	3	2000	DYNAMICH	652634	652634	0				652	0	77
3	3	2000	DYNAMIC	596502	553742	42760				554	21	70
3	3	2000	DYNAMIC3C	539635	487355	52280	428	123	4	555	26	64
3	3	3000	DETERM.	876500	773540	102960				774	34	100
3	3	3000	PASSIVE	1600620	0	1600620				0	533	183
3	3	3000	STATIC	822860	619340	203520				619	68	94
3	3	3000	DYNAMICH	652634	652634	0				652	0	74
3	3	3000	DYNAMIC	612466	565606	46860				566	16	70
3	3	3000	DYNAMIC3C	561569	362389	199180	425	15	2	442	66	64
3	3	3000	DYNAMIC3C2	430662	362282	68380	436	134	0	570	23	49
3	4	1200	DETERM.	603950	557990	45960				558	34	100
3	4	1200	PASSIVE	588960	0	588960				0	491	98
3	4	1200	STATIC	546710	267590	279120				267	233	91
3	4	1200	DYNAMICH	603950	601190	2760				601	2	100
3	4	1200	DYNAMIC	511896	402816	109080				403	91	85
3	4	1200	DYNAMIC3C	455372	390932	64440	401	62	0	463	54	75
3	4	1600	DETERM.	619270	557990	61280				558	34	100
3	4	1600	PASSIVE	785280	0	785280				0	491	127
3	4	1600	STATIC	619270	557990	61280				558	34	100
3	4	1600	DYNAMICH	615676	611996	3680				612	37	99
3	4	1600	DYNAMIC	535760	523600	12160				523	8	87
3	4	1600	DYNAMIC3C	475635	397075	78560	402	63	3	468	49	75
3	4	2000	DETERM.	634590	557990	76600				558	34	100
3	4	2000	PASSIVE	981600	0	981600				0	491	155
3	4	2000	STATIC	634590	557990	76600				558	34	100
3	4	2000	DYNAMICH	616596	611996	4600				623	2	97
3	4	2000	DYNAMIC	538800	523600	15200				524	8	85
3	4	2000	DYNAMIC3C	481268	464268	1700	403	122	3	528	9	76
3	4	3000	DETERM.	672890	557990	114900				558	34	100
3	4	3000	PASSIVE	1472400	0	1472400				0	491	219
3	4	3000	STATIC	672890	557990	114900				558	34	100
3	4	3000	DYNAMICH	618896	611996	6900				612	2	92

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
3	4	3000	DYNAMIC	545132	526532	18600				527	6	81
3	4	3000	DYNAMIC3C	489286	468286	21000	405	123	4	532	7	73
3	4	3000	DYNAMIC3C2	364363	343063	21300	409	129	0	538	7	54
4	1	1600	DETERM.	896070	491430	404640				491	253	100
4	1	1600	PASSIVE	656800	0	656800				0	411	73
4	1	1600	STATIC	562120	235720	326400				236	204	63
4	1	1600	DYNAMICH	460668	457308	3360				457	2	51
4	1	1600	DYNAMIC	423330	398210	25120				398	16	47
4	1	1600	DYNAMIC3C	402355	360755	41600	248	129	12	389	26	45
5	1	1600	DETERM.	140740	137060	3680				137	2	100
5	1	1600	PASSIVE	205920	0	205920				0	129	146
5	1	1600	STATIC	140740	137060	3680				137	2	100
5	1	1600	DYNAMICH	128754	128754	0				129	0	91
5	1	1600	DYNAMIC	128754	128754	0				129	0	91
5	1	1600	DYNAMIC3C	103507	103507	0	129	0	0	129	0	73
6	1	1600	DETERM.	630820	607940	22880				607	14	100
6	1	1600	PASSIVE	654880	0	654880				0	409	104
6	1	1600	STATIC	587230	176350	410880				176	256	93
6	1	1600	DYNAMICH	411760	411760	0				412	0	70
6	1	1600	DYNAMIC	411760	411760	0				412	0	70
6	1	1600	DYNAMIC3C	355470	355470	0	344	70	0	414	0	56
7	1	1600	DETERM.	548320	92000	456320				92	285	100
7	1	1600	PASSIVE	600960	0	600960				0	375	110
7	1	1600	STATIC	548320	92000	456320				92	285	100
7	1	1600	DYNAMICH	481881	481881	0				482	0	88
7	1	1600	DYNAMIC	481881	481881	0				482	0	88
7	1	1600	DYNAMIC3C	465967	441150	24817	273	173	19	465	16	84
8	1	1600	DETERM.	998910	988990	9920				989	6	100
8	1	1600	PASSIVE	1280000	0	1280000				0	800	128
8	1	1600	STATIC	935350	690230	245120				690	153	94
8	1	1600	DYNAMICH	852548	841348	11200				841	7	85
8	1	1600	DYNAMIC	834780	775100	59680				775	37	84
8	1	1600	DYNAMIC3C	778501	677062	101439	515	225	8	748	63	78

CAP.CASE	PROBS.	AIRCOST	SOLUTION	EXPCOST	EGNDCOST	EAIRCOST	EGD1	EGD2	EGD3	EGDT	EAD	%VSDETERM
9	1	1600	DETERM.	1222070	701590	520480				702	325	100
9	1	1600	PASSIVE	1073440	0	1073440				0	671	89
9	1	1600	STATIC	942230	346390	595840				346	372	77
9	1	1600	DYNAMICH	710825	710825	0				711	0	58
9	1	1600	DYNAMIC	710054	710054	0				710	0	58
9	1	1600	DYNAMIC3C	637605	629925	7680	526	173	10	709	5	52
10	1	3000	DETERM.	1293520	877720	415800				878	139	100
10	1	3000	PASSIVE	2056200	0	2056200				0	685	159
10	1	3000	STATIC	1103570	780470	323100				780	108	85
10	1	3000	DYNAMICH	754198	754198	0				754	0	58
10	'1	3000	DYNAMIC	738683	705683	33000				705	11	57
10	1	3000	DYNAMIC3C	688664	616664	72000	541	142	18	701	24	53
10	1	3000	DYNAMIC3C2	528583	445783	82800	560	147	6	713	28	40