

# STOCHASTIC PROGRAMMING APPROACHES TO AIR TRAFFIC FLOW MANAGEMENT UNDER THE UNCERTAINTY OF WEATHER

A Thesis  
Presented to  
The Academic Faculty

by

Yu-Heng Chang

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of Industrial and Systems Engineering

Georgia Institute of Technology  
December 2010

**STOCHASTIC PROGRAMMING APPROACHES TO AIR  
TRAFFIC FLOW MANAGEMENT UNDER THE  
UNCERTAINTY OF WEATHER**

Approved by:

Professor Ellis L. Johnson, Advisor  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Professor John-Paul B. Clarke,  
Co-Advisor  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Professor Senay Solak  
Department of Finance and  
Operations Management  
*University of Massachusetts, Amherst*

Professor Shabbir Ahmed  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Professor Joel S. Sokol  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Date Approved: August 2, 2010

## ACKNOWLEDGEMENTS

I would like to thank my advisors Ellis Johnson and John-Paul Clarke for their support through my graduate studies. I cannot dive into such an interesting topic without their professional knowledge and experience. I would also show my appreciation for their financial assistance for not only student life but also opportunities to attend conferences, which broaden my views of academic research.

I am thankful to Senay Solak for his friend-like support which always gives me confidence to continue. I appreciate the valuable insights and suggestions from Shabbir Ahmed and Joel Sokol. I am also grateful to the members in the Air Transportation Laboratory for their advisory comments, especially Liling Ren.

The friendship I have earned in Georgia Tech is important to me, thank them all, especially Tsung-lin Wu, Che-yu Chang, and Wen-wei Cao, for their tolerance when I was screaming. And I do appreciate the distant friends - Evien Wu and Janet Chiu - for their patience to share the good and bad I have faced.

Finally, I would like to express my deepest love to my parents, brother, Caori, and fiancé Jin-Jyh Su for their unconditioned love. I could not have gone through the whole process without their ultimate encouragement and inspiration.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . .	iii
LIST OF TABLES . . . . .	vi
LIST OF FIGURES . . . . .	viii
SUMMARY . . . . .	x
I INTRODUCTION . . . . .	1
1.1 Air Traffic Flow Management . . . . .	2
1.2 Stochastic Programming . . . . .	5
1.2.1 Two-Stage Stochastic Programming Models . . . . .	5
1.2.2 Multistage Stochastic Programming Models . . . . .	6
1.3 Approaches to Stochastic Programs . . . . .	10
1.3.1 Decomposition Methods . . . . .	10
1.3.2 Statistical Methods . . . . .	12
1.3.3 Stochastic Decomposition . . . . .	13
1.4 Thesis Outline . . . . .	13
II LITERATURE REVIEW . . . . .	15
2.1 Air Traffic Flow Management Problems . . . . .	16
2.2 Stochastic Programming Approaches in Other Fields . . . . .	22
III TWO-STAGE STOCHASTIC PROGRAMMING APPROACH TO TRAF- FIC FLOW MANAGEMENT PROBLEM . . . . .	25
3.1 Two-Stage Stochastic Integer Programming Model for Traffic Flow Management Problem . . . . .	26
3.1.1 General Assumptions . . . . .	26
3.1.2 The Two-Stage Stochastic Integer Programming Model . . . . .	28
3.1.3 Experimental Setting . . . . .	31
3.1.4 Computational Results . . . . .	35
3.2 Approaches to the Two-Stage Model: The Rolling Horizon Method	38

3.2.1	The Rolling Horizon Method . . . . .	39
3.2.2	Computational Experiments . . . . .	41
3.2.3	Justification of the Rolling Horizon Method . . . . .	44
3.3	Lagrangian Relaxation . . . . .	45
3.3.1	Computational Results . . . . .	53
IV	MULTISTAGE STOCHASTIC PROGRAMMING APPROACH TO TRAFFIC FLOW MANAGEMENT PROBLEM . . . . .	63
4.1	Three-Stage Stochastic Integer Programming Model for Traffic Flow Management Problem . . . . .	65
4.1.1	The Three-Stage Stochastic Integer Programming Model . . . . .	66
4.1.2	Experimental Setting . . . . .	70
4.1.3	Computational Results . . . . .	70
4.2	Approaches to the Three-Stage Model: The Rolling Horizon Method . . . . .	73
4.2.1	The Rolling Horizon Method . . . . .	73
4.2.2	Computational Experiments . . . . .	74
4.3	Lagrangian Relaxation . . . . .	75
4.3.1	Computational Results . . . . .	82
V	CONCLUSION AND FUTURE WORK . . . . .	90
5.1	Conclusion . . . . .	90
5.2	Future Work . . . . .	93
	REFERENCES . . . . .	95

## LIST OF TABLES

1	Summary of the existing literature applying mathematical programming approaches to traffic flow management and the proposed methodology . . . . .	21
2	List of bounds on possible flight actions . . . . .	29
3	List of cost parameters . . . . .	29
4	Ground-delay costs . . . . .	32
5	Air-holding costs . . . . .	32
6	An FCA capacity forecast matrix illustration . . . . .	35
7	Computational results for two-stage stochastic integer programming model . . . . .	36
8	List of numbers of columns and rows for two-stage model . . . . .	37
9	Example of how flights are chosen in the rolling horizon method . . .	42
10	Solution format for the rolling horizon method . . . . .	42
11	Final solutions for the rolling horizon method . . . . .	43
12	Second-stage solution format . . . . .	43
13	Computational results for two-stage rolling horizon method - PD=9 .	44
14	Computational results for two-stage rolling horizon method - PD=10	45
15	Computational results for two-stage rolling horizon method - PD=11	46
16	Computational results for two-stage rolling horizon method - flight count=22 . . . . .	47
17	Computational results for two-stage rolling horizon method - flight count=27 . . . . .	48
18	Computational results for two-stage rolling horizon method - flight count=28 . . . . .	49
19	Computational results for two-stage rolling horizon method - flight count=31 . . . . .	50
20	Computational results for two-stage rolling horizon method - flight count=35 . . . . .	51
21	List of different parameters in subgradient method for two-stage model, flight count=22 . . . . .	55

22	Computational comparison among optimality, rolling horizon method, and Lagrangian relaxation for two-stage model . . . . .	57
23	Computational time of the subgradient method for two-stage model . . . . .	58
24	List of parameters in subgradient method producing the best lower bounds for two-stage model . . . . .	58
25	Computational results for three-stage stochastic integer programming model . . . . .	71
26	List of numbers of columns and rows for three-stage model . . . . .	72
27	Computational results for three-stage rolling horizon method - PD=9 . . . . .	75
28	Computational results for three-stage rolling horizon method - PD=10 . . . . .	75
29	Computational results for three-stage rolling horizon method - PD=11 . . . . .	76
30	List of different parameters in subgradient method for three-stage model, flight count=22 . . . . .	84
31	Computational comparison among optimality, rolling horizon method, and Lagrangian relaxation for three-stage model . . . . .	84
32	Computational time of the subgradient method for three-stage model . . . . .	85
33	List of parameters in subgradient method producing the best lower bounds for three-stage model . . . . .	86

## LIST OF FIGURES

1	Causes of NAS delays in 2006 . . . . .	3
2	An illustration of National Airspace System . . . . .	4
3	Node-based implementation of the nonanticipativity constraints . . .	8
4	Scenario-based implementation of the nonanticipativity constraints .	9
5	Cost composition for two-stage model . . . . .	38
6	An illustration of the rolling horizon method . . . . .	39
7	Comparison of paths between two parameters for two-stage model, flight count=22 . . . . .	56
8	Path of subgradient method for two-stage model, flight count=22 . .	59
9	Path of subgradient method for two-stage model, flight count=27 . .	59
10	Path of subgradient method for two-stage model, flight count=28 . .	60
11	Path of subgradient method for two-stage model, flight count=31 . .	60
12	Path of subgradient method for two-stage model, flight count=35 . .	61
13	Path of subgradient method for two-stage model, flight count=44 . .	61
14	Path of subgradient method for two-stage model, flight count=52 . .	62
15	Path of subgradient method for two-stage model, flight count=60 . .	62
16	An illustration of multistage problem in traffic flow management . . .	64
17	Implementations of three-stage stochastic programming model . . . .	66
18	Comparison between 2- and 3-stage models . . . . .	71
19	Cost composition for three-stage model . . . . .	73
20	Comparison of speed-change costs in different stages . . . . .	74
21	Comparison of paths between two parameters for three-stage model, flight count=22 . . . . .	85
22	Path of subgradient method for three-stage model, flight count=22 . .	86
23	Path of subgradient method for three-stage model, flight count=27 . .	87
24	Path of subgradient method for three-stage model, flight count=28 . .	87
25	Path of subgradient method for three-stage model, flight count=31 . .	88



26	Path of subgradient method for three-stage model, flight count=35 . .	88
27	Path of subgradient method for three-stage model, flight count=44 . .	89

## SUMMARY

As air traffic congestion grows, air traffic flow management (ATFM) is becoming a great concern. ATFM deals with air traffic and the efficient utilization of the airport and airspace. Air traffic efficiency is heavily influenced by unanticipated factors, or uncertainties, which can come from several sources such as mechanical breakdown; however, weather is the main unavoidable cause of uncertainty. Because weather is unpredictable, it poses a critical challenge for ATFM in current airport and airspace operations. Convective weather results in congestion at airports as well as in airspace sectors. During times of congestion, the decision as how and when to send aircraft toward an airspace sector in the presence of weather is difficult. To approach this problem, we first propose a two-stage stochastic integer program by emphasizing a given single sector. By considering ground delay, cancellation, and cruise speed for each flight on the ground in the first stage, as well as air holding and diversion recourse actions for each flight in the air in the second stage, our model determines how aircraft are sent toward a sector under the uncertainty of weather. However, due to the large number of weather scenarios, the model is intractable in practice. To overcome the intractability, we suggest a rolling horizon method to solve the problem to near optimal. Lagrangian relaxation and subgradient method are used to justify the rolling horizon method. Since the rolling horizon method can be solved in real time, we can apply it to actual aircraft schedules to reduce the costs incurred on the ground as well as in airspace. We then extend our two-stage model to a multistage stochastic program, which increases the number of possible weather realizations and results a more efficient schedule in terms of costs. The rolling horizon method as well as Lagrangian relaxation and subgradient method are applied to this multistage

model. An overall comparison among the previously described methodologies are presented.

# CHAPTER I

## INTRODUCTION

Most real life decisions involve uncertainty. To incorporate the uncertain factors into mathematical programming models, stochastic programming (SP) is a sound approach. The idea of SP is first introduced by Dantzig (1955) and is being used in several fields such as capacity expansion, financial planning, or airline scheduling problems. The purpose of a stochastic program is to find a decision process that minimizes or maximizes the expected value over all possible realizations under uncertainties. An important assumption in SP is that the probability distribution of the uncertain parameter is known. Two-stage SP models are widely studied and can be readily extended to multistage stochastic programs.

In airline operations, the uncertainties come from several parts such as mechanical breakdown or the unexpected unavailability of flights or crew. Among these uncertain factors, weather is an unavoidable one. Air traffic flow management (TFM) under uncertain weather is an important issue in current airport and airspace operations. Indeed, convective weather results in congestion at airports as well as in airspace sectors. Since the weather is uncertain, the issue of how to deal with the aircraft departures in response to the weather and its corresponding impact on the capacity availability is a difficult problem. Stochastic programs are often used to solve such probabilistic problems. In the following sections, we introduce the air traffic flow management problems that we are facing and the general concepts about stochastic programs.

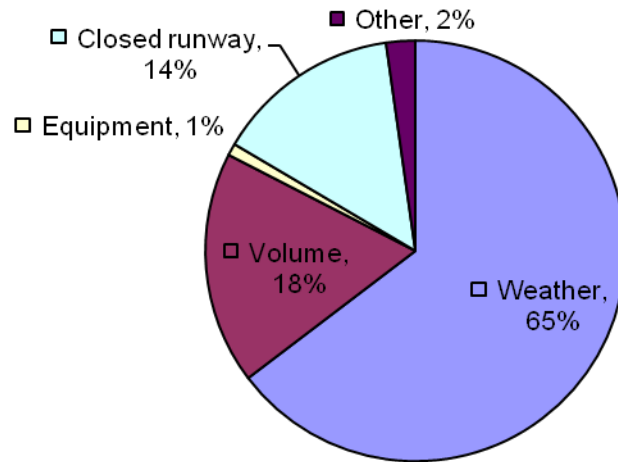
## ***1.1 Air Traffic Flow Management***

Today, air transportation is one of the most convenient transportation tools. Due to the high demands, airlines provide a variety of flight schedules for passengers to choose from. However, the large number of flights compete for the capacitated airport and airspace resources, such as runway facilities and airspace capacities, respectively. Even worse, convective weather such as thunderstorms or snowstorms play important uncertain roles in the airline industry. Storms can cause the capacities of the resources to reduce, however, the strength and the time of the storms are difficult to correctly predict. The focus of the air traffic flow management is to identify and resolve the imbalance between demand and supply of the limiting resources.

The United States Congress recognized this problem and established the Joint Planning and Development Office (JPDO) to coordinate several agencies, which include National Aeronautics and Space Administration (NASA) and Federal Aviation Administration (FAA), to develop and implement the Next Generation Air Transportation System (NextGen) to continuously improve the performance of the entire National Airspace System (NAS). NextGen includes projects involving runway utilization, flights separation, integrated weather prediction, lower-emission aircraft, and dynamic resource allocation. The air traffic flow management problem we focused here is one of the projects, and the main purpose of this project is to develop methodologies for a higher capacity system. One part of our team focus on determining the capacity of the airspace given the weather forecast, and the other part determines the number of aircraft to send toward the airspace given the determined capacities. The goal of this thesis is to develop an optimization algorithm to effectively solve the TFM problem in the presence of weather uncertainties.

Weather affects the airline operations a lot. According to Bureau of Transportation Statistics (BTS), in 2006, 64.67 percent of NAS delays in terms of number of

operations were due to weather (see Figure 1) and it corresponds to 19,454,394 delayed minutes, which accounts for 76 percent of NAS delayed minutes. If we introduce the delay attributed to air carriers and security, weather still accounts for 44.2 percent of delay minutes.

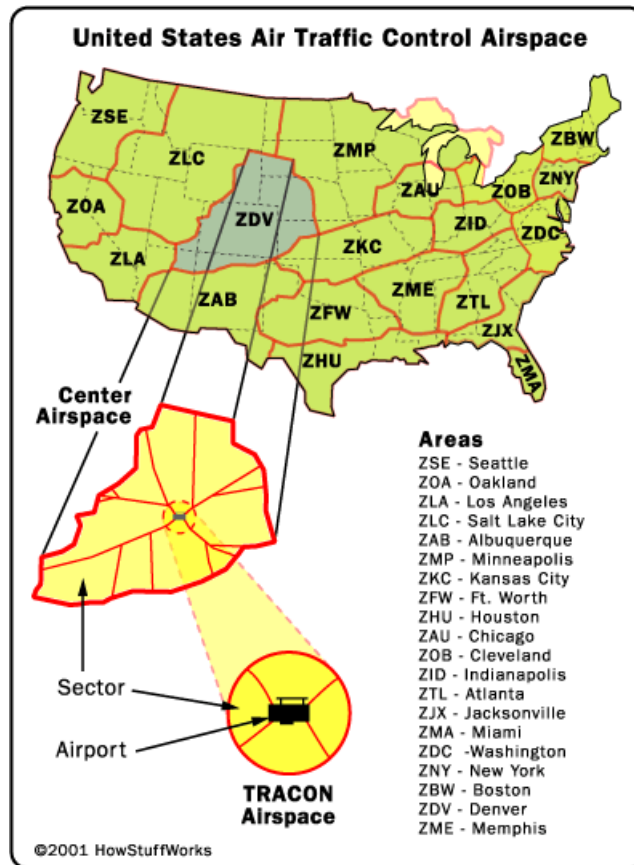


**Figure 1:** Causes of NAS delays in 2006

Current NAS operations under the weather uncertainty depend highly on the air traffic controller's past experience. If he is optimistic, aircraft might be departed and later find no way to pass through a thunderstorm. A diversion decision might be made which costs the airline a lot. Or if he is pessimistic, aircraft might be sitting in the airports and later find the storm disappeared which causes the waste of capacities. Therefore, it is important to develop a dynamic algorithm so that decisions such as when and how the aircraft should be sent in response to the different weather forecasts can easily be made.

NAS supports all aircraft operations in the United States. The primary functions of NAS are to communicate, navigate, and manage flights. Because of its complexity, which consists of a collection of facilities, procedures, airports, and thousands of people, NAS is divided into twenty-one Air Route Traffic Control Centers (ARTCC), or simply centers. A center's main task is to control and separate flights within a

designated airspace, which can range over 100,000 square miles and cover from the base of the airspace up to 60,000 feet high. Each of the centers is divided into several air traffic control (ATC) sectors, which range from 50 to 200 miles wide, see Figure 2 for a general idea of the NAS. Air traffic controllers manage flights based on arrival and departure rates in airports, airport weather conditions, en route congestion, and en route weather. Since an airspace sector is the fundamental operation unit for the air traffic control, in this study, we target the problem at sector level.



**Figure 2:** An illustration of National Airspace System

Weather forecast is notoriously uncertain. Therefore, a deterministic traffic flow management model might not be able to fully capture the uncertainties which may leave the capacities unused. The capacity is lost if aircraft are re-routed to a longer paths while shorter paths are expected to be blocked, but turns out that the weather is clear enough for the aircraft to fly the shorter paths. Likewise, capacity is lost

if the aircraft are delayed on the ground for longer than actually necessary. The problems of lost of capacities might be reduced if the weather forecast becomes more reliable. Nevertheless, the uncertainties will not be eliminated such that a deterministic TFM is sufficient. As a result, stochastic programs are used to include the weather uncertainties.

In Chapter 3 and 4, we will describe in details how we incorporate the weather uncertainties into the stochastic programming model. However, in the following section, we will first introduce the general concepts of stochastic programs.

## ***1.2 Stochastic Programming***

Stochastic programming techniques are often used when uncertainty involves. In the following, two-stage and multistage stochastic programs are introduced and the solution methodologies used to solve the stochastic programs are also presented.

### **1.2.1 Two-Stage Stochastic Programming Models**

In a two-stage SP model with fixed recourse, decisions are made at the current point knowing only a set of possible outcomes of the uncertain factors while minimizing or maximizing the overall expected value. The overall expected value contains a deterministic term of first-stage decisions and an expected form of second-stage decisions, which based on different realizations of the uncertain factors. We say the second-stage decisions are the recourse actions as one of them will be picked depending on how the uncertainty is revealed.

A classical two-stage stochastic program can be written as (1). First-stage decision variables are represented by  $x$  and the uncertain factors are denoted by  $\xi$ . Based on different  $q(\xi)$ ,  $h(\xi)$ , and  $T(\xi)$  under uncertainties, a set of second-stage decisions of



$y(\xi)$  are projected.

$$\begin{aligned}
\min \quad & c^T x + \mathbf{E}_\xi [\min q(\xi)^T y(\xi)] \\
\text{s.t.} \quad & Ax = b \\
& T(\xi)x + Wy(\xi) = h(\xi) \\
& x \geq 0, y(\xi) \geq 0
\end{aligned} \tag{1}$$

If the uncertainties can be interpreted as a discrete probability distribution, that is, the number of outcome is finite, then we can describe each of the outcome as  $\omega \in \Omega$ , with corresponding probability of  $p_\omega$ ,  $\sum_{\omega \in \Omega} p_\omega = 1$ . As a result, the stochastic program of (1) can be interpreted in a deterministic equivalent form of (2).

$$\begin{aligned}
\min \quad & c^T x + \sum_{\omega \in \Omega} p_\omega q_\omega^T y_\omega \\
\text{s.t.} \quad & Ax = b \\
& T_\omega x + W y_\omega = h_\omega \quad \forall \omega \\
& x \geq 0 \\
& y_\omega \geq 0 \quad \forall \omega
\end{aligned} \tag{2}$$

In a real-life problem, if we can discretize the uncertainties into several possibilities and identify the corresponding probabilities, a stochastic programming model can easily be constructed by the ideas of deterministic equivalent form. However, if the number of  $\omega$  is large, we are not able to solve the problem directly. Instead, decomposition method might be required to make the problem smaller. We will discuss the approaches in Section 1.3.

### 1.2.2 Multistage Stochastic Programming Models

A multistage stochastic program with recourse is an extension of the two-stage SP model. In a two-stage SP model, a set of decisions are made once and kept forever.

However, in a multistage stochastic program, a set of recourse decisions, which constitute a decision process, are made consecutively over time. We call them "recourse decisions" because these recourse decisions are made at the time we solve the multistage model, not at the time the decisions are to be executed. The multistage model is solved based on everything we know or we predict at the time we solve the problem, including the numerous possibilities.

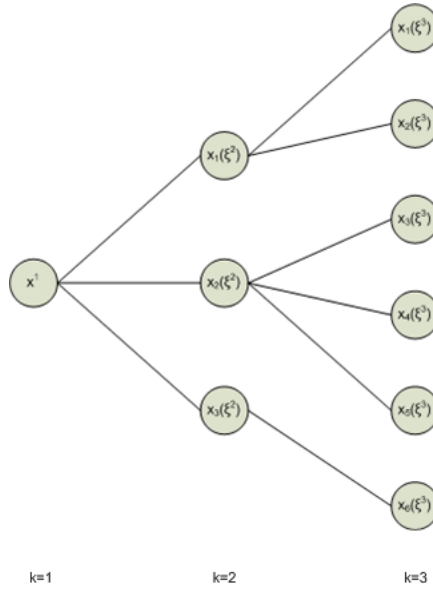
To solve a multistage stochastic program, we need to construct a set of probabilities corresponding to each stage. A series of decisions that we are about to execute including recourse actions are made based on the possible outcomes and the corresponding probabilities in the future stages. As time evolves, recourse decisions are being executed depending on the realizations of the uncertain factors at each stage. The advantage over two-stage model is that, as we are closer to the time when recourse decisions are to be executed, we can be more certain about the future. Therefore, the prediction of the outcomes with corresponding probabilities can be more accurate.

A  $K$ -stage stochastic program can be modeled as follows.

$$\begin{aligned}
 \min \quad & c^1 x^1 + \mathbf{E}_{\xi^2} \left[ \min c(\xi^2) x(\xi^2) + \cdots + \mathbf{E}_{\xi^K} \left[ \min c(\xi^K) x(\xi^K) \right] \right] \\
 \text{s.t.} \quad & W^1 x^1 = h^1 \\
 & T(\xi^1) x^1 + W^2 x(\xi^2) = h(\xi^2) \\
 & \quad \quad \quad \vdots \\
 & T(\xi^{K-1}) x(\xi^{K-1}) + W^K x(\xi^K) = h(\xi^K) \\
 & x^1 \geq 0 \\
 & x(\xi^k) \geq 0 \quad k = 2, \dots, K
 \end{aligned} \tag{3}$$

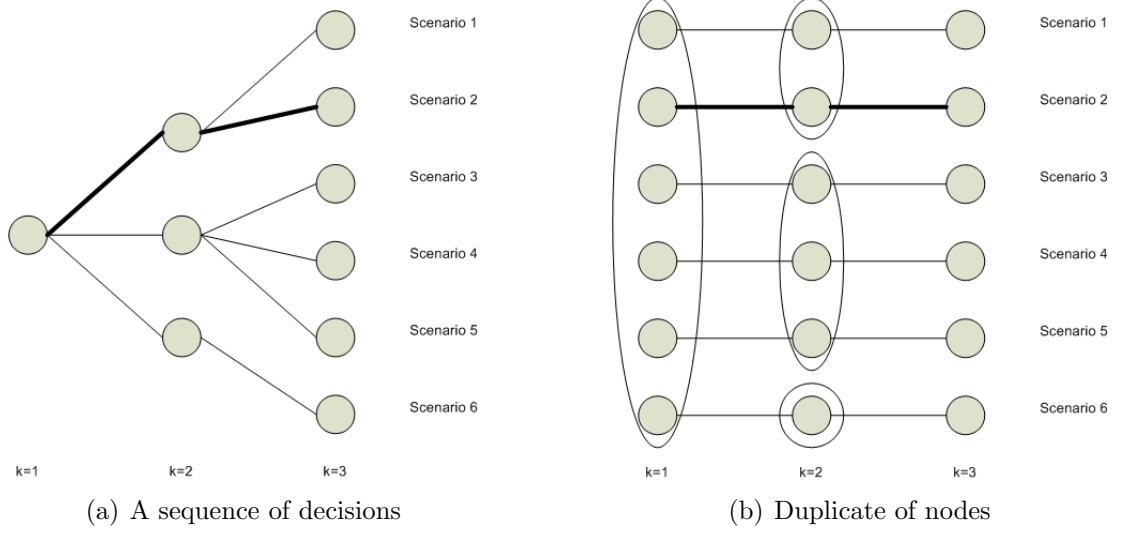
Here,  $x^1$  is the set of first-stage decisions and  $x(\xi^k)$  represent the  $k^{\text{th}}$ -stage decisions. Furthermore,  $x(\xi^k)$  can only depend on the information known up to time  $k$ , and this is often referred as nonanticipativity. In stochastic program (3), the nonanticipativity constraints are implicit in the model since each decision variable represents each node

in a tree, see Figure 3 for a three-stage example, which illustrates the idea of node-based implementation.



**Figure 3:** Node-based implementation of the nonanticipativity constraints

Another implementation of the nonanticipativity constraints is by scenarios. In a tree shown in Figure 4(a), the number of leaf nodes represents the number of scenarios, and the bold lines between nodes illustrate the second scenario, for example. A scenario here describes a series of decisions that are to be made over time. In addition, nodes are duplicated in order to match the total number of scenarios, and the idea is shown in Figure 4(b). The decisions made in the circled nodes need to be identical since information is known up to that point without future information certainties. In order to integrate this idea into the model, we need to express the nonanticipativity constraints explicitly in the stochastic program. A deterministic equivalent form of multistage stochastic program (3) is described below. Here, constraint (4) ensures that the decisions made at stage  $k$  are the same under different scenarios that have



**Figure 4:** Scenario-based implementation of the nonanticipativity constraints

the same history up to stage  $k$ .

$$\begin{aligned}
 \min \quad & \sum_{\omega \in \Omega} p_{\omega} [c_{\omega}^1 x_{\omega}^1 + c_{\omega}^2 x_{\omega}^2 + \cdots + c_{\omega}^T x_{\omega}^T] \\
 \text{s.t.} \quad & W_{\omega}^1 x_{\omega}^1 = h_{\omega}^1 \quad \forall \omega \in \Omega \\
 & T_{\omega}^1 x_{\omega}^1 + W_{\omega}^2 x_{\omega}^2 = h_{\omega}^2 \quad \forall \omega \in \Omega \\
 & \vdots \\
 & T_{\omega}^{K-1} x_{\omega}^{K-1} + W_{\omega}^K x_{\omega}^K = h_{\omega}^K \quad \forall \omega \in \Omega \\
 & x_{\omega}^k = x_{\omega'}^k, \quad \forall \omega, \omega' : \xi_{\omega}^{[1,k]} = \xi_{\omega'}^{[1,k]}, \quad (4) \\
 & \omega, \omega' \in \Omega, \quad k = 1, \dots, K \\
 & x^k(\omega) \geq 0 \quad \forall k = 1, \dots, K
 \end{aligned}$$

Or, if we define the set of scenarios that pass through node  $n$  at stage  $k$  as  $S_k(n)$ , then constraint (4) can be replaced by (5):

$$x_{\omega}^k = x_{\omega'}^k, \quad \forall S_k(x_{\omega}^k) = S_k(x_{\omega'}^k), \quad \omega, \omega' \in \Omega, \quad k = 1, \dots, K \quad (5)$$

In (5), the nonanticipativity constraint is the same while the way to specify the nodes is different. Moreover, the nonanticipativity constraints can be interpreted in several

ways (Higle 2005). One of them is to take the expectation and another is to take the average, which are described in (6) and (7), respectively.

$$x_{\omega}^k = \sum_{\omega' \in \Omega} x_{\omega'}^k \cdot p_{\omega} \quad \forall \omega \in \Omega \quad (6)$$

$$x_{\omega}^k = \sum_{\omega' \in \Omega} x_{\omega'}^k \cdot \frac{1}{\|\Omega\|} \quad \forall \omega \in \Omega \quad (7)$$

The nonanticipativity constraint plays an important role in stochastic programs. If we decompose the stochastic program into subproblems by scenarios, then the nonanticipativity constraint is the only linking constraint among subproblems. As a result, decomposition is a reasonable technique to approach the stochastic programming problems. We will discuss in details in the following section, which introduces the solution methodologies for stochastic programs.

### ***1.3 Approaches to Stochastic Programs***

In Section 1.2.1 and 1.2.2, we described the concepts of the two-stage and multistage stochastic programs. Although it seems not complicated to formulate an SP model, the difficulty of a stochastic program is its intractability in problem size. Real-life problems often involve with a large set of choices, therefore, the number of scenarios can be huge. To give an idea of how large the problem size is, we assume three possible outcomes at each stage. Then in a ten-stage model, the number of scenarios will explode to  $3^9 = 19,683$ . The amount of scenarios grow exponentially as the number of stages increase. Due to the enormous size of the problem, several solution methodologies are suggested to solve the SP problems.

#### **1.3.1 Decomposition Methods**

Van Slyke and Wets (1969) proposed an L-shaped method, whose name was derived from the shape of the constraints, to a two-stage stochastic program. They decomposed the stochastic program into a master problem and subproblems based

on Benders decomposition, or Dantzig-Wolfe decomposition to its dual. Birge (1985) extended the L-shaped method to a multistage stochastic program by successively applying Benders decomposition algorithm which was referred as nested Benders decomposition method. Birge and Louveaux (1988) suggested a multicut L-shaped algorithm that allowing several cuts to be added at a time so that the number of iterations will be reduced.

As introduced in Section 1.2.2, stochastic programs can also be decomposed into scenarios since the nonanticipativity constraints are the only constraints that link up the scenarios. Rockafellar and Wets (1991) presented a progressive hedging algorithm that decomposed a stochastic program into admissible scenarios and imposed the nonanticipativity constraints. Mulvey and Ruszczyński (1995) applied augmented Lagrangian function to the nonanticipativity constraints and solved it by an interior point algorithm. Rosa and Ruszczyński (1996) also used augmented Lagrangian method to multistage stochastic programs and decomposed the problems into either scenarios or stages. Pennanen and Kallio (2006) suggested an operator splitting method to decompose over nodes of a scenario tree.

The literature we just described are for stochastic linear programs. If we impose integrality or binary requirements on decision variables in (4), the model becomes a stochastic integer program. The difficulty will increase since the inherent convexity property in stochastic linear programs is lost. A two-stage stochastic integer program is constructed such that the objective function is composed of several integer subproblems, which are NP-hard. Several studies have focused on the stochastic integer programs.

Laporte and Louveaux (1993) proposed a branch-and-cut algorithm, or integer L-shaped, to a two-stage stochastic program with binary first-stage decisions. Carøe and Tind (1998) extended the L-shaped method to a two-stage stochastic programs with integer recourse by applying generalized Benders decomposition. Schultz, Stougie,

and van der Vlerk (1998) used Gröbner basis methods to handle the second-stage integer problems since Gröbner basis does not depend on the right-hand-side coefficient. Since the difficulty raised in stochastic integer programs is due to the discontinuity of the objective, Ahmed, Tawarmalani, and Sahinidis (2004) reformulated the two-stage problem by a special branch-and-bound strategy that led to a special structure of the value function. Carøe and Schultz (1999) decomposed the multistage stochastic integer programs by scenarios and dualized the nonanticipativity constraints by Lagrangian relaxation. The optimal value of the Lagrangian dual provides an upper bound for a maximizing problem, and a branch-and-bound procedure was then applied to achieve feasibility. Sen and Sherali (2006) suggested decomposing the two-stage stochastic mixed-integer programs so that the second-stage subproblems can be solved by branch-and-cut algorithms.

### **1.3.2 Statistical Methods**

Due to the large number of scenarios in the expected term of the objective, statistical methods are introduced to handle the stochastic programs so that the expected value term can be estimated by sampling. Norikin, Pflug, and Ruszczyński (1998) proposed a stochastic branch-and-bound method that used stochastic upper and lower estimates of the optimal objective value, and the stochastic upper and lower bounds could be derived from Monte Carlo simulation. Kleywegt, Shapiro, and Homem-de-mello (2002) studied the Monte Carlo sampling techniques to approach the stochastic integer programs. A set of random samples were selected and the expected value form was estimated by a sample average function. The problem was then solved by a deterministic optimization algorithm, and the procedure was repeated until the optimality gap becomes sufficiently small. Shapiro (2003) showed that although random sampling provided a valid statistical lower bound for the multistage stochastic programs, conditional sampling procedure would be required to obtain a consistent

lower bound. An investigation of the quality of solutions derived from sample average approximations for two-stage stochastic linear programs was discussed by Linderoth, Shapiro, and Wright (2006), and they showed that the statistical methods provided good solutions within reasonable computational time.

### **1.3.3 Stochastic Decomposition**

Another method to approach the SP is stochastic decomposition (SD), which combines the decomposition algorithms and statistical approximation methods described above. SD is first introduced by Higle and Sen (1991). They incorporated sample-based approximations with Benders decompositions for two-stage stochastic linear programs. Higle and Sen (1999) furthered their previous study by improving the approximation methods.

## ***1.4 Thesis Outline***

Although an SP model is not difficult to formulate, the intractability of large problem size makes it hard to solve such problem to optimality. In this thesis, we will utilize the stochastic programming formulations to model the traffic flow management problem in the presence of weather uncertainties. The goal is to determine the number of aircraft to send towards the sector given the weather forecast and sector capacities. By considering both flights on the ground and in the air, our stochastic programs determine optimally how and when aircraft should be sent toward the sector for the flights on the ground and how aircraft can react after the realizations of weather for flights in the air. Two-stage and three-stage models are presented.

The thesis is organized as the following. In Chapter 2, we review the literature related to the stochastic TFM problems and list studies applying stochastic programming techniques in other fields. In Chapter 3, we present a two-stage stochastic integer program to model the TFM problem. Since solving the stochastic program to optimality is intractable in realistic cases, we propose a solution methodology called



”rolling horizon method” to approach it. We describe this method and provide the computational comparisons between the optimal objective values and the solutions derived from this new method in both solution time and solution qualities. In order to justify the rolling horizon method, we then apply the Lagrangian relaxation on the nonanticipativity constraints. Subgradient method is used to find a reasonable lower bound to the original two-stage problem. In chapter 4, we extend our two-stage model to a multistage (three-stage) stochastic integer program, which increases the dynamics to the two-stage model. Finding an efficient way to solve the multistage model to near-optimality is also a challenging work. Therefore, we apply the rolling horizon method to the multistage model and the computational results are described. Lagrangian relaxation and subgradient method are also used in order to find a tight lower bound. In Chapter 5, we will conclude how our research can be applied to the real air traffic operations. Possible variations or extensions to our models will also be addressed.

## CHAPTER II

### LITERATURE REVIEW

The flow management problems in air traffic control were first raised by Odoni (1987). The author described that the congestion can occur in several places such as airport of origin, en route airways at a waypoint or a sector, and airport of destination. Various approaches can be done to deal with the congestion problems. Long-term goals such as increasing the capacity of the NAS through improved technologies can be costly and time-consuming. Medium-term solutions can be to regulate the rate of departure and arrival by imposing time-varying fees to encourage off-peak uses. As for short-term, actions proposed to alleviate the delay were to delay the departure on the ground, to regulate the aircraft flow rates on the limiting resources, en route re-routing, en route speed control, and path-stretching maneuvers. The TFM problem is stochastic and dynamic in nature because it involves with probabilistic weather forecast and the probabilistic weather forecast evolves over time, respectively.

We have described in section 1.1 that NAS is ultimately divided into sectors, which are the fundamental operation units in terms of air traffic control. Restrictions on the number of aircraft in a given sector at a given time can be affected by the ability that a controller can handle within the time frame, the geographic location, and the most important factor, the weather conditions. Therefore, we refer our problem as single-sector TFM problem.

Although TFM under uncertain weather is an important problem to be resolved, the probabilistic nature of this problem makes it hard to be addressed. Due to the essential difficulties and the enormous size of the problem, only a few studies are done. In this chapter, we review different approaches in previous studies that were related

to the TFM problem under the uncertainty of weather. Most of these studies dealt with the aircraft departures under constrained uncertain factors. Some substantial work related to TFM and SP were proposed, however, most of them simplify the real-life situations by the limiting assumptions.

We describe the literature in two categories. The first category in section 2.1 involves TFM problems. SP as well as other kinds of approaches will be discussed. Due to the small number of studies done for SP approaches in TFM, we list a few efforts that apply stochastic programs in other industries with other different uncertain factors in the second category, which is in section 2.2.

### ***2.1 Air Traffic Flow Management Problems***

When talking about how to depart the scheduled aircraft under the weather uncertainties, ground holding program (GHP) may first come to mind. GHP determines the amount of ground delay for aircraft bounding for a given capacitated airport so that these flights will land at the airport without further holding in the air while the total expected delay cost is minimized. The main concern of GHP is that, delay in the air may cause circling which increases the fuel burn. Therefore, it is much more expensive than delay on the ground, where the aircraft engine is shutting down. Some general assumptions to SAGHP are listed here.

- Airport is the only capacitated resource in the problem.
- Departure time and flight time are deterministic and known.
- Ground delay and air delay costs are known.

Our single-sector TFM problem is similar to the single-airport GHP (SAGHP) in a way that both of them involve a single capacitated resource. The constrained location for single-sector TFM is the airspace sector while for SAGHP is the airport. To our

best knowledge, SAGHP has been researched since late 80's, and a good amount of studies are done to capture the problem.

Andreatta and Romanin-Jacur (1987) were the first to use mathematical programming approaches for the SAGHP. They suggested an objective function based on the aircraft landing priority subject to probabilistic single-period airport capacities, and proposed a polynomial dynamic programming algorithm to solve the fixed landing priority case and another optimal decision policy to solve a random priority, which depends on the chronological order of landing requests, case. Terrab and Odoni (1993) addressed both deterministic and stochastic models for multi-period SAGHP. Minimum cost flow algorithm was used to solve the deterministic case and an exact dynamic program was developed based on Andreatta and Romanin-Jacur's work to approach the stochastic case.

Richetta and Odoni (1993) were the first to use SP approaches. The SAGHP was formulated as a one-stage stochastic linear program with static weather scenarios, which were not updated as time evolves. However, they reduced the problem sizes by including only a small number of weather scenarios and aggregating flights into cost classes, and simplified the problem by assuming a constant air delay cost to achieve a reasonable solution time. Kept all these assumptions up, Richetta and Odoni (1994) improved their previous study by providing a partially dynamic multistage stochastic integer programming formulation with recourse actions. By partially dynamic, it means that ground delay decisions were made at each stage as the weather forecast was the most up-to-date, but once the ground delays were assigned to aircraft, decisions were final. Rikfin (1998) further reduced the size of the problem proposed by Richetta and Odoni (1993) based on an idea that the structural information about the flights can be recovered under a first-come-first-serve discipline. Hoffman (1997) and Ball et al. (2003) proposed a stochastic integer program based on a two-state stochastic network flow model. They proved that the constraint matrix in the stochastic model

is dual network, implying the matrix is totally unimodular. As a result, the original stochastic integer program can be solved by network flow techniques or linear programming relaxation. Nevertheless, they still treat flights in an aggregate level which do not account for different cost structures for each individual flight. Mukherjee and Hansen (2007) proposed a dynamic revisable ground holding model that improved Richetta and Odoni (1994) in two ways. They assigned ground delays to individual flights and allowed for revision of the ground delay decisions based on the updated information for flights that have not yet departed. Liu, Hansen, and Mukherjee (2008) applied scenario-based approaches to Ball et al. (2003) and Mukherjee and Hansen (2007) models. Because the scenarios were assumed in previous efforts, they studied the capacity scenarios based on the historical data and found that the scenarios follow a tree structure that certain scenarios have similar capacities in the early part of tree while branch out later on.

Vranas, Bertsimas, and Odoni (1994) extended the SAGHP to a multi-airport GHP (MAGHP). MAGHP differs from SAGHP in that MAGHP takes into consideration the propagation of delays in the network of airports as aircraft perform consecutive flights. They gave a pure 0-1 integer program for the static deterministic MAGHP while taking flight cancellation and the propagation of delay into consideration, and suggested a heuristic based on the feasible solution of the linear programming relaxation to approach the problem. They furthered their previous study so that a MAGHP in a dynamic environment is handled by pure integer programming models (Vranas, Bertsimas, and Odoni (1994)). Probabilistic features of the weather were also discussed. A slightly varied version of MAGHP by Navazio and Romanin-Jacur (1998) was to consider a traffic situation that successive flight can not depart until all its preceding flights have landed, and a 0-1 integer program was proposed to model the problem.

Although GHP seems to be heavily studied and its similarity to our single-sector TFM problem, the solution methodologies described in previous works can not be directly applied to our problem due to the essential differences that GHP involves with destination airport capacities and our problem deals with airspace sector capacities. Sectors are the volume that aircraft can occupy simultaneously within a specific time frame, while airports require separation procedure that aircraft are landing in sequence. The differences make us turn our attention to the studies focused on the airspace sectors.

Landing could be the only activity in SAGHP, while operations inside a sector are many. Aircraft trajectories and flight levels can be highly affected by the echo top and precipitation during convective weather, so some re-route decisions can be made in response to different kinds of weather realization. As a result, the need for optimization tools to include en route capacities is first addressed in Helme and Lindsay (1992).

Sector capacities are addressed in some studies when the airport and airspace are taking into consideration. Helme (1992) was the first to use mathematical programming models to account for en route waypoints and airport capacities at the same time. The author suggested deterministic models for the single- and multi-destination minimum delay problems, and could be viewed as a minimum cost flow problem and multicommodity minimum cost flow problem, respectively, so that the waypoints capacities were constrained by the arcs in a space-time network. Bertsimas and Stock-Patterson (1998) presented a deterministic integer programming model to account for en route sectors and airports capacities. Bertsimas and Stock-Patterson (2000) introduced a dynamic multicommodity integer network flow model with the consideration of rerouting. The problem was solved by a Lagrangian generation algorithm, which consisted of a series of methodologies. Lagrangian relaxation was used to generate aggregate flows, which were later decomposed into a set of flight

paths for each flight using a randomized rounding heuristic. The set of paths were then used in a packing integer program whose solution generates near-optimal routes for individual flight. Alonso, Escudero, and Ortuño (2000) furthered Bertsimas and Stock-Patterson (1998) by including the uncertain factors in airport and airspace. They provided a 0-1 stochastic program with weather uncertainty modeled by scenarios, and a fix-and-relax approach, which considered iteratively the integrality of the variables, was used to solve the program to near optimal. Hoffman et al. (2007) modeled a two-stage stochastic program based on a network describing the potential route options under different weather scenarios, which were constructed by decision tree. Lulli and Odoni (2007) presented a deterministic optimization model for the European TFM. Their model could spread delays among aircraft in a way that one unit of delay to each of two flights instead of two units delay to single flight. They also argued that assigning airborne delay rather than ground delay could sometimes be cost efficient in terms of total delay cost. Bertsimas, Lulli, and Odoni (2008) improved Bertsimas and Stock-Patterson (1998) by providing the rerouting decisions without adding decision variables but only constraints. They assumed that the minimum amount of time to fly through a sector is the same for all flights. Mukherjee and Hansen (2009) proposed a stochastic model with weather scenarios that made static ground holding decisions at an early stage while the rerouting decisions were dynamically established based on updated weather information.

Although stochastic programs are good approaches for the TFM problem, we can see from the previous paragraph that mathematical programming models for the TFM problems involving sector capacities are very limited. As a result, a comprehensive stochastic program is needed to handle this problem. In Table 1, we list a comparison between our proposed stochastic program and the existing literature applying mathematical programming approaches to TFM. And we show that the proposed solution methodology in this thesis covers all aspects in terms of various flight activities under

convective weather. Note that a rerouting decision is similar to an air-holding action when single sector is involved. Therefore, rerouting and air-holding are put in the same column.

**Table 1:** Summary of the existing literature applying mathematical programming approaches to traffic flow management and the proposed methodology

	dynamic	stochastic	ground-delay	cancellation	cruise speed	air-holding/rerouting	diversion
Helme (1992)			✓			✓	
Bertsimas and Stock-Patterson (1998)			✓			✓	
Bertsimas and Stock-Patterson (2000)	✓		✓			✓	
Alonso et al. (2000)		✓	✓			✓	
Hoffman et al. (2007)		✓	✓			✓	
Lulli and Odoni (2007)			✓			✓	
Bertsimas, Lulli, and Odoni (2008)			✓			✓	
Mukherjee and Hansen (2009)	✓	✓	✓			✓	
<b>PROPOSED APPROACH</b>	✓	✓	✓	✓	✓	✓	✓

Besides, methodologies other than mathematical programming are also used to model the various TFM problems. Wang (1991) defined an alert event corresponding to the limiting resource and the time of congestion. A shortest path algorithm was then used to minimize costs. Oussedik and Delahaye (1998) proposed a time-route assignment problem with objective to minimize the sector workload and used a Genetic Algorithm-based method to simulate each aircraft. Barnier and Brisset (2004) presented a network of direct routes and vertically separated intersecting flights by allocating them to different flight levels. The issue turned into a graph coloring minimization problem, which was tackled by constraint programming. Robelin et al. (2006) studied a set of sectors where the flight paths will pass by and derived an



Eulerian model of the selected airspace. A deterministic integer programming model was proposed to control the sector aircraft count while the dynamics of the system appeared in the constraints. Other studies such as van Kemenade, van den Akker, and Kok (1996), van den Akker, van Kemendae, and Kok (1997), and Baten et al. (2005) focused on how to reroute aircraft within a sector under the weather instead of scheduling them beforehand. Hoffman et al. (2005) suggested a collaborative routing resource allocation tool that assigned NAS resources to each flight based on the available capacities, traffic classes, and priority hierarchy. In addition to modeling uncertainty by scenarios, the probabilistic property of the weather can be handled in other ways. Nilim, El Ghaoui, and Duong (2004) proposed a Markov decision process model for multiple flights by assuming that the weather evolved as a stationary Markov chain.

We have seen various efforts dedicated to the complicated TFM problems. In chapters 3 and 4, we will present our stochastic integer programs with recourse actions. They are the most comprehensive models as ground-delay, cancellation, cruise speed, air-holding, and diversions decisions are included. In addition, the sector capacities can be updated under real weather forecast in each consecutive time period.

## ***2.2 Stochastic Programming Approaches in Other Fields***

Although stochastic programming models to the air traffic management problems are not many, its wide applicability can be seen in various industries such as transportation, finance, and manufacturing. To give an idea about how different uncertainties can be incorporated with SP models, we discuss some SP applications in the following.

Santoso et al. (2005) used a two-stage stochastic program to model a supply chain network design under the uncertainties of processing/transportation costs, demands, supplies, and capacities. They proposed a solution methodology, the sample average approximation scheme with an accelerated Benders decomposition algorithm, to

approach it. Lium, Crainic, and Wallace (2009) designed a multistage SP model to approach a service network which determined the services the carrier would operate to move the forecast demand. SP can also be applied to banking industry. Castro (2009) modeled the cash management in automatic teller machines (ATMs) and in the compensation of credit card transaction. The author used stochastic integer programs to account for future customer demands, which obtained through historical data. Restaurant revenue management is an interesting topic. Bertsimas and Shioda (2003) used stochastic integer programs to maximize the revenue in a restaurant in terms of average customer waiting time and fairness. The model determined when to seat an incoming party based on customer arrivals, customer exits after service completion, and the status of customer waiting queue. Maatman et al. (2002) proposed a two-stage stochastic program to model the decisions made by farmers in response to uncertain rainfall. The first-stage decisions were for agricultural production and the second-stage accounted for consumption, storage, selling, and purchasing actions after the harvest levels realized. Ahmed, King, and Parija (2003) presented a multi-stage stochastic integer program for capacity expansion under uncertain demand and costs. A series of reformulation, heuristic, and branch and bound algorithm were used to solve the problem to global optimal. In addition, Huang and Ahmed (2009) justified the value of multistage SP approaches in a capacity planning problem.

Similar to the TFM problems, other airline-related problems can also be handled by SP approaches. Yen and Birge (2006) applied a two-stage stochastic integer program to airline crew scheduling problem under uncertain schedule disruptions. They presented a flight-pair branching algorithm to solve it. Listes and Dekker (2005) proposed a scenario aggregation-based method to solve airline fleet composition problem. Based on the stochastic nature of passenger demand, their model determined the optimal number of aircraft of each type so that it was most profitable for the airline schedule.

SP is indeed a good approach to a problem when uncertainties involve, however, finding an applicable solution methodology is a challenging work. We have described that a comprehensive stochastic program is needed to capture the possible actions the aircraft can take under inclement weather and realized that approaches such as decomposition and heuristic are used to solve the SP models, in the following chapters, we will provide stochastic integer programming models for the TFM problem and propose a series of methodologies to solve it.

## CHAPTER III

### TWO-STAGE STOCHASTIC PROGRAMMING APPROACH TO TRAFFIC FLOW MANAGEMENT PROBLEM

As described in Chapter 2, we understand the necessity of a comprehensive model for the TFM problem in the presence of inclement weather. By including the probabilistic forecast of sector capacities, in this chapter, we propose a two-stage stochastic integer programming model. As an extension of the existing studies, we include five possible actions for flights in response to the convective weather. During the weather, flights on the ground have the choices to hold on the ground till weather becomes clear, to be cancelled if the weather in the future is forecasted to be not good enough to pass through, or to change the speed of the aircraft so that the flight can avoid the weather and fly through the sector in an alternative time. Since these actions are for flights to make before the actual weather becomes known, ground-delay, cancellation, and speed-change options are the first-stage decisions for the model. Besides, the departed flights later in the air can find that the actual weather falls into one of the possible weather forecasts. As a result, these flights can choose to divert to nearby airports or to hold in the air for a short period of time depending on how good or bad the weather will be. Since these actions are for flights to make after the weather is realized, diversion and air-holding actions are the second-stage decisions for the model. In addition, costs corresponding to each of these decisions are introduced for each single flight. Then, given the flight schedule with flight time, the probabilistic sector capacity forecasts, and the cost structures corresponding to each possible action, our two-stage stochastic integer programming model determines how aircraft are sent

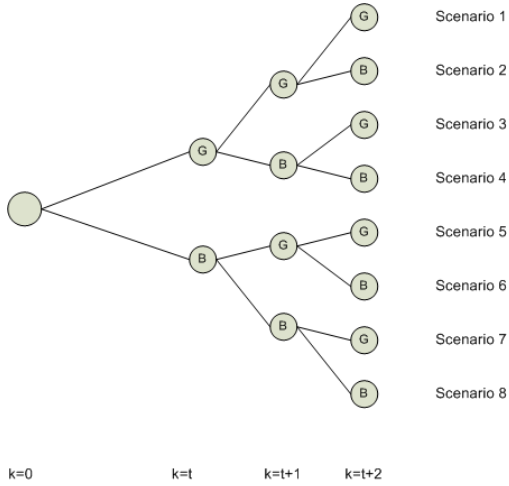
towards a sector while the overall expected costs are minimized. In Section 3.1.1, we make general assumptions for modeling the TFM. In Section 3.1.2, we present our two-stage stochastic integer programming model. We then introduce the proposed heuristic called "rolling horizon method" to approach the two-stage stochastic integer programming model in Section 3.2. Finally, we justify the rolling horizon method by the Lagrangian relaxation technique in Section 3.3. We also present the computational results after describing each method in each section.

### ***3.1 Two-Stage Stochastic Integer Programming Model for Traffic Flow Management Problem***

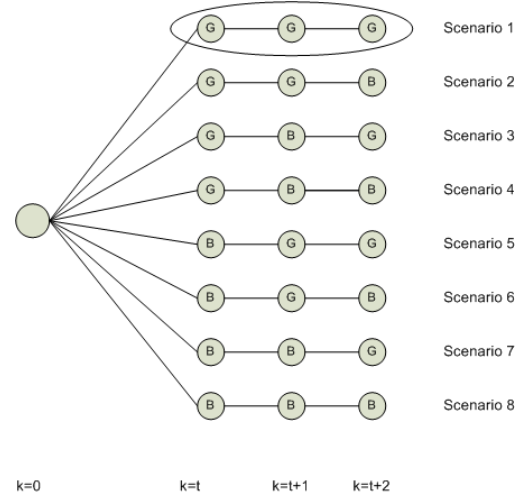
#### **3.1.1 General Assumptions**

Air traffic flow system is very complicated not only in its wide 3-dimensional range but also the time involving. Although the aircraft departures are scheduled in continuous time horizon in real life, we need to discretize the time in order to have countable number of periods. A typical implementation is 15-minute, as weather forecast is updated every 15 minutes. For analysis purposes, we also need to introduce a discrete probability distribution of sector capacities. Weather is discretized into good and bad in each time period to generate weather scenarios. Take Figure 5(a) for example, at time 0, we are about to make the decisions as how to send aircraft into the sector and the time frame we are looking at is from time  $t$  to  $t+2$ . Two weather types are assumed at each time, therefore, a total of eight scenarios are generated. If we duplicate the nodes to match the number of scenarios, in Figure 5(b), each circle contains three nodes to represent the weather situations from time  $t$  to  $t+2$ . We refer each circle as a scenario.

Flights entering a sector can spend various amounts of time in that sector. We assume that each flight entering the sector will leave the sector within one period, that is, 15 minutes. This assumption is for the ease of implementation and can easily be released without further difficulties. We also assume that the only congested area



(a) Scenario tree generated by two weather types



(b) Scenario-based representation

in the whole system is in the sector.

Departure time, cancellation, cruise speed, air-holding, and diversion decisions are included in our model. Due to aircraft capabilities, cruise speed and air-holding can be different for each flight. Airline policies may affect the ground-delay and diversion decisions under different situations. As a result, we set bounds for each of these decisions so that the model will not produce a solution set that is not implementable. Cost structures corresponding to the five actions will be discussed in details in Section 3.1.3. Here, we list a summary of general assumptions for easy reference.

- Discrete time interval of 15 minutes, described as time period or period.
- Two levels of sector capacities, either good or bad, are known.
- Single-period occupation in sector for each flight.
- Possible congestion will occur only in a given sector.
- Flight schedule is known for each flight. In addition, the bounds for possible schedule changes are known.
- Cost parameters corresponding to the five possible actions are known.

### 3.1.2 The Two-Stage Stochastic Integer Programming Model

In this section, we first define the decision variables for the stochastic integer programming model and their corresponding bound notations and cost parameters, and then present the model. Although we have five sets of actions to be determined, the ground-delay and speed-change decisions are incorporated into one set of decisions variables. Therefore, we have four sets of decision variables, each of them determines either yes or no to the corresponding questions such as "Should the flight be departed at time 1?" or "Should the pilot divert the flight to a nearby airport?" As a result, all the decision variables are binary.

$$\begin{aligned}
 x_m^{st} &= \begin{cases} 1 & \text{if flight } m \text{ is sent at time } s \text{ and arrives the sector at time } t \\ 0 & \text{otherwise} \end{cases} \\
 y_m &= \begin{cases} 1 & \text{if flight } m \text{ is cancelled} \\ 0 & \text{otherwise} \end{cases} \\
 q_m^t(\xi) &= \begin{cases} 1 & \text{if flight } m \text{ arrives the sector and diverts at time } t \text{ under weather } \xi \\ 0 & \text{otherwise} \end{cases} \\
 p_m^{tu}(\xi) &= \begin{cases} 1 & \text{if flight } m \text{ arrives the sector at time } t \text{ and enters the sector at time } u \\ & \text{under weather } \xi \\ 0 & \text{otherwise} \end{cases}
 \end{aligned}$$

Let  $M$  be the set of flights and  $m$  be the index, as well as  $S$  be the set of time periods and  $s$ ,  $t$ , and  $u$  be the indices. Let  $b_m$  be the scheduled departure period of flight  $m$  and  $\Delta k_m$  be the scheduled flight period of flight  $m$  from the origin airport to the sector. As described, several bounds and cost parameters need to be determined. For a better reference, we list the four sets of bounds and the corresponding costs into two tables, see Tables 2 and 3, respectively.

Furthermore, the sector capacity and the corresponding probability at time  $t$  under weather  $\xi$  are represented as  $C_{\xi_t}^t$  and  $p_{\xi_t}^t$ , respectively. As described in Figure 3.1.1,

**Table 2:** List of bounds on possible flight actions

Parameters	Description
$\Delta s_m$	Maximum number of periods flight $m$ can be ground-delayed
$\Delta t_m^+$	Maximum number of periods flight $m$ can be scheduled to arrive early
$\Delta t_m^-$	Maximum number of periods flight $m$ can be scheduled to arrive late
$\Delta h_m$	Maximum number of periods flight $m$ can be air-held

**Table 3:** List of cost parameters

Parameters	Description
$g_m^s$	Ground delay cost for flight $m$ if it is sent at time $s$
$can_m$	Cancellation cost for flight $m$
$c_m^{t-s-\Delta k_m}$	Speed change cost for flight $m$ if it is sent at time $s$ and arrived the sector at time $t$
$a_m^{u-t}$	Air holding cost for flight $m$ if it arrives the sector at time $t$ and enters the sector at time $u$
$d_m$	Diversion cost for flight $m$

we consider a set of consecutive periods at a time. As a result, the probability  $p_s$  for each scenario  $s$  will be calculated as  $p_s = p_{\xi_t}^t \times p_{\xi_{t+1}}^{t+1} \times \dots \times p_{\xi_{t+n}}^{t+n}$ , where  $t$  and  $n$  depend on how we look at the problem and a series of  $\xi_t$  rely on the specific weather types in scenario  $s$ .



**Two-stage stochastic program for traffic flow management problem:**

$$\begin{aligned} \min \quad & \sum_{m \in M} \left\{ \sum_{s=b_m}^{b_m+\Delta s_m} \sum_{t=s+\Delta k_m-\Delta t_m^+}^{s+\Delta k_m+\Delta t_m^-} \left[ (g_m^s + c_m^{t-s-\Delta k_m}) \cdot x_m^{st} \right] + can_m \cdot y_m \right. \\ & \left. + \sum_{t=b_m+\Delta k_m-\Delta t_m^+}^{b_m+\Delta s_m+\Delta k_m+\Delta t_m^-} \left[ d_m \cdot \mathbf{E}[\mathbf{Q}_m^t(\xi)] + \sum_{u=t}^{t+\Delta h_m} \left( a_m^{u-t} \cdot \mathbf{E}[\mathbf{P}_m^{tu}(\xi)] \right) \right] \right\} \quad (8) \end{aligned}$$

$$s.t. \quad \sum_{m \in M} \sum_t p_m^{tu}(\xi) \leq C_{\xi_u}^r \quad \forall u, \xi \quad (9)$$

$$q_m^t(\xi) + \sum_{u=t}^{t+\Delta h_m} p_m^{tu}(\xi) = \sum_s x_m^{st} \quad \forall t, m, \xi \quad (10)$$

$$y_m + \sum_{s=b_m}^{b_m+\Delta s_m} \sum_{t=s+\Delta k_m-\Delta t_m^+}^{s+\Delta k_m+\Delta t_m^-} x_m^{st} = 1 \quad \forall m \quad (11)$$

$$x_m^{st} \in \{0, 1\} \quad \forall s, t, m \quad (12)$$

$$y_m \in \{0, 1\} \quad \forall m \quad (13)$$

$$p_m^{tu}(\xi) \in \{0, 1\} \quad \forall t, u, m, \xi \quad (14)$$

$$q_m^t(\xi) \in \{0, 1\} \quad \forall t, m, \xi \quad (15)$$

The objective function (8) is to minimize the ground-delay, speed-change, and cancellation costs plus the expected air-holding and diversion costs. Constraint (9) ensures that the number of flights in the sector does not exceed sector capacity at every time period  $u$ , the second summation sums  $t$  from  $\max\{b_m + \Delta k_m - \Delta t_m^+, u - \Delta h_m\}$  to  $\min\{b_m + \Delta s_m + \Delta k_m + \Delta t_m^- + \Delta h_m, u\}$ . Constraint (10) ensures that each sent flight will either divert or enter the sector during its allowed time periods, the summation on the right-hand side sums  $s$  from  $\max\{b_m, t - \Delta k_m - \Delta t_m^-\}$  to  $\min\{b_m + \Delta s_m, t - \Delta k_m + \Delta t_m^+\}$ . Constraint (11) ensures that each flight will be sent or cancelled exactly once. Constraints (12)-(15) ensure that all variables are binary.

Before presenting the detailed computational results in 3.1.4, we need to introduce the sources of the defined parameters and the computational environment.

### 3.1.3 Experimental Setting

In order to test our model, we need to gather the related information for the defined parameters. We introduce the sources of the input data such as flight schedule, costs, and sector capacity in this section. The information we gathered will be processed based on the assumptions we made. The entire set of parameters will be used throughout this thesis unless otherwise noted.

#### 3.1.3.1 Flight Schedule

We extract the real flight schedule on June 1, 2007 from Bureau of Transportation Statistics. Delta Airline and American Airline flights with destination Atlanta are selected. Based on the assumption of discrete time interval of 15-minute, the flight schedule will be handled by period instead of exact time. Because the weather is uncertain and can not be correctly forecasted long beforehand, we include the flights that are scheduled to arrive the sector within two hours (eight periods) so that the produced decisions can be effectively used.

#### 3.1.3.2 Cost Parameters

We divide the flights into two sets by aircraft types which are narrow body and wide body. Narrow body aircraft has only one passenger aisle which includes B-737, B-757, and MD-88, while wide body aircraft has two passenger aisles which includes B-767, B-747, and MD-10.

**Ground-delay and air-holding costs:** Transport Studies Group in University of Westminster (2004) provided a complete analysis of "the cost of delay on the ground and in the air" based on the aircraft types. We take their results and apply some modifications. The generated two sets of costs are listed in Table 4 and Table 5. We can see that both costs increase nonlinearly with the length of delay time. This can be justified by the intuition that as the length of delay increases, the disconnection for passengers will increase which results the increased costs. Based on these two tables,

**Table 4:** Ground-delay costs

Delay period	Wide body	Narrow body
1	\$463	\$208
2	\$4,569	\$2,013
3	\$8,906	\$4,154
4	\$14,276	\$6,933
5	\$20,680	\$10,352
6	\$28,116	\$14,409

**Table 5:** Air-holding costs

Delay period	Wide body	Narrow body
1	\$895	\$394
2	\$5,304	\$2,396
3	\$10,424	\$4,846
4	\$16,811	\$7,977
5	\$24,466	\$11,790
6	\$33,388	\$16,284

we then assign the ground-delay and air-holding costs for each flight in accordance with the aircraft body type that serves the particular flight.

**Speed-change cost:** An analysis by Liling Ren in the Air Transportation Laboratory at Georgia Tech provided fuel-burn curves for both narrow-body and wide-body aircraft. The curves are depicted for flights at flight level 37,000 ft (FL370). When analyzing the costs, we assume zero wind and that the aircraft fly at their optimal speeds in terms of minimal fuel burn. Based on the optimal speed, we calculate the increased fuel burn when either increasing or decreasing the aircraft speed for each flight.

**Cancellation cost:** We estimate the number of passengers in each flight by multiplying the load ratio based on the Air Carrier Statistics T-100 Domestic Segment on the BTS website and the nominal number of seats for each flight. In order to evaluate the flight cancellation costs, we distinguish the passengers into two groups,

which are direct passengers and connecting passengers. The destination of each origin-destination pair is the place the connecting passengers make connections. Although no regulation is forced to accommodate passengers due to weather, airlines still make efforts to settle them. We assume that the flight cancellation compensation includes meal voucher for all passengers and hotel accommodation for connection passengers. The percentage of connecting passengers is obtained through Hartsfield-Jackson Atlanta International Airport. Besides, when a flight is cancelled at gate, airlines still need to pay some mandatory costs such as pilots, crew, and catering. We calculate these based on the Air Carrier Financial Reports schedule P-52. Compensation to the passengers and mandatory costs constitute the cancellation cost.

**Diversion cost:** According to Dr. Clarke’s conversation with American Airline, the diversion cost is approximated to be around 40,000 US dollars for both aircraft types.

#### *3.1.3.3 Bounds*

Four sets of bounds are constructed in our problem. Ground-delay bounds can be determined by several factors such as aircraft connections or safety issues. We assume a random number of periods when implementing the model. Speed-change (increase and decrease) bounds can be decided by the specifications of the aircraft such as the maximum speed an aircraft can be operated. We generate the bounds based on the real flight time and aircraft speed. The air-holding bounds can be affected by not only the aircraft capabilities but also the remaining fuel in the specific aircraft. Therefore, a set of randomly chosen air-holding bounds are used.

#### *3.1.3.4 Sector Capacity*

Sector capacity estimation is the most difficult part of the input data since weather can hardly be forecasted especially several hours into future. Since airspace is a 3-dimensional volume, tactical level reactions such as to alter aircraft trajectory or

height can be taken so that the aircraft can still fly through a sector even if a storm is within that sector. The flight route and height modifications can heavily rely upon the meteorological data. To measure different aspects of weather, several kinds of meteorological equipment are used. Over-flight can be assessed by storm height metrics such as echo tops and no-fly zones can be established by looking at turbulence metrics. An integration of data from all kinds of equipment is needed, however, it is not an easy task to generate a single number to represent the sector capacity for a period of time.

We have seen several studies trying to estimate the sector capacity in the presence of inclement weather. Hunter and Ramamoorthy (2005) used route-based approach to evaluate the reduction of sector capacity, which is proportional to the fraction of routes that are judged to be unavailable that no alternate routes can avoid the weather. Zobell, Wanke, and Song (2006) first identified a set of primary traffic flow patterns based on which sectors the flights are from and to. Probabilistic weather forecasts were then used to determine the route blockage probability for each flow type, which was then subtracted from the overall sector capacity. Based on the Corridor Integrated Weather System (CIWS), which is operated by MIT Lincoln Laboratory and provides both echo tops and precipitation data, Chen and Sridhar (2008) used Weather-Impacted Traffic Index (WITI) to count flights that is affected by weather within given areas. Martin, Evans, and DeLaura (2006) focused on ten en route sectors that were highly congested and calculated the route blockage probability using twenty convective weather days based on the algorithm developed by DeLaura and Allen (2003). Statistical methods were used to generate a model to predict the route blockage based on the weather forecast. In a recent technical interchange meeting between Georgia Tech Air Transportation Laboratory and MIT Lincoln Laboratory, we understand that Lincoln Laboratory has an undergoing project to collect CIWS and Consolidated Storm Prediction Algorithm (CoSPA) data and translate them into

a Flow Constrained Area (FCA) Capacity Forecast Matrix. The matrix provides forecasted sector capacities for every hour, and the numbers on the diagonal are the real sector capacity, see Table 6. Nevertheless, single number of sector capacity is still unavailable at the time we test our model. As a result, we randomly generate the sector capacities and the corresponding probabilities for both high and low levels of congestion.

**Table 6:** An FCA capacity forecast matrix illustration

	8	9	10	11	12	13	14	15
8	23	31	34	37	27	22	20	25
9		31	28	34	22	27	23	21
10			30	35	25	12	20	22
11				32	26	14	15	25
12					27	10	13	23
13						8	10	27
14							10	28
15								30

### 3.1.3.5 Computation Environment

All computational work described in this thesis is performed by C application using ILOG CPLEX Callable Library. CPLEX 12 is executed on RedHat Enterprise Linux 5 in 64 bit mode on a combination of three sets of systems, which include two 2.33GHz Intel Xeon E5345 CPUs with 12GB RAM, two 2.66 GHz Intel Xeon E5430 CPUs with 32GB RAM, and two 2.26 GHz Intel Xeon E5520 CPUs with 24GB RAM.

### 3.1.4 Computational Results

Based on the assumptions made in Section 3.1.1 and the input data information described in Section 3.1.3.1 through Section 3.1.3.4, a set of flight schedule is generated. We then test our model in the environment described in Section 3.1.3.5. The computational results for two-stage stochastic program of (8)-(15) are presented in Table

7.

**Table 7:** Computational results for two-stage stochastic integer programming model

Flights departure periods	Periods involved	Number of scenarios	Number of flights	Objective value	Time (sec)
1-1	4-13	1,024	22	26,230.07	4
1-2	4-13	1,024	27	30,563.07	6
1-3	4-14	2,048	28	30,563.07	15
1-4	4-16	8,192	31	30,708.07	122
1-5	4-17	16,384	35	30,715.07	304
1-6	4-17	16,384	44	30,838.07	341
1-7	4-18	32,768	52	31,015.07	3,880
1-8	4-19	65,536	60	n/a	n/a

The first column shows the periods that the flights are scheduled to depart, the second column describes the periods that the flights are entering the sector, the third column includes the number of scenarios in this specific example, and the fourth column depicts the number of flights included. Take the second row for example, "1-2" in the first column, "4-13" in the second column, "1,024" in the third column, and 27 in the fourth column indicate that in this particular instance, twenty-seven flights are included and the scheduled departure time for these flights fall into period one and two, and they are scheduled to enter the sector between time four and thirteen. The fifth and sixth columns show the optimal value and the computational time.

To present the table in words, 31 flights scheduled to arrive within a 3.25-hour time horizon case takes about 2 minutes to solve to optimal. A 44-flights case within 3.5-hour horizon takes about 6 minutes to obtain the optimal schedule. Large amounts of binary variables and the huge numbers of weather scenarios cause the long solution time. In particular, the described environment can not solve to optimality for a 60 flights, 4-hour horizon problem due to the lack of memory to store the binary tree. To understand how big the problems are, we summarize the number of columns and

rows for each instance in Table 8. Note that the number of rows does not include the binary variable constraints. The number of columns and rows basically grow proportionally as the number of scenarios increase, which are due to the length of period involved.

**Table 8:** List of numbers of columns and rows for two-stage model

Flight count	Number of scenarios	Number of columns	Number of rows
22	1,024	178,294	81,942
27	1,024	221,315	97,307
28	2,048	458,888	204,828
31	8,192	2,064,538	925,727
35	16,384	4,718,770	2,129,955
44	16,384	5,865,690	2,572,332
52	32,768	14,123,271	6,062,132
60	65,536	32,047,412	13,959,228

Figure 5 depicts the cost composition for the instance presented in Table 7 by different flight counts. We can see that in this particular example, the two-stage model tends to cancel or ground-delay flights rather than counting on the future weather uncertainties. Even if the weather turns to be bad, the flights are sent to the sector so that no flights under inclement weather will be diverted. As a result, the two-stage model leans to conservatively incur low prices in advance to avoid huge costs incurred under severe weather. Although the two-stage SIP formulation seems not too difficult, it takes minutes to solve to optimal. As the number of flights and the number of periods involved increase, the computational time increases. Therefore, we understand the importance to find an efficient way to solve the problem. In the following section, we will present a heuristic to solve the two-stage SIP model efficiently.



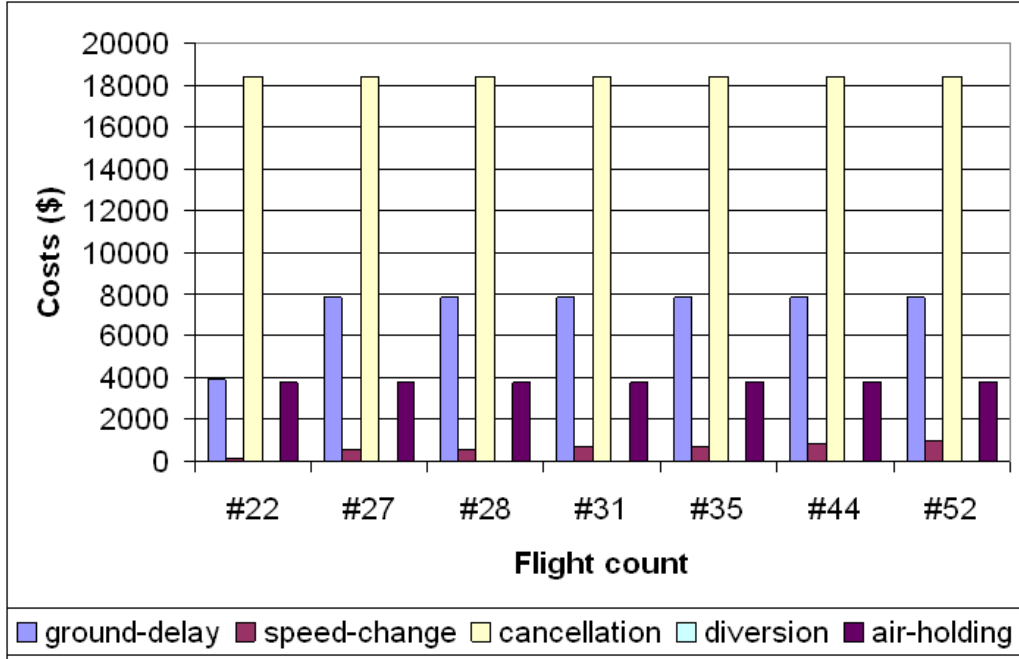
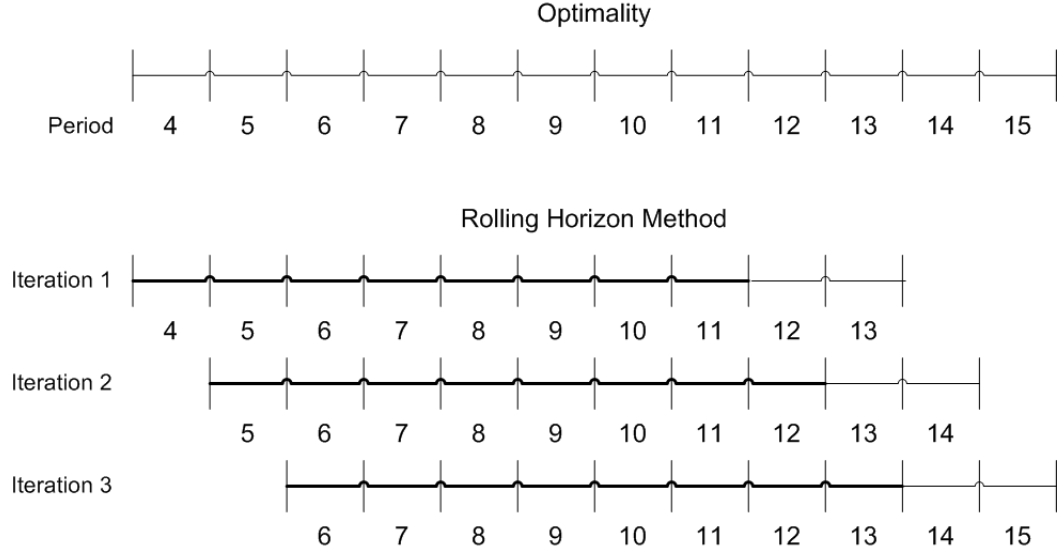


Figure 5: Cost composition for two-stage model

### 3.2 Approaches to the Two-Stage Model: The Rolling Horizon Method

In the previous section, we understand that it takes several minutes to solve the two-stage stochastic integer programming model to optimality which is not ideal for real-time decision making. After looking into the model, we realize that the main reason for the long running time is due to the large number of scenarios. If we can reduce the scenarios in some ways, we can probably efficiently improve the running time. Here, we propose a heuristic which we refer to as the "rolling horizon method," to approach the problem. The main idea of this heuristic is to reduce the number of scenarios. As an example shown in Figure 6, the optimization problem includes periods from four to fifteen which is twelve periods. A three-iteration rolling horizon method includes three subproblems, each of them includes ten periods.



**Figure 6:** An illustration of the rolling horizon method

### 3.2.1 The Rolling Horizon Method

The rolling horizon method decomposes the two-stage stochastic integer program by time and solves those subproblems in a rolling fashion. The decomposed subproblems include fewer periods than the original optimization model, as a result, the number of scenarios is reduced. Since subproblems are solved in sequence, the sector capacities need to be updated once the subproblem is solved. The ideas are as following.

Step 1.  $t = 0$

Step 2. Do until all the fights in the original problem are analyzed

2.1 Select a set of fights according to time parameter setting

2.2 Run the modified two-stage stochastic program (16)

2.3 Subtract the capacity used

2.4  $t = t + 1$ . Go to Step 2

And the modified two-stage stochastic program is described here.

$$\begin{aligned}
\min \quad & \sum_{m \in M} \left\{ \sum_{s=b_m}^{b_m+\Delta s_m} \sum_{t=s+\Delta k_m-\Delta t_m^+}^{s+\Delta k_m+\Delta t_m^-} \left[ (g_m^s + c_m^{t-s-\Delta k_m}) \cdot x_m^{st} \right] + can_m \cdot y_m \right. \\
& \left. + \sum_{t=b_m+\Delta k_m-\Delta t_m^+}^{b_m+\Delta s_m+\Delta k_m+\Delta t_m^-} \left[ d_m \cdot \mathbf{E}[\mathbf{Q}_m^t(\xi)] + \sum_{u=t}^{t+\Delta h_m} (a_m^{u-t} \cdot \mathbf{E}[\mathbf{P}_m^{tu}(\xi)]) \right] \right\} \\
s.t. \quad & \sum_{m \in M} \sum_t p_m^{tu}(\xi) \leq C_{\xi_u}^r - \sum_{m \in M} \sum_{t'} p_m^{t'u}(\xi) \quad \forall u, \xi \quad (17) \\
& q_m^t(\xi) + \sum_{u=t}^{t+\Delta h_m} p_m^{tu}(\xi) = \sum_s x_m^{st} \quad \forall t, m, \xi \\
& \sum_{s=b_m}^{b_m+\Delta s_m} \sum_{t=s+\Delta k_m-\Delta t_m^+}^{s+\Delta k_m+\Delta t_m^-} x_m^{st} = 1 \quad \forall m \\
& x_m^{st} \in \{0, 1\} \quad \forall s, t, m \\
& y_m \in \{0, 1\} \quad \forall m \\
& p_m^{tu}(\xi) \in \{0, 1\} \quad \forall t, u, m, \xi \\
& q_m^t(\xi) \in \{0, 1\} \quad \forall t, m, \xi
\end{aligned}$$

The  $t$  in (17) sums from  $\max\{b_m + \Delta k_m - \Delta t_m^+, u - \Delta h_m\}$  to  $\min\{b_m + \Delta s_m + \Delta k_m + \Delta t_m^- + \Delta h_m, u\}$ , and the  $t'$  sums the same things but in the previous iteration.

The model presented here is essentially the same as the two-stage stochastic programming model except for the first set of constraints (17). The rolling horizon method solves the two-stage model in a rolling sense, which solves the problem by dividing it into several smaller problems and solve by sequence. Therefore, the sector capacity is updated using the previous iteration's solutions, thus comes the first set of constraints. To be more precise, the rolling horizon method chooses some flights in the first iteration and solves it to optimal. Before going to the second iteration, the sector capacity is updated according to the first iteration solutions. It then chooses another set of flights with possibly repeated flights from the first iteration. The rolling horizon method keeps solving the modified two-stage stochastic program until all the

flights in the original problem are considered.

### 3.2.2 Computational Experiments

Besides the setting we already described in Section 3.1.3, we need further clarifications on the parameter setting for the rolling horizon method. In 2.1 of (16), we mentioned that we choose flights using additional criteria. We introduce two additional time parameters. The first one of "PD" is the number of periods that we mainly focus on which describes the periods that flights are scheduled to enter the sector while the second one of "fuPD" is the number of periods that we deem as buffer periods that only flights with modified schedule can enter the sector. For the ease of understanding, we describe the flight schedule in Table 9. Different sets of flights are included when different PDs are used. For example, if we set PD=9, then three iterations will be involved. The first iteration includes flight 1, 2, 3, and 4, the second includes flight 3 and 4, and finally the third includes only flight 5. If we set PD=10, then two iterations are involved. The first iteration includes flight 1, 2, 3, and 4, and the second iteration includes flight 3, 4, and 5. And finally if we set PD=11, then only one iteration is involved which includes all five flights. We keep the buffer periods defined by fuPD small in order to obtain solutions within reasonable computational time. However, a small fuPD such as two will not tolerate flight 2 to enter the sector in period thirteen if we set PD=10, while the original two-stage model will. This is the trade-off between computational time and solution quality. In the following computational results tables, we implement different PDs to include different numbers of flights in each iteration while keeping fuPD the same as two.

In addition to the PD and fuPD parameters, several important concepts during the implementation are described below. Since we solve the rolling horizon method every single period, some flights might be considered several times. Therefore, we include the latest results from the rolling horizon method for those flights that are

**Table 9:** Example of how flights are chosen in the rolling horizon method

m	$b_m$	$\Delta k_m$	$\Delta t_m^+$	$\Delta t_m^-$	$\Delta s_m$	$\Delta h_m$
1	1	6	0	1	1	1
2	1	8	0	3	1	0
3	2	4	0	1	0	0
4	2	7	0	2	1	1
5	3	8	0	1	0	0

**Table 10:** Solution format for the rolling horizon method

iteration 1	iteration 2	iteration 3
$x_1^{1,10}$		
$x_2^{1,9}$		
$x_3^{1,10}$		
$x_4^{2,6}$	$x_4^{2,6}$	
$x_5^{2,8}$	$x_5^{2,10}$	
$x_6^{2,8}$	$x_6^{3,10}$	
$x_7^{2,10}$	$x_7^{2,11}$	
	$x_8^{3,11}$	$x_8^{3,11}$
$x_9^{4,8}$	$x_9^{4,8}$	$x_9^{4,8}$
	$x_{10}^{4,11}$	$x_{10}^{4,11}$
		$x_{11}^{4,12}$

repeated considered. The solution format of the rolling horizon method is shown in Table 10 and the final departure decisions we will eventually execute are listed in Table 11.

Each of the final departure decisions has different recourse actions based on the weather realized. Suppose the solutions in the first column in Table 10 are optimized under 16 weather scenarios, then each of the solutions in the first column will have 16 possible recourse-action solutions. Take the solution  $x_2^{1,9}$  for flight two for example, the recourse-action solutions are listed in Table 12. The solutions tell us that under weather scenarios 1, 2, 3, 9, 11, and 13, flight two will arrive and enter the sector at

**Table 11:** Final solutions for the rolling horizon method

Final Departure Decisions
$x_1^{1,10}$
$x_2^{1,9}$
$x_3^{1,10}$
$x_4^{2,6}$
$x_5^{2,10}$
$x_6^{3,10}$
$x_7^{2,11}$
$x_8^{3,11}$
$x_9^{4,8}$
$x_{10}^{4,11}$
$x_{11}^{4,12}$

time 9. Under weather scenarios 4, 7, 8, 10, and 14, flight two will arrive the sector at time 9 and enter the sector at time 10. Under weather scenario 12, the flight will circle around for 2 periods and enter the sector at time 11. Finally, when weather scenarios 5, 6, 15, and 16 are realized, the flight will divert eventually.

**Table 12:** Second-stage solution format

$p_2^{9,9}(1)$	$q_2^9(5)$	$p_2^{9,9}(9)$	$p_2^{9,9}(13)$
$p_2^{9,9}(2)$	$q_2^9(6)$	$p_2^{9,10}(10)$	$p_2^{9,10}(14)$
$p_2^{9,9}(3)$	$p_2^{9,10}(7)$	$p_2^{9,9}(11)$	$q_2^9(15)$
$p_2^{9,10}(4)$	$p_2^{9,10}(8)$	$p_2^{9,11}(12)$	$q_2^9(16)$

Now, we present the computational results for comparisons between optimal objective values and the solutions obtained from the rolling horizon method in the following three tables, each of them describes the "modified objective value" from the rolling horizon method under different PDs. We describe the optimality gap in the last column.

From Table 13, 14, and 15, we can see that the rolling horizon method works very well for the described set of flights. In one second or two, we have a set of reasonable

**Table 13:** Computational results for two-stage rolling horizon method - PD=9

Flights depart in period	# of flight	Solve to Optimality			Rolling Horizon Method			
		Objective value	Time (sec)	Period	Modified objective	Time (sec)	# of iteration	% from optimal
1-1	22	26,230.07	4	4-13	26,230.07	1	1	0.00%
1-2	27	30,563.07	6	4-13	30,563.07	1	1	0.00%
1-3	28	30,563.07	15	4-14	30,628.07	1	3	0.21%
1-4	31	30,708.07	122	4-16	30,817.12	1	4	0.36%
1-5	35	30,715.07	304	4-17	30,825.11	1	5	0.36%
1-6	44	30,838.07	341	4-17	30,956.00	2	6	0.38%
1-7	52	31,015.07	3,880	4-18	31,140.08	3	6	0.40%

decisions that the overall expected costs are within 0.4 percent of optimal. Or, by spending about 30 seconds, we can have the optimal decisions. Different PDs can be chosen depending on how soon the controller needs the decisions.

### 3.2.3 Justification of the Rolling Horizon Method

In order to see if the rolling horizon method does provide good solutions to various weather forecasts, in this section, different sets of examples will be tested.

As described in Section 3.1.3 that the bounds for ground-delay and air-holding and the sector capacities with their corresponding probabilities are randomly generated data, we produce another nine sets of random parameters to cover the deficiency as well as to justify that the rolling horizon method is a good approach for the two-stage stochastic program. Note that since we use the same set of flights, the number of iterations for the rolling horizon method, which are shown in Table 13-15, do not change.

Tables 16-20 give us an idea about how the rolling horizon method works. For smaller flight counts such as 22, 27, and 28, a choice of PD=11 can give us the optimal solution in about a minute while the original two-stage stochastic program can take

**Table 14:** Computational results for two-stage rolling horizon method - PD=10

Flights depart in period	# of flight	Solve to Optimality			Rolling Horizon Method			
		Objective value	Time (sec)	Period	Modified objective	Time (sec)	# of iteration	% from optimal
1-1	22	26,230.07	4	4-13	26,230.07	2	1	0.00%
1-2	27	30,563.07	6	4-13	30,563.07	2	1	0.00%
1-3	28	30,563.07	15	4-14	30,628.07	2	2	0.21%
1-4	31	30,708.07	122	4-16	30,781.96	2	3	0.24%
1-5	35	30,715.07	304	4-17	30,789.32	2	4	0.24%
1-6	44	30,838.07	341	4-17	30,839.26	5	5	0.00%
1-7	52	31,015.07	3,880	4-18	31,020.73	8	5	0.02%

up to twenty minutes. For larger flight counts such as 31 and 35, a choice of PD=12 can provide solutions within 2 percent of the optimal in about a minute while the original stochastic program can take up to hours. We do not list the comparison for instances with larger flight counts since we could not obtain the optimal value for most of them. In conclusion, the rolling horizon method provides good results within reasonable computational time.

As we can see from Tables 7 that the computational systems are not able to produce optimal solutions for the 60-flight case, we can not judge directly how the rolling horizon method works. In order to obtain a reasonable conclusion about the presented rolling horizon method, we will next implement the Lagrangian relaxation methodology.

### ***3.3 Lagrangian Relaxation***

Lagrangian relaxation techniques usually provide high-quality bounds within a few iterations for linear programs. We introduce the Lagrangian relaxation here since we are trying to justify that the rolling horizon method gives a good estimate of the original problem. In our cost-minimizing problem, the Lagrangian relaxation is



**Table 15:** Computational results for two-stage rolling horizon method - PD=11

Flights depart in period	# of flight	Solve to Optimality			Rolling Horizon Method			
		Objective value	Time (sec)	Period	Modified objective	Time (sec)	# of iteration	% from optimal
1-1	22	26,230.07	4	4-13	26,230.07	4	1	0.00%
1-2	27	30,563.07	6	4-13	30,563.07	6	1	0.00%
1-3	28	30,563.07	15	4-14	30,563.07	6	1	0.00%
1-4	31	30,708.07	122	4-16	30,708.81	7	2	0.00%
1-5	35	30,715.07	304	4-17	30,715.29	9	3	0.00%
1-6	44	30,838.07	341	4-17	30,838.40	16	4	0.00%
1-7	52	31,015.07	3,880	4-18	31,017.81	23	4	0.01%

to produce a lower bound, which is obtained by relaxing one of the constraints in a Lagrangian fashion. A penalty corresponding to the relaxed constraint is added to the original objective function. Subgradient method is then used to find the maximum possible lower bounds.

Although we can not see exactly the nonanticipativity constraints in the original two-stage model, the constraints are actually implicitly included in the model. That is because we only introduce one first-stage decision variable which corresponds to all the second-stage scenarios. If we add additional first-stage decision variables so that each of them corresponds to each of the second-stage scenario,

$$\mathbf{x}_m^{st} \Rightarrow \mathbf{x}_m^{st}(\xi) \quad \forall \xi, \text{ and}$$

$$\mathbf{y}_m \Rightarrow \mathbf{y}_m(\xi) \quad \forall \xi,$$

then we need to add the explicit nonanticipativity constraints of

$$\mathbf{x}_m^{st}(1) = \mathbf{x}_m^{st}(\xi) \quad \forall \xi \neq 1, \text{ and} \tag{18}$$

$$\mathbf{y}_m(1) = \mathbf{y}_m(\xi) \quad \forall \xi \neq 1, \tag{19}$$

to the original model.

**Table 16:** Computational results for two-stage rolling horizon method - flight count=22

Data set	Periods involved	Optimality		RHM, PD=10			RHM, PD=11		
		Objective value	Time (sec)	Modified objective	Time (sec)	% from optimal	Modified objective	Time (sec)	% from optimal
1	4-13	26,230.07	4	26,230.07	2	0.00%	26,230.07	4	0.00%
2	4-13	2,443.76	12	2,443.76	5	0.00%	2,443.76	14	0.00%
3	4-13	888.10	23	888.10	4	0.00%	888.10	8	0.00%
4	4-13	616.00	10	616.00	3	0.00%	616.00	9	0.00%
5	4-13	32,572.05	22	32,572.05	5	0.00%	32,572.05	22	0.00%
6	4-13	33,348.96	20	33,348.96	6	0.00%	33,348.96	20	0.00%
7	4-14	119.00	11	119.00	2	0.00%	119.00	5	0.00%
8	4-13	196.00	5	196.00	2	0.00%	196.00	5	0.00%
9	4-13	13,092.58	20	13,092.58	3	0.00%	13,092.58	8	0.00%
10	4-13	29,069.94	31	29,069.94	3	0.00%	29,069.94	12	0.00%

**Table 17:** Computational results for two-stage rolling horizon method - flight count=27

Data set	Periods involved	Optimality		RHM, PD=10			RHM, PD=11		
		Objective value	Time (sec)	Modified objective	Time (sec)	% from optimal	Modified objective	Time (sec)	% from optimal
1	4-13	30,563.07	6	30,563.07	2	0.00%	30,563.07	6	0.00%
2	4-13	16,488.83	21	16,488.83	7	0.00%	16,488.83	29	0.00%
3	4-13	6,933.34	42	6,933.34	6	0.00%	6,933.34	16	0.00%
4	4-13	1,758.41	12	1,758.41	4	0.00%	1,758.41	12	0.00%
5	4-13	49,302.51	69	49,302.51	9	0.00%	49,302.51	64	0.00%
6	4-13	50,698.96	22	50,698.96	7	0.00%	50,698.96	21	0.00%
7	4-14	279.00	13	279.00	2	0.00%	279.00	5	0.00%
8	4-13	1,576.88	11	1,576.88	4	0.00%	1,576.88	11	0.00%
9	4-13	25,327.99	41	25,765.21	4	1.73%	25,327.99	16	0.00%
10	4-13	42,968.46	38	42,968.46	5	0.70%	42,968.46	12	0.00%

**Table 18:** Computational results for two-stage rolling horizon method - flight count=28

Data set	Periods involved	Optimality			RHM, PD=10			RHM, PD=11		
		Objective value	Time (sec)	Modified objective	Time (sec)	% from optimal	Modified objective	Time (sec)	% from optimal	
1	4-14	30,563.07	15	30,628.07	3	0.21%	30,563.07	6	0.00%	
2	4-15	16,544.83	243	16,544.83	7	0.00%	16,544.83	30	0.00%	
3	4-14	6,989.34	109	6,989.45	6	0.00%	6,989.34	19	0.00%	
4	4-14	1,758.54	41	1,759.51	5	0.05%	1,758.54	14	0.00%	
5	4-14	49,358.51	450	49,415.07	9	0.11%	49,358.51	64	0.00%	
6	4-15	50,754.96	1,160	50,813.83	7	0.12%	50,754.96	28	0.00%	
7	4-15	286.00	38	286.00	2	0.00%	286.00	6	0.00%	
8	4-14	1,583.88	29	1,625.18	3	2.61%	1,583.88	11	0.00%	
9	4-15	25,383.99	83	25,825.85	5	1.74%	25,383.99	16	0.00%	
10	4-14	42,968.46	144	42,968.46	5	0.00%	42,968.46	13	0.00%	

**Table 19:** Computational results for two-stage rolling horizon method - flight count=31

Data set	Periods involved	Optimality			RHM, PD=11			RHM, PD=12		
		Objective value	Time (sec)	Modified objective	Time (sec)	% from optimal	Modified objective	Time (sec)	% from optimal	
1	4-16	30,708.07	122	30,781.96	2	0.24%	30,708.81	7	0.00%	
2	4-16	19,348.91	2,573	19,362.19	10	0.07%	19,348.91	31	0.00%	
3	4-16	9,253.01	606	9,253.36	6	0.00%	9,253.01	27	0.00%	
4	4-17	2,194.11	1,273	2,194.11	5	0.00%	2,194.11	17	0.00%	
5	4-17	53,692.51	43,922	54,457.78	12	1.43%	54,103.74	67	0.77%	
6	4-16	54,966.35	1,817	55,542.07	7	1.05%	54,973.24	33	0.01%	
7	4-16	372.00	121	397.26	2	6.79%	372.00	7	0.00%	
8	4-17	2,214.75	7,398	2,280.79	4	2.98%	2,228.95	13	0.64%	
9	4-17	29,254.99	349	29,986.85	5	2.50%	29,254.99	18	0.00%	
10	4-17	46,287.86	1,829	47,057.33	6	1.66%	46,288.67	18	0.00%	

**Table 20:** Computational results for two-stage rolling horizon method - flight count=35

Data set	Periods involved	Optimality			RHM, PD=11			RHM, PD=12		
		Objective value	Time (sec)	Modified objective	Time (sec)	% from optimal	Modified objective	Time (sec)	% from optimal	
1	4-16	30,715.07	304	30,789.32	2	0.24%	30,715.29	9	0.00%	
2	4-16	19,571.91	4,089	19,586.36	10	0.07%	19,573.10	37	0.01%	
3	4-16	9,378.01	2,891	9,378.27	7	0.00%	9,378.01	24	0.00%	
4	4-17	2,206.11	2,595	2,206.11	6	0.00%	2,206.11	21	0.00%	
5	4-17	54,463.12	46,912	56,733.85	12	4.17%	54,871.73	82	0.75%	
6	4-16	n/a	n/a	58,778.13	7	n/a	55,412.83	38	n/a	
7	4-16	384.00	266	416.26	3	8.40%	391.00	10	1.82%	
8	4-17	2,347.75	1,544	2,417.43	5	2.97%	2,375.97	16	1.20%	
9	4-17	29,428.99	1,116	30,199.05	6	2.62%	29,557.81	22	0.44%	
10	4-17	46,296.86	2,391	47,056.77	6	1.64%	46,296.86	26	0.00%	

The nonanticipativity constraints of (18) and (19) are the only constraints that link among scenarios. We can see this fact by ignoring the nonanticipativity constraints, and find out that the problem can be decomposed by scenarios. As a result, the nonanticipativity constraints are the constraints that we are going to relax. Lagrange multipliers of  $\lambda_m^{st}(\xi), \forall \xi \neq 1$  are introduced for not satisfying constraint (18) and  $\theta_m(\xi), \forall \xi \neq 1$  are placed for not satisfying constraint (19). Therefore,

$$\lambda_m^{st}(\xi) [\mathbf{x}_m^{st}(1) - \mathbf{x}_m^{st}(\xi)] \quad \forall m, s, t, \xi \neq 1, \text{ and} \quad (20)$$

$$\theta_m(\xi) [\mathbf{y}_m(1) - \mathbf{y}_m(\xi)] \quad \forall m, \xi \neq 1 \quad (21)$$

are added to the objective of the original two-stage model. We then form the Lagrangian relaxation problem as follows.

$$\begin{aligned} \min \quad & \sum_m \left\{ \sum_s \sum_t \left[ (g+c) \cdot \mathbf{E}[\mathbf{X}_m^{st}(\xi)] \right] + can \cdot \mathbf{E}[\mathbf{Y}_m(\xi)] \right. \\ & \left. + \sum_t \left[ d \cdot \mathbf{E}[\mathbf{Q}_m^t(\xi)] + \sum_u \left( a \cdot \mathbf{E}[\mathbf{P}_m^{tu}(\xi)] \right) \right] \right\} \\ & + \sum_m \sum_s \sum_t \sum_{\xi \neq 1} \lambda_m^{st}(\xi) [\mathbf{x}_m^{st}(1) - \mathbf{x}_m^{st}(\xi)] + \sum_m \sum_{\xi \neq 1} \theta_m(\xi) [\mathbf{y}_m(1) - \mathbf{y}_m(\xi)] \end{aligned} \quad (22)$$

$$s.t. \quad \sum_m \sum_t \mathbf{P}_m^{tu}(\xi) \leq C_{\xi_u}^{tu} \quad \forall u, \xi \quad (23)$$

$$\mathbf{q}_m^t(\xi) + \sum_u \mathbf{P}_m^{tu}(\xi) = \sum_s \mathbf{x}_m^{st}(\xi) \quad \forall t, m, \xi \quad (24)$$

$$\mathbf{y}_m(\xi) + \sum_s \sum_t \mathbf{x}_m^{st}(\xi) = 1 \quad \forall m, \xi \quad (25)$$

$$\mathbf{x}_m^{st}(\xi) \in \{0, 1\} \quad \forall s, t, m, \xi \quad (26)$$

$$\mathbf{y}_m(\xi) \in \{0, 1\} \quad \forall m, \xi \quad (27)$$

$$\mathbf{P}_m^{tu}(\xi) \in \{0, 1\} \quad \forall t, u, m, \xi \quad (28)$$

$$\mathbf{q}_m^t(\xi) \in \{0, 1\} \quad \forall t, m, \xi \quad (29)$$

Let  $L(x, y, p, q)$  be the objective of the above described Lagrangian relaxation problem, then the problem can be decomposed by scenarios and the objective function

can be presented as

$$L(x, y, p, q) = \sum_{\xi \in \Xi} L_{\xi}(x(\xi), y(\xi), p(\xi), q(\xi)), \quad (30)$$

where  $L_1(x(1), y(1), p(1), q(1))$

$$\begin{aligned} &= \sum_m \left\{ \sum_s \sum_t \left[ (g+c) \cdot p_1 + \sum_{\xi \neq 1} \lambda_m^{st}(\xi) \right] \cdot \mathbf{x}_m^{st}(1) + \left[ can \cdot p_1 + \sum_{\xi \neq 1} \theta_m(\xi) \right] \cdot \mathbf{y}_m(1) \right. \\ &+ \left. \sum_t \left[ d \cdot \mathbf{q}_m^t(1) + \sum_u a \cdot \mathbf{p}_m^{tu}(1) \right] \cdot p_1 \right\}, \text{ and} \\ &L_{\xi}(x(\xi), y(\xi), p(\xi), q(\xi)) \\ &= \sum_m \left\{ \sum_s \sum_t \left[ (g+c) \cdot p_{\xi} - \lambda_m^{st}(\xi) \right] \cdot \mathbf{x}_m^{st}(\xi) + \left[ can \cdot p_{\xi} - \theta_m(\xi) \right] \cdot \mathbf{y}_m(\xi) \right. \\ &+ \left. \sum_t \left[ d \cdot \mathbf{q}_m^t(\xi) + \sum_u a \cdot \mathbf{p}_m^{tu}(\xi) \right] \cdot p_{\xi} \right\}, \quad \forall \xi \neq 1. \end{aligned}$$

The problem (22)-(29) provides a lower bound for the original two-stage stochastic program. Let the optimal value of problem (22)-(29) be  $Z^2(LR)$ , then we introduce the Lagrangian dual problem as

$$\max Z^2(LR) \quad (31)$$

in order to obtain a tighter lower bound. Here, since the nonanticipativity constraints we relaxed are equalities, the corresponding Lagrange multipliers are unrestricted in sign. As a result, problem (31) subjects to nothing.

### 3.3.1 Computational Results

Problem (31) is a nondifferentiable optimization issue which can be solved by the subgradient method (Fisher (1981)). It solves problem (31) sequentially by updating several parameters. Given initial values of defined  $\lambda^0(\xi)$  and  $\theta^0(\xi)$ ,  $\xi \neq 1$ , sets of  $\lambda^k(\xi)$  and  $\theta^k(\xi)$ ,  $\xi \neq 1$  can be updated by

$$\lambda^{k+1}(\xi) = \lambda^k(\xi) + \alpha^k \cdot [\mathbf{x}^k(1) - \mathbf{x}^k(\xi)] \quad \forall \xi \neq 1, \text{ and} \quad (32)$$

$$\theta^{k+1}(\xi) = \theta^k(\xi) + \beta^k \cdot [\mathbf{y}^k(1) - \mathbf{y}^k(\xi)] \quad \forall \xi \neq 1, \quad (33)$$



respectively.  $\alpha^k$  and  $\beta^k$  are called the stepsizes at iteration  $k$ , while  $\mathbf{x}^k(\xi)$  and  $\mathbf{y}^k(\xi)$  are the optimal solutions for the decomposed-by-scenario problems of (22)-(29). The initial values of  $\lambda^0$  and  $\theta^0$  can be chosen randomly, here, we set them to zeros.

Choice of stepsizes is important so that a good lower bound can be reached within reasonable computational time. Rules to update the stepsizes include

$$\alpha^k = \frac{1}{k}, \quad (34)$$

$$\alpha^k = \alpha^0 \cdot \rho^k, \quad 0 < \rho < 1, \text{ and} \quad (35)$$

$$\alpha^k = \frac{\hat{Z}(LR) - Z(LR)^k}{\|s^k\|^2} \rho^k, \quad (36)$$

where  $0 < \rho < 1$ ,  $s$  is the subgradient, and  $\hat{Z}(LR)$  is an estimate of the optimal value of problem (31). Although these rules are common in the existing studies, we find that the subgradient method with these rules does not provide good bounds for our two-stage model. As a result, alternative stepsize rules of

$$\alpha^0 \in \mathbb{R}^+, \quad \alpha^{k+1} = \frac{\alpha^k}{\kappa} \quad \text{and} \quad (37)$$

$$\beta^0 \in \mathbb{R}^+, \quad \beta^{k+1} = \frac{\beta^k}{\nu}, \quad (38)$$

are used, where  $\kappa$  and  $\nu$  are small numbers to make  $\{\alpha^k\}$  and  $\{\beta^k\}$  converge, respectively. Note that big numbers of  $\kappa$  and  $\nu$  can cause the series of  $\{\alpha^k\}$  and  $\{\beta^k\}$ , respectively, to converge quickly but to a bad lower bound. However, it is not an easy task to find a reasonable combination of  $\alpha^0$ ,  $\beta^0$ ,  $\kappa$  and  $\nu$ . Several sets of parameters need to be tested by trial-and-error.

We first present the comparison among different sets of parameters used in the subgradient method for the twenty-two-flight case. Table 21 lists the parameter values and their corresponding best lower bound produced by running for 2,000 iterations. The paths of the objective calculations of two selected sets of parameters are depicted in Figure 7. Each point on the plot represents a lower bound calculated by the subgradient method. The points in the  $\alpha = \beta = 0.06$  curve do not zigzag very much

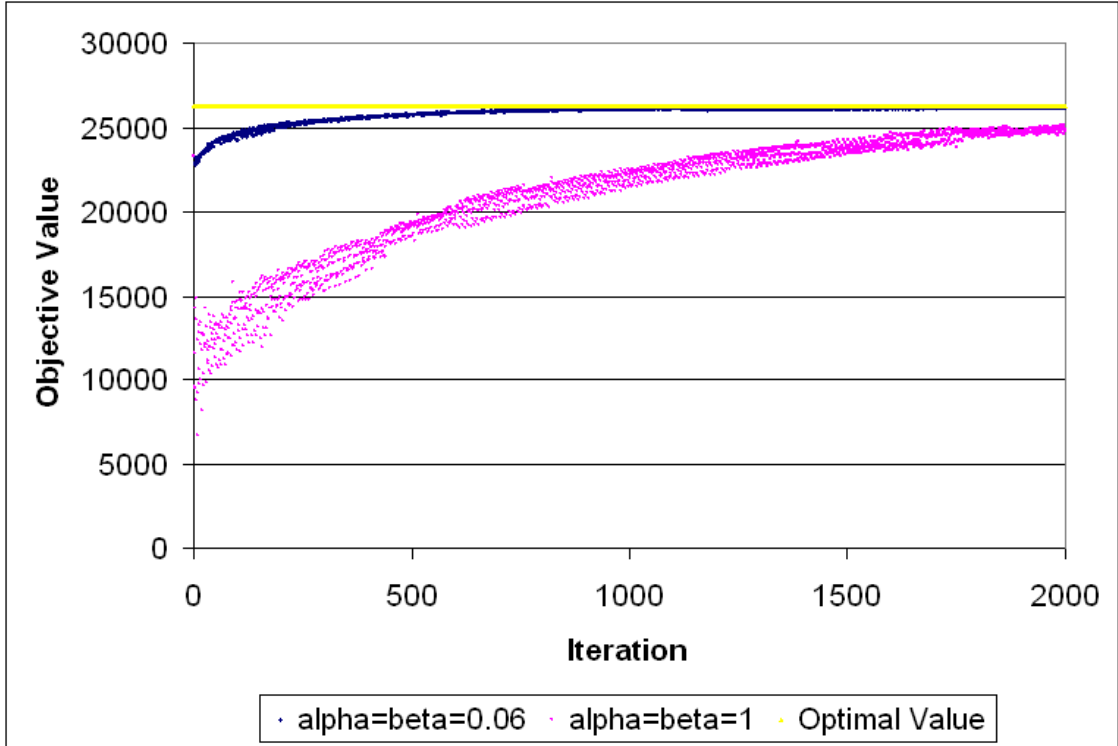
compared to the  $\alpha = \beta = 1$  curve. As a result, the  $\alpha = \beta = 0.06$  curve converges to the optimal value quicker than the  $\alpha = \beta = 1$  curve. By both Table 21 and Figure 7, we can see that the choice of the stepsizes plays an important role in the subgradient method.

**Table 21:** List of different parameters in subgradient method for two-stage model, flight count=22

$\alpha^0$	$\kappa$	$\beta^0$	$\nu$	lower bound
0.01	1.001	0.01	1.001	25,383.74
0.05	1.001	0.05	1.001	26,180.40
0.10	1.001	0.10	1.001	26,206.51
0.15	1.001	0.15	1.001	26,186.64
0.20	1.001	0.20	1.001	26,160.95
0.30	1.001	0.30	1.001	26,089.61
0.40	1.001	0.40	1.001	25,873.78
0.50	1.001	0.50	1.001	25,817.04
1.00	1.001	1.00	1.001	25,122.88

The subgradient method implementations with stepsize rules (37) and (38) running for 6,000 iterations together with the optimization model and the rolling horizon method are presented in Table 22 for a complete comparison. RHM stands for the rolling horizon method and LR represents the Lagrangian relaxation method using subgradient method implementation. Note that the gap presented in the last column is calculated by the ratio of  $\frac{OPT-LR}{OPT} \times 100\%$  if the optimal value exists and by the ratio of  $\frac{RHM-LR}{RHM} \times 100\%$  if the optimal value can not be obtained. We can see that the Lagrangian relaxation with subgradient method implementation does provide good lower bounds for the original two-stage problem.

The subgradient method produces good lower bounds for each case, however, the time to obtain the lower bound can heavily depend on the number of iterations we run which affects the solution quality a lot. To give an idea about the time we



**Figure 7:** Comparison of paths between two parameters for two-stage model, flight count=22

spent in order to obtain the lower bound shown in Table 22, we list the detailed computational time in Table 23. In addition to the long computational time to obtain lower bounds, we need to set different sets of parameters in order to avoid early convergence and obtain a reasonable bound which has been discussed earlier. In conclusion, subgradient method provides good lower bounds, but the computational work is cumbersome.

It is worth noting that, at the very beginning, we implement the subgradient method by writing the subproblem formulation to a .LP file. CPLEX reads the file and solves it. But the computational time for doing so is lengthy. Later, we construct the coefficient matrix and store it in the memory so that only right-hand-side of constraint (23) and the objective coefficient need to be updated in different scenarios  $\xi$ . Indeed, this technique improves the computational time by eight times which makes the subgradient method implementation possible.

**Table 22:** Computational comparison among optimality, rolling horizon method, and Lagrangian relaxation for two-stage model

flight count	OPT value	RHM (UB)	LR (LB)	% from optimal/UB
22	26,230.07	26,230.07	26,229.85	0.00%
27	30,563.07	30,563.07	30,562.60	0.00%
28	30,563.07	30,563.07	30,562.25	0.00%
31	30,708.07	30,708.81	30,706.19	0.01%
35	30,715.07	30,715.29	30,713.26	0.01%
44	30,838.07	30,838.40	30,836.32	0.01%
52	31,015.07	31,017.81	31,011.56	0.01%
60	n/a	32,138.21	31,997.71	0.44%

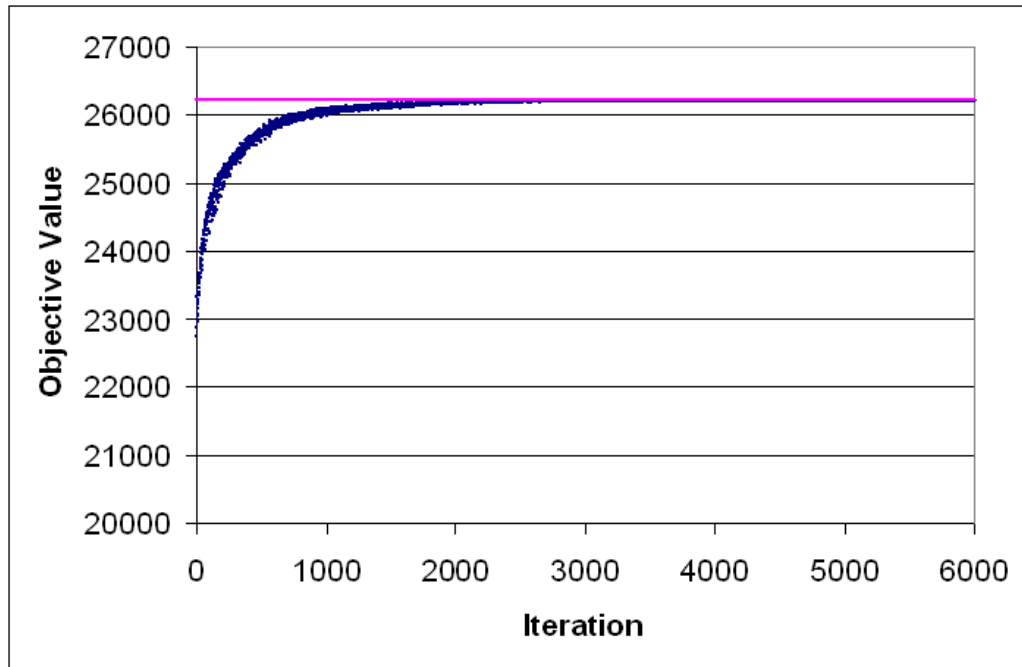
Among different sets of parameters we tested, one of them produces the best lower bound which is listed in Table 24. In accordance with these parameters, we plot the objective calculation paths for different flight counts in Figure 8-15. Note that except for sixty-flight case that the horizontal line is depicted by the objective produced by the rolling horizon method, other cases are by the optimal objective value. We can see from these figures that although we run the subgradient method for 6,000 iterations, the method can actually produce a reasonable lower bound in about 2,000 iterations. Therefore, to implement the subgradient method, we need to leverage between the computational quality and computational time.

**Table 23:** Computational time of the subgradient method for two-stage model

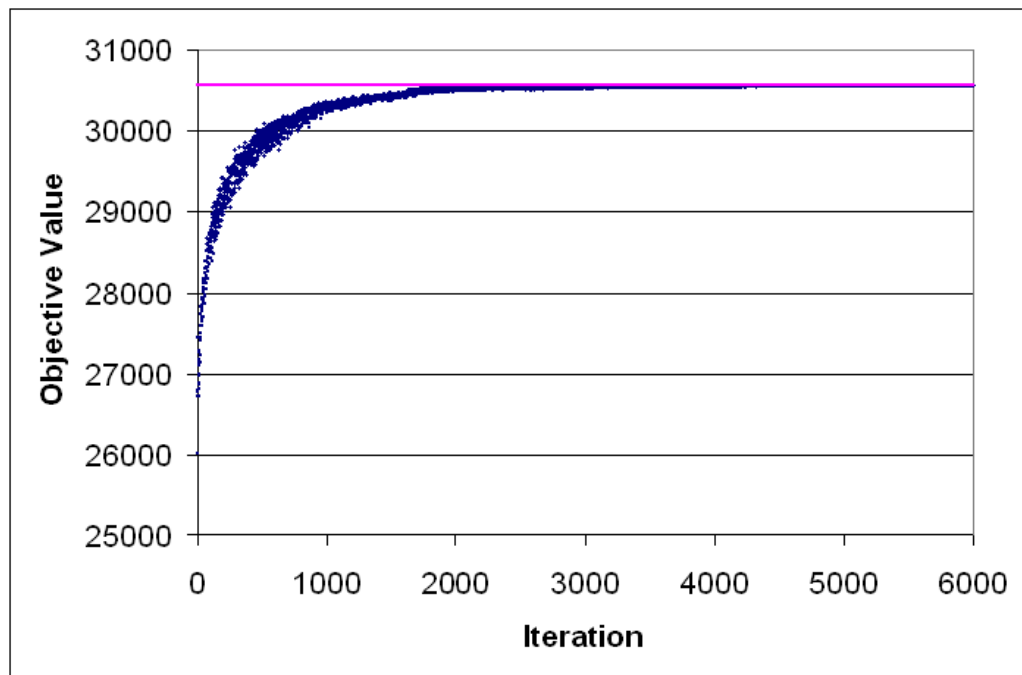
flight count	number of scenario	# of iteration	total time (sec)	second/iteration	second /scenario
22	1,024	6,000	8,444	1.41	0.001374
27	1,024	6,000	9,915	1.65	0.001614
28	2,048	6,000	21,507	3.58	0.001750
31	8,192	6,000	90,418	15.07	0.001840
35	16,384	6,000	198,809	33.13	0.002022
44	16,384	6,000	277,844	46.31	0.002826
52	32,768	6,000	605,600	100.93	0.003080
60	65,536	6,000	1,433,090	238.85	0.003645

**Table 24:** List of parameters in subgradient method producing the best lower bounds for two-stage model

flight count	$\alpha^0$	$\kappa$	$\beta^0$	$\nu$
22	0.06	1.001	0.06	1.001
27	0.08	1.001	0.08	1.001
28	0.08	1.001	0.08	1.001
31	0.03	1.001	0.03	1.001
35	0.02	1.001	0.20	1.001
44	0.015	1.001	0.30	1.001
52	0.01	1.001	0.40	1.001
60	0.015	1.001	0.015	1.001



**Figure 8:** Path of subgradient method for two-stage model, flight count=22



**Figure 9:** Path of subgradient method for two-stage model, flight count=27

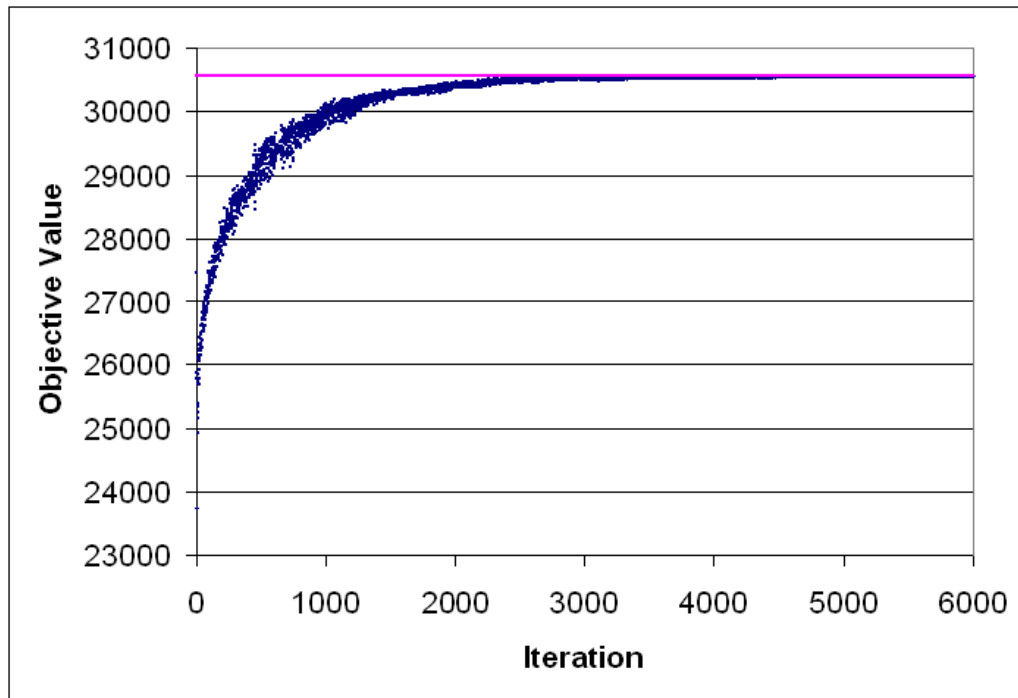


Figure 10: Path of subgradient method for two-stage model, flight count=28

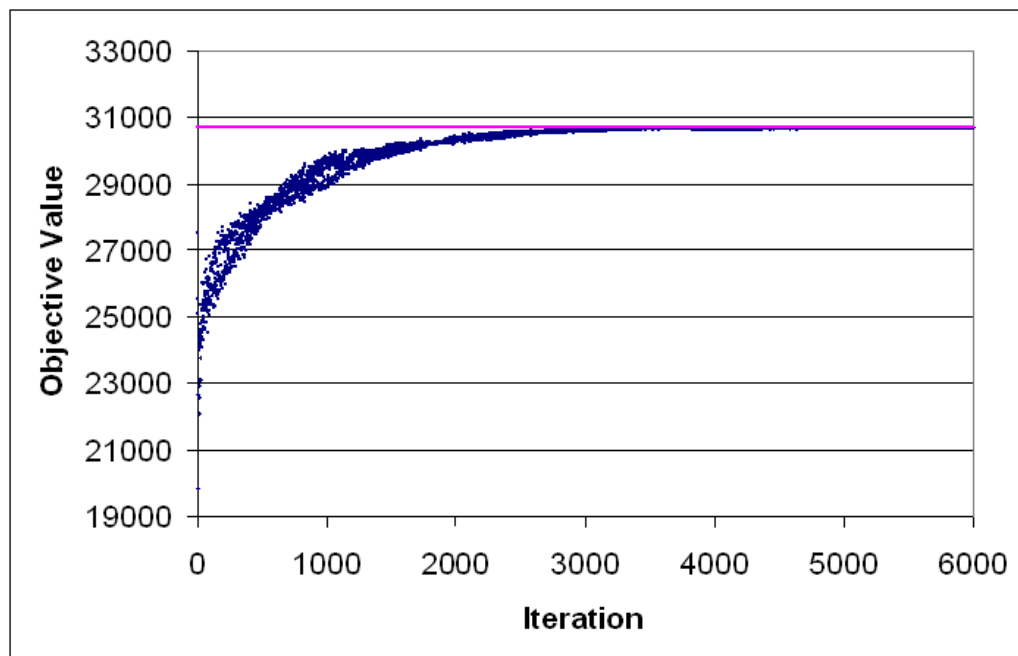
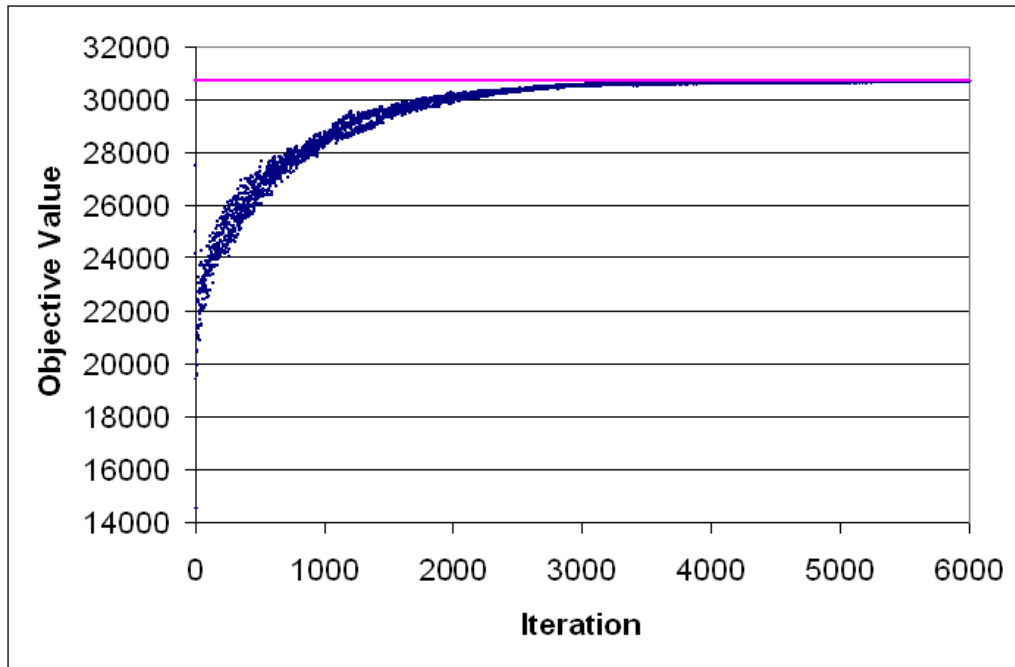
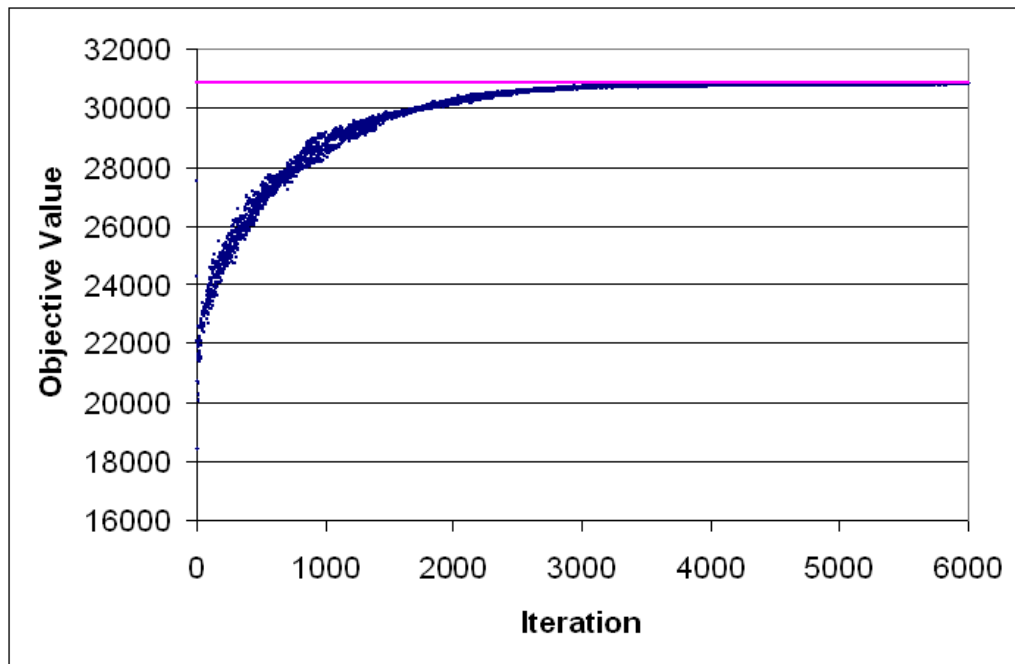


Figure 11: Path of subgradient method for two-stage model, flight count=31



**Figure 12:** Path of subgradient method for two-stage model, flight count=35



**Figure 13:** Path of subgradient method for two-stage model, flight count=44



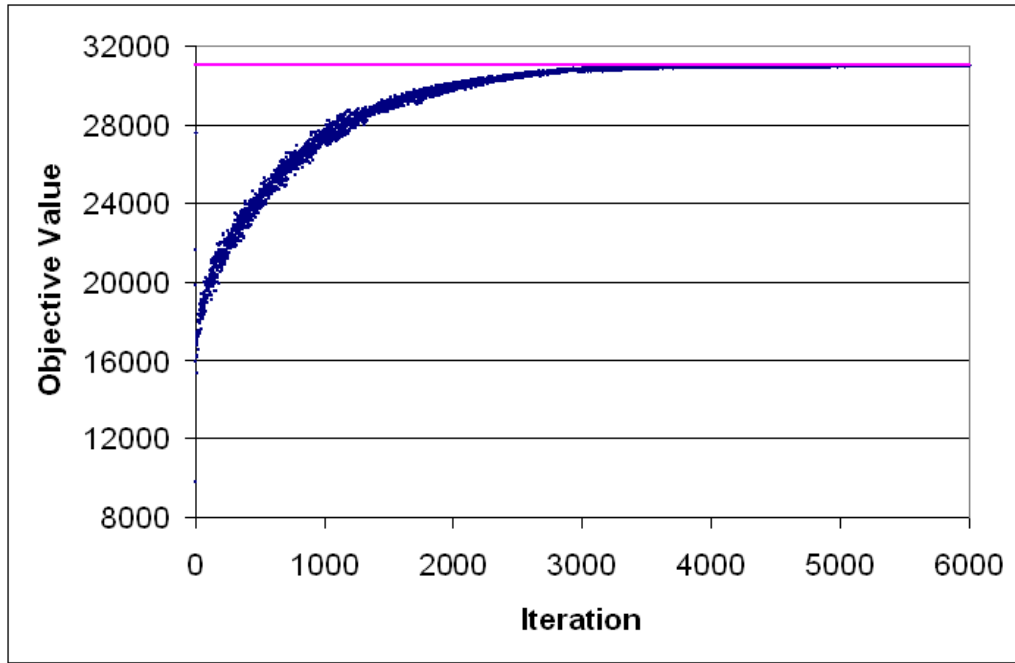


Figure 14: Path of subgradient method for two-stage model, flight count=52

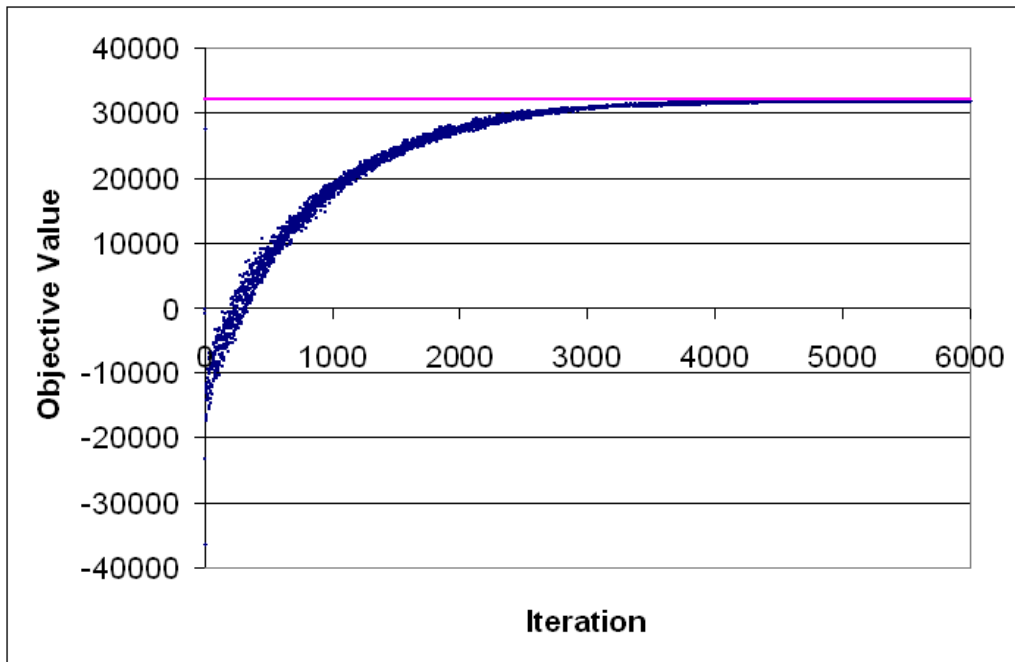


Figure 15: Path of subgradient method for two-stage model, flight count=60

# CHAPTER IV

## MULTISTAGE STOCHASTIC PROGRAMMING

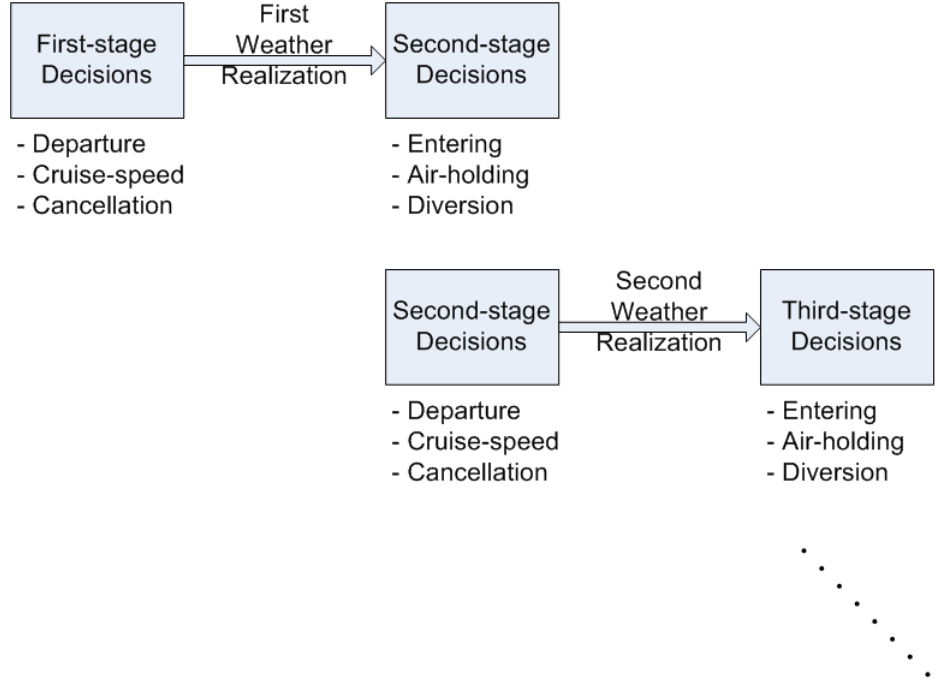
### APPROACH TO

## TRAFFIC FLOW MANAGEMENT PROBLEM

We have introduced the two-stage stochastic programming approach in previous chapter. The next idea is to extend the two-stage model to a multistage stochastic program. As described in Section 1.2.2, multistage stochastic programs can further the two-stage models by introducing additional decision points. These additional decision points may benefit the whole decision process by providing more accurate forecast for the future and give better solutions in terms of overall expected costs. In this chapter, we will explain how multistage stochastic models can be incorporated into the traffic flow management problem.

The problem description follows Chapter 3, that is, we want to decide how aircraft are sent under convective weather. Airspace sector is the only constrained place. Five possible flight actions are included in the problem and weather is modeled by different scenarios. For flights on the ground, decisions such as when the flights will be departing, what their cruise speeds are, or if we should cancel the flights are made. After the weather becomes known and one of the weather scenarios is realized, for aircraft already in the air, they can be accepted into the sector, be held in the air for a short period of time, or be diverted to nearby airports. These decisions are made consecutively in a multistage TFM problem, which is illustrated in Figure 16.

A multistage problem is different from a two-stage problem solving by rolling horizon method. Suppose we have  $n$  periods in a problem. A two-stage model distinguishes the  $n$  periods so that decisions made before and after the realized uncertain



**Figure 16:** An illustration of multistage problem in traffic flow management

factors are separated into two different stages. The rolling horizon method solves  $n$  periods in sequence based on the same realization of the uncertainty. However, a multistage problem defines the  $n$  periods into several stages while several realizations of the uncertain factors are taking place. A stochastic program including all these realizations is solved at a time. Therefore, a multistage model is bigger in size and harder to solve, compared with the two-stage stochastic program.

The idea of a multistage model is described, however, in order to implement a multistage stochastic program, we need to set up a number for the number of stages. In the following, we will introduce a three-stage stochastic program for the traffic flow management problem. Further implementations of four-stage, five-stage, and so on are possible, but the problems might be too big resulting unreasonable long computational time.

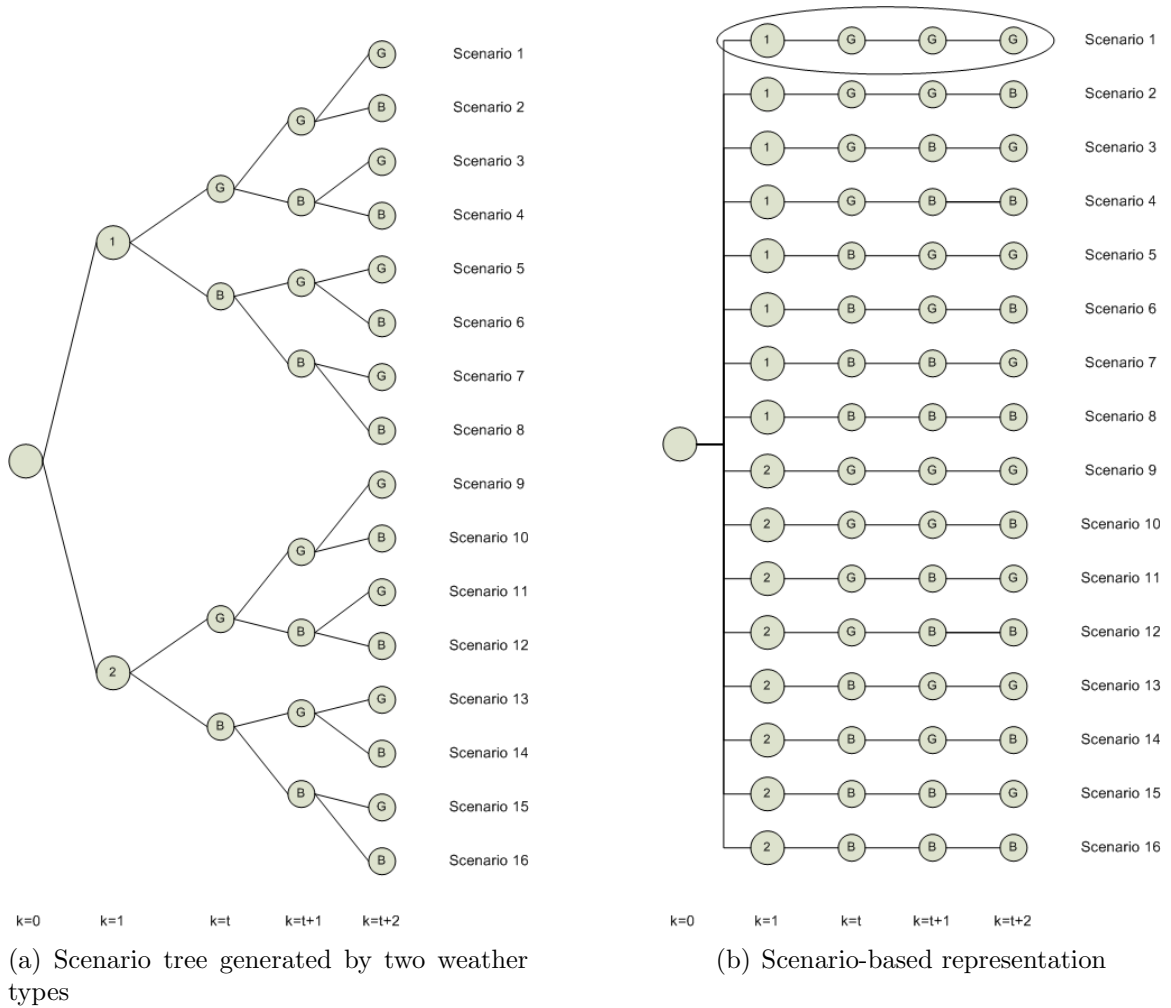
In Section 4.1, we introduce the three-stage stochastic integer programming model. Then the rolling horizon method is tested on the multistage model in Section 4.2.

Finally, Lagrangian relaxation technique is applied to the multistage model in Section 4.3. We will present the computational results corresponding to the methods we introduced in each section.

#### ***4.1 Three-Stage Stochastic Integer Programming Model for Traffic Flow Management Problem***

General assumptions such as discrete time and weather are fully described in Section 3.1.1. In addition, we introduce the concepts that will be used in the three-stage model. Assume we are at time 0 and are about to make decisions as how aircraft are sent toward the sector. In a three-stage model, we introduce additional decision point, which can be determined by when the weather forecast is going to be updated. Let it be time one, representing that the forecast is updated quarter-hour later. Suppose at time one, we can be more certain about weather in the future which is the case if the forecast improves over time. Therefore, by including both the current weather forecast at time zero and the predicted weather forecast at time, a three-stage stochastic programming model can be constructed. In Figure 17(a), we can see that by assuming two different types of weather realizations at time one, the tree grows from eight scenarios into sixteen scenarios, compared with Figure 5(a). We will implement the three-stage model as depicted in Figure 17(a) as it involves with fewer decision variables and constraints. Figure 17(b) describes the scenario-based implementation for three-stage model.

Without loss of generality, the second decision point can be set at any time. That is, we can update the future weather when it is predicted to become available. However, in order for the additional decision point to be valuable, we need to set it up so that it falls before the last period that the flights enter the sector. Moreover, for the additional decision point to benefit the most, we want it to be set between current period and the first period that the flights enter the sector. Take schedule presented in Table 9 for example, second decision point can be set between period



**Figure 17:** Implementations of three-stage stochastic programming model

zero and period thirteen ( $\max\{b_m + \Delta k_m + \Delta t_m^- + \Delta s_m + \Delta h_m\} = 13$ ). But if we set it between current period and period six ( $\min\{b_m + \Delta k_m\} = 6$ ), all possible weather realizations and flight schedule will be included so that the three-stage model will perform the best.

#### 4.1.1 The Three-Stage Stochastic Integer Programming Model

Except for the decisions variables, notations such as bounds and cost parameters are follow as introduced in Table 2 and 3, respectively. In order to distinguish the decisions made in different stages in the three-stage model, we disregard the decision

variables described in previous chapter and define sets of new decision variables here. The superscript  $n$ ,  $n = 1, 2$ , or  $3$  on each variable represents that the specific variable is the decision made in the  $n^{th}$  stage. All the decision variables are binary as they are corresponding to yes-no questions. For  $n = 1$ , we have only  $x^1$  and  $y^1$  representing the departure decisions.

$$x_m^{st1} = \begin{cases} 1 & \text{if flight } m \text{ is sent at time } s \text{ and reaches the sector at time } t \text{ in first} \\ & \text{stage} \\ 0 & \text{otherwise} \end{cases}$$

$$y_m^1 = \begin{cases} 1 & \text{if flight } m \text{ is cancelled in first stage} \\ 0 & \text{otherwise} \end{cases}$$

Based on the first weather realization, in the second stage, we have recourse actions  $q^2(\xi_1)$  and  $p^2(\xi_1)$  corresponding to the first-stage decisions. In addition, second-stage departure decisions of  $x^2(\xi_1)$  and  $y^2(\xi_1)$  are made according to the first weather realization as well.

$$q_m^{t2}(\xi_1) = \begin{cases} 1 & \text{if flight } m \text{ reaches the sector and diverts at time } t \text{ under} \\ & \text{weather } \xi_1 \text{ in second stage} \\ 0 & \text{otherwise} \end{cases}$$

$$p_m^{tu2}(\xi_1) = \begin{cases} 1 & \text{if flight } m \text{ reaches the sector at time } t \text{ and enters the sector} \\ & \text{at time } u \text{ under weather } \xi_1 \text{ in second stage} \\ 0 & \text{otherwise} \end{cases}$$

$$x_m^{st2}(\xi_1) = \begin{cases} 1 & \text{if flight } m \text{ is sent at time } s \text{ and reaches the sector at time } t \text{ under} \\ & \text{weather } \xi_1 \text{ in second stage} \\ 0 & \text{otherwise} \end{cases}$$

$$y_m^2(\xi_1) = \begin{cases} 1 & \text{if flight } m \text{ is cancelled under weather } \xi_1 \text{ in second stage} \\ 0 & \text{otherwise} \end{cases}$$

Finally, after the second weather realization, recourse actions of  $q^3(\xi_2)$  and  $p^3(\xi_2)$

corresponding to the second-stage departure decisions are made in the third stage.

$$\begin{aligned}
 q_m^{t3}(\xi_2) &= \begin{cases} 1 & \text{if flight } m \text{ reaches the sector and diverts at time } t \text{ under} \\ & \text{weather } \xi_2 \text{ in third stage} \\ 0 & \text{otherwise} \end{cases} \\
 p_m^{tu3}(\xi_2) &= \begin{cases} 1 & \text{if flight } m \text{ reaches the sector at time } t \text{ and enters the sector} \\ & \text{at time } u \text{ under weather } \xi_2 \text{ in third stage} \\ 0 & \text{otherwise} \end{cases}
 \end{aligned}$$

In addition, the probability  $p_s$  for each scenario  $s$  will be calculated that  $p_s = p^i \times p_{\xi_t}^t \times p_{\xi_{t+1}}^{t+1} \times \cdots \times p_{\xi_{t+n}}^{t+n}$ , where  $p^i$  represents the probability that the weather will be realized as type  $i$ ,  $i = 1, 2$  in the second stage,  $t$  and  $n$  depend on how we look at the problem, and a series of  $\xi_t$  rely on the specific weather types in scenario  $s$ .

Three-stage stochastic program for traffic flow management problem:

$$\begin{aligned}
\min \quad & \sum_{m \in M} \left\{ \sum_s \sum_t \left[ (g_m^s + c_m^{t-s-\Delta k_m}) \cdot \mathbf{x}_m^{st1} \right] + can_m \cdot \mathbf{y}_m^1 \right. \\
& + \sum_t \left[ d_m \cdot \mathbf{E}[\mathbf{Q}_m^{t2}(\xi_1)] + \sum_{u=t}^{t+\Delta h_m} \left( a_m^{u-t} \cdot \mathbf{E}[\mathbf{P}_m^{tu2}(\xi_1)] \right) \right] \\
& + \sum_s \sum_t \left[ (g_m^s + c_m^{t-s-\Delta k_m}) \cdot \mathbf{E}[\mathbf{X}_m^{st2}(\xi_1)] \right] + can_m \cdot \mathbf{E}[\mathbf{Y}_m^2(\xi_1)] \\
& \left. + \sum_t \left[ d_m \cdot \mathbf{E}[\mathbf{Q}_m^{t3}(\xi_2)] + \sum_{u=t}^{t+\Delta h_m} \left( a_m^{u-t} \cdot \mathbf{E}[\mathbf{P}_m^{tu3}(\xi_2)] \right) \right] \right\} \quad (39)
\end{aligned}$$

$$s.t. \quad \sum_{m \in M} \sum_t [\mathbf{p}_m^{tu2}(\xi_1) + \mathbf{p}_m^{tu3}(\xi_2)] \leq C_{\xi_u}^u \quad \forall u, \xi_1, \xi_2 \quad (40)$$

$$\mathbf{q}_m^{t2}(\xi_1) + \sum_{u=t}^{t+\Delta h_m} \mathbf{p}_m^{tu2}(\xi_1) = \sum_s \mathbf{x}_m^{st1} \quad \forall t, m, \xi_1 \quad (41)$$

$$\mathbf{q}_m^{t3}(\xi_2) + \sum_{u=t}^{t+\Delta h_m} \mathbf{p}_m^{tu3}(\xi_2) = \sum_s \mathbf{x}_m^{st2}(\xi_1) \quad \forall t, m, \xi_1, \xi_2 \quad (42)$$

$$\sum_{s=b_m}^{b_m+\Delta s_m} \sum_{t=s+\Delta k_m-\Delta t_m^+}^{s+\Delta k_m+\Delta t_m^-} [\mathbf{x}_m^{st1} + \mathbf{x}_m^{st2}(\xi_1)] + \mathbf{y}_m^1 + \mathbf{y}_m^2(\xi_1) = 1 \quad \forall m, \xi_1 \quad (43)$$

$$\mathbf{x}_m^{st1} \in \{0, 1\} \quad \forall s, t, m \quad (44)$$

$$\mathbf{x}_m^{st2}(\xi_1) \in \{0, 1\} \quad \forall s, t, m, \xi_1 \quad (45)$$

$$\mathbf{y}_m^1 \in \{0, 1\} \quad \forall m \quad (46)$$

$$\mathbf{y}_m^2(\xi_1) \in \{0, 1\} \quad \forall m, \xi_1 \quad (47)$$

$$\mathbf{p}_m^{tu2}(\xi_1) \in \{0, 1\} \quad \forall t, u, m, \xi_1 \quad (48)$$

$$\mathbf{p}_m^{tu3}(\xi_2) \in \{0, 1\} \quad \forall t, u, m, \xi_2 \quad (49)$$

$$\mathbf{q}_m^{t2}(\xi_1) \in \{0, 1\} \quad \forall t, m, \xi_1 \quad (50)$$

$$\mathbf{q}_m^{t3}(\xi_2) \in \{0, 1\} \quad \forall t, m, \xi_2 \quad (51)$$

Due to the typesetting, we comment some of the summations in words. In the first and third lines of the objective function, the  $s$  sums from  $b_m$  to  $b_m + \Delta s_m$  while the  $t$  sums from  $s + \Delta k_m - \Delta t_m^+$  to  $s + \Delta k_m + \Delta t_m^-$ . The  $t$  in the second and fourth lines sums from  $b_m + \Delta k_m - \Delta t_m^+$  to  $b_m + \Delta s_m + \Delta k_m + \Delta t_m^-$ .



The objective function (39) contains four lines. The first line is the summation of the ground-delay, speed-change, and cancellation costs in the first stage while the second line contains the expected air-holding and diversion costs in the second stage. The third line of the objective function includes the second-stage expected ground-delay, speed-change, and cancellation costs while the fourth line sums the expected air-holding and diversion costs in the third stage. Constraint (40) ensures that the number of flights in the sector does not exceed the sector capacity. Constraints (41) and (42) ensure that each sent flight will either be diverted or be accepted by the sector in second and third stages, respectively. Constraint (43) ensures that every flight in the problem will be sent or cancelled exactly once either in the first or second stages. Constraints (44)-(51) ensure that all the decision variables are binary.

#### 4.1.2 Experimental Setting

The experimental setting follows the descriptions in Section 3.1.3. In addition, in three-stage model, we need to define  $p^i$ ,  $i = 1, 2$ , which are the probabilities for the weather realizations at the second decision point. As a result, we need two sets of sector capacities and corresponding probabilities. In the following implementations, we will utilize the weather forecast as used in two-stage model for  $i = 1$ , and generate another weather forecast for  $i = 2$ . Per the difficulties described in Section 3.1.3.4, here, we will randomly generate the second set of weather forecast.

#### 4.1.3 Computational Results

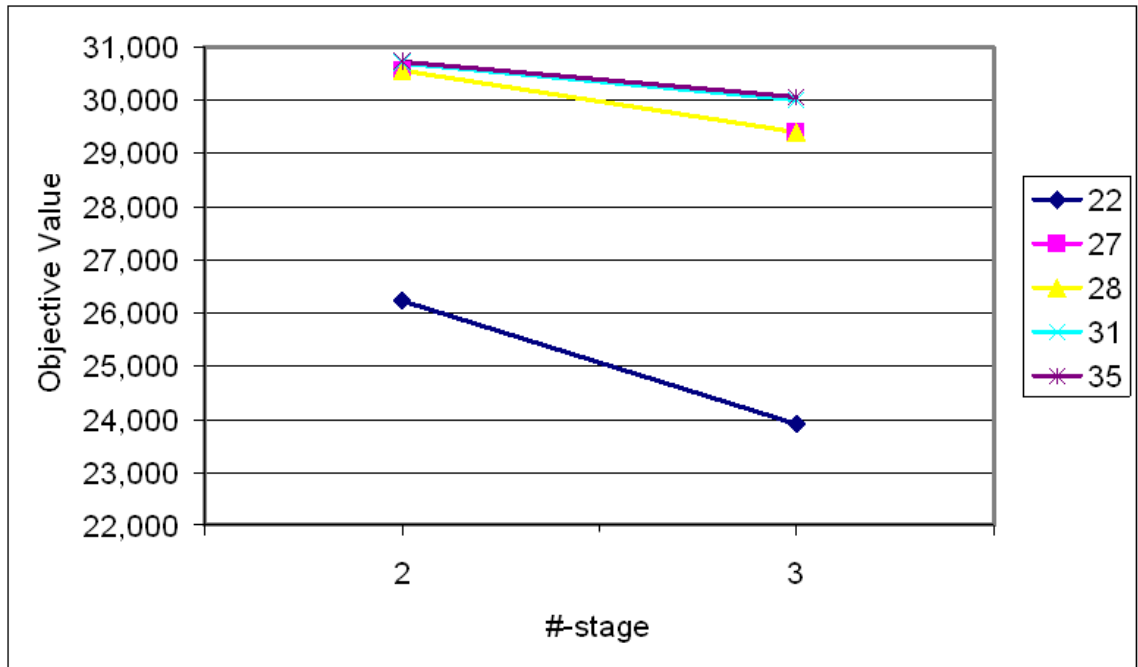
Based on the computational setting described in the previous section, we test our three-stage model in the environment introduced in Section 3.1.3.5. The computational results for the three-stage model are presented in Table 25. We can see that due to the increased number of scenarios, the capabilities to produce optimal solutions for the three-stage model decrease.

Compared with Table 7, we can see that the three-stage model indeed improves the

**Table 25:** Computational results for three-stage stochastic integer programming model

Flights departure periods	Periods involved	Number of scenarios	Number of flights	Objective value	Time (sec)
1-1	4-13	2,048	22	23,916.18	55
1-2	4-13	2,048	27	29,402.30	79
1-3	4-14	4,096	28	29,404.99	182
1-4	4-16	16,384	31	30,008.38	4,783
1-5	4-17	32,768	35	30,050.45	9,996
1-6	4-17	32,768	44	n/a	n/a

objective value. We plot the objective values in Figure 18 for a comparison between two- and three- stage models.



**Figure 18:** Comparison between 2- and 3-stage models

To compare how big the problems are with two-stage model, we summarize the number of columns and rows for each instance in Table 26. The number of constraints does not include the binary variables. The number of columns and rows basically doubled compared with two-stage model which is due to the additional stage.

**Table 26:** List of numbers of columns and rows for three-stage model

Flight count	Number of scenarios	Number of columns	Number of rows
22	2,048	357,024	164,024
27	2,048	443,193	194,784
28	4,096	918,360	409,834
31	16,384	4,129,734	1,851,654
35	32,768	9,438,294	4,260,142
44	32,768	11,732,314	5,144,950

Figure 19 depicts the cost composition for the instance presented in Table 25 by different flight counts. We can see that for this particular example, similar to the two-stage model, the three-stage model tends to cancel or ground-delay flights rather than sending out them, which will face future weather uncertainties. In those scenarios that the weather turns to be bad, the flights are sent to the sector so that only a few flights under inclement weather will be diverted. As a result, we say that the three-stage model leans to conservatively incur low prices before the realizations of uncertainties to avoid huge costs resulted in severe weather.

It is interesting to see that almost all costs incurred in the three-stage model come from the decisions made in the second decision point. The only cost results in the first decision is the speed-change cost. See Figure 20 for the comparison. We can draw conclusion for this specific example that the three-stage model tends to make decisions when more weather information is available.

The example above shows that it takes almost three hours to solve a 35-flight case, which is inapplicable to real-time problem-solving. We have seen in Section 3.2 about the successful implementations of the rolling horizon method, here, we will test the rolling horizon method again on the three-stage model.

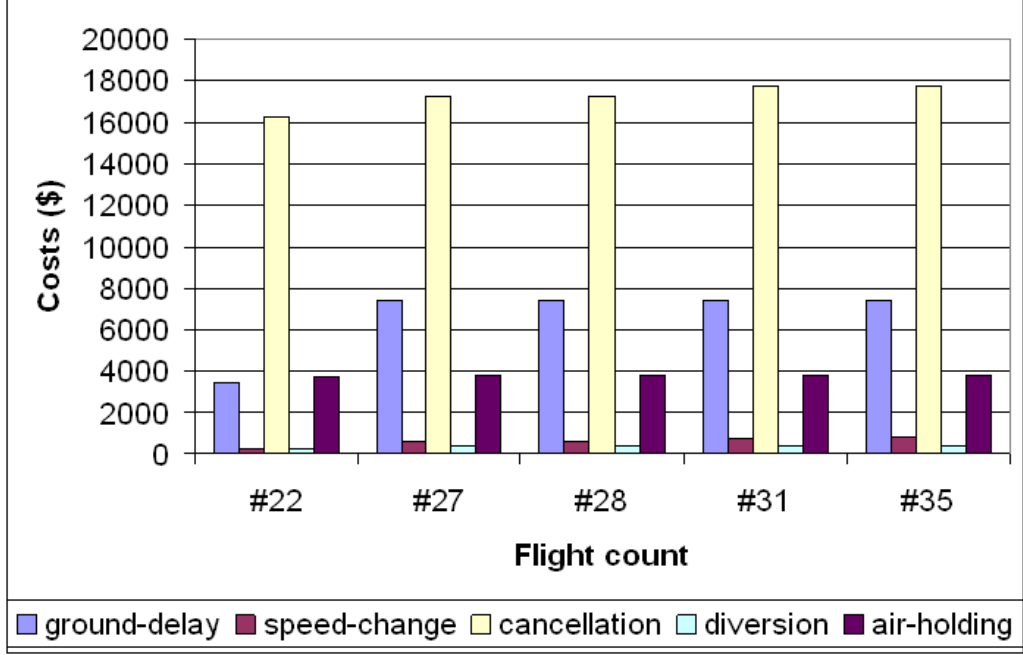


Figure 19: Cost composition for three-stage model

## 4.2 Approaches to the Three-Stage Model: The Rolling Horizon Method

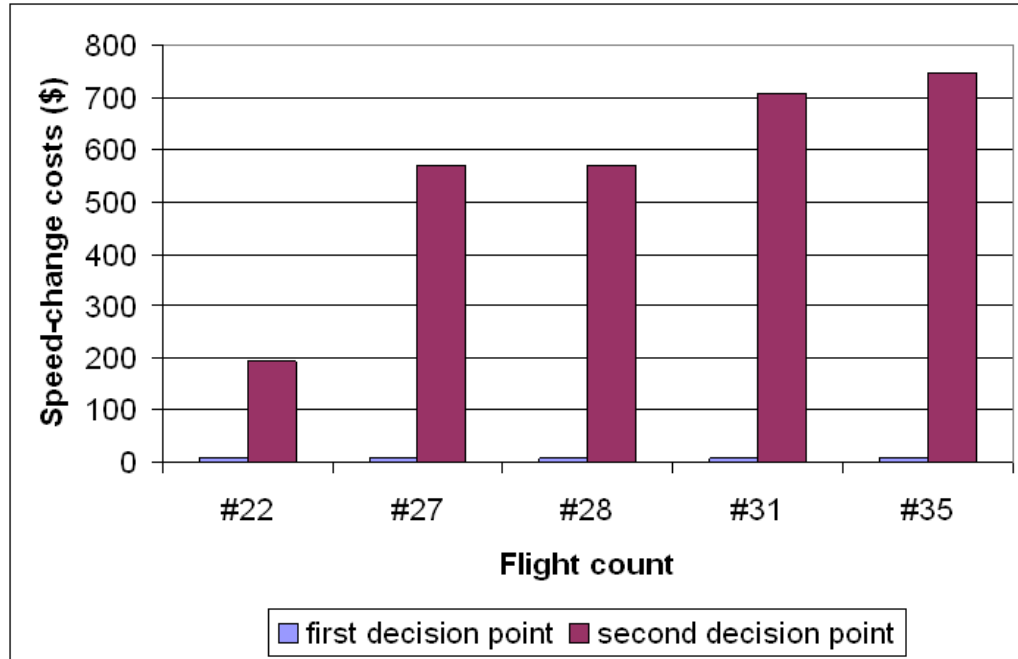
The ideas and techniques used in the two-stage rolling horizon method remain the same as described previously. And as the assumption made in the three-stage model that the second decision point is at time one, the three-stage rolling horizon method still keeps it.

### 4.2.1 The Rolling Horizon Method

The steps for three-stage rolling horizon method follow (16) and the modified three-stage stochastic program is described as (39)-(51) except that (40) should be replaced as follows.

$$\sum_{m \in M} \sum_t [\mathbf{p}_m^{tu}(\xi_1) + \mathbf{p}_m^{tu}(\xi_2)] \leq C_{\xi_u}^u - \sum_{m \in M} \sum_{t'} [\mathbf{p}_m^{tu}(\xi_1) + \mathbf{p}_m^{tu}(\xi_2)] \quad \forall u, \xi_1, \xi_2 \quad (52)$$

The  $t$  in the left-hand side of (52) sums from  $\max\{b_m + \Delta k_m - \Delta t_m^+, u - \Delta h_m\}$  to  $\min\{b_m + \Delta s_m + \Delta k_m + \Delta t_m^- + \Delta h_m, u\}$ , and the  $t'$  in the right-hand side sums the



**Figure 20:** Comparison of speed-change costs in different stages

same things but in the previous iteration. The sector capacity is updated to account for the capacities used by aircraft in the previous iteration.

#### 4.2.2 Computational Experiments

The comparison between the solutions from the three-stage stochastic program and the rolling horizon method is presented in Table (27)-(29). The ideas of PD and fuPD are the same as defined in Section 3.2.2. PD indicates the number of periods that we mainly focus on so that the flights scheduled to reach the sector during these periods will be included, while fuPD is the buffer periods that only flights with modified schedule can enter the sector during these periods. Table (27), (28), and (29) show the computational results for PD=9, 10, and 11, while the fuPD is set to two in all instances.

The rolling horizon method gives good results within reasonable time. For an extreme large instance that takes about three hours to solve optimally, the rolling

**Table 27:** Computational results for three-stage rolling horizon method - PD=9

Flights depart in period	# of flight	Solve to Optimality			Rolling Horizon Method			
		Objective value	Time (sec)	Period	Modified objective	Time (sec)	# of iteration	% from optimal
1-1	22	23,916.18	55	4-13	24,437.46	3	1	2.18%
1-2	27	29,402.30	79	4-13	30,359.27	4	1	3.25%
1-3	28	29,404.99	182	4-14	30,360.07	5	3	3.25%
1-4	31	30,008.38	4,783	4-16	31,050.15	4	4	3.47%
1-5	35	30,050.45	9,996	4-17	31,061.90	4	5	3.37%
1-6	44	n/a	n/a	4-17	31,702.18	8	6	n/a

**Table 28:** Computational results for three-stage rolling horizon method - PD=10

Flights depart in period	# of flight	Solve to Optimality			Rolling Horizon Method			
		Objective value	Time (sec)	Period	Modified objective	Time (sec)	# of iteration	% from optimal
1-1	22	23,916.18	55	4-13	23,916.18	22	1	0.00%
1-2	27	29,402.30	79	4-13	29,402.30	12	1	0.00%
1-3	28	29,404.99	182	4-14	29,408.72	13	2	0.01%
1-4	31	30,008.38	4,783	4-16	30,075.74	14	3	0.22%
1-5	35	30,050.45	9,996	4-17	30,122.08	16	4	0.24%
1-6	44	n/a	n/a	4-17	30,746.53	70	5	n/a

horizon method produces an modified objective that is within 0.01 percent of the optimal value in less than one minute. To approach a real-life flight schedule, the rolling horizon method does really good by providing near-optimal solutions in seconds.

### 4.3 Lagrangian Relaxation

In order to justify the rolling horizon method for those cases without optimal solutions, we apply the Lagrangian relaxation technique to approach the optimal objective

**Table 29:** Computational results for three-stage rolling horizon method - PD=11

Flights depart in period	# of flight	Solve to Optimality			Rolling Horizon Method			
		Objective value	Time (sec)	Period	Modified objective	Time (sec)	# of iteration	% from optimal
1-1	22	23,916.18	55	4-13	23,916.18	55	1	0.00%
1-2	27	29,402.30	79	4-13	29,402.30	36	1	0.00%
1-3	28	29,404.99	182	4-14	29,404.99	40	1	0.00%
1-4	31	30,008.38	4,783	4-16	30,011.09	46	2	0.01%
1-5	35	30,050.45	9,996	4-17	30,054.24	49	3	0.01%
1-6	44	n/a	n/a	4-17	30,663.20	88	4	n/a

value. Similar to the Lagrangian relaxation method done for the two-stage stochastic program, we are going to relax the nonanticipativity constraints in Lagrangian fashion. However, our decision variables are defined so that no explicit nonanticipativity constraints are listed in problem (39)-(51), see Figure 17(a) for illustration. As a result, we need to add decision variables to first and second stages so that each variable corresponds to each of the third-stage scenarios. Decision variables added for first-stage variables are

$$\mathbf{x}_m^{st1} \Rightarrow \mathbf{x}_m^{st1}(\xi_2) \quad \forall \xi_2, \text{ and} \quad (53)$$

$$\mathbf{y}_m^1 \Rightarrow \mathbf{y}_m^1(\xi_2) \quad \forall \xi_2, \quad (54)$$

and for second-stage variables are

$$\mathbf{q}_m^{t2}(\xi_1) \Rightarrow \mathbf{q}_m^{t2}(\xi_2) \quad \forall \xi_2, \quad (55)$$

$$\mathbf{p}_m^{tu2}(\xi_1) \Rightarrow \mathbf{p}_m^{tu2}(\xi_2) \quad \forall \xi_2, \quad (56)$$

$$\mathbf{x}_m^{st2}(\xi_1) \Rightarrow \mathbf{x}_m^{st2}(\xi_2) \quad \forall \xi_2, \text{ and} \quad (57)$$

$$\mathbf{y}_m^2(\xi_1) \Rightarrow \mathbf{y}_m^2(\xi_2) \quad \forall \xi_2. \quad (58)$$

If we let  $\Omega_2$  be the set of possible weather realizations in the third stage, each of (53) and (54) expands the number of first-stage decision variables from one to  $|\Omega_2|$  while

each of (55)-(58) expands the number of second-stage decision variables from two to  $\|\Omega_2\|$ . Note that all variables we constructed in (53)-(58) are also binary.

The next step is to add explicitly the nonanticipativity constraints to the (39)-(51) model. For (53) and (54), the values of all the replicas should be the same. Therefore, nonanticipativity constraints can be described by (59) and (60).

$$\mathbf{x}_m^{st1}(1) = \mathbf{x}_m^{st1}(\xi_2) \quad \forall \xi_2 \neq 1 \quad (59)$$

$$\mathbf{y}_m^1(1) = \mathbf{y}_m^1(\xi_2) \quad \forall \xi_2 \neq 1 \quad (60)$$

For (55)-(58), it is a little tricky to add the nonanticipativity constraints. Note that each of the right-hand side of (55)-(58) is duplicated from two second-stage variables. As a result, not all of the duplicated variables are equal. Instead, they can be distinguished by two sets, which correspond to two different decisions made in the second-stage. If we let  $\Xi_a$  be the set of  $\xi_2$  such that  $1 < \xi_2 \leq \frac{\|\Omega_2\|}{2}$ ,  $\Xi_b$  be the set of  $\xi_2$  such that  $\frac{\|\Omega_2\|}{2} + 1 < \xi_2 \leq \|\Omega_2\|$ , and  $k = \frac{\|\Omega_2\|}{2} + 1$ , then the nonanticipativity constraints correspond to (55)-(58) can be described by (61)-(68), which are described in details in the following.

$$\text{constraint (55)} \Rightarrow \quad \mathbf{q}_m^{t2}(1) = \mathbf{q}_m^{t2}(\xi_2) \quad \forall \xi_2 \in \Xi_a \quad (61)$$

$$\text{constraint (55)} \Rightarrow \quad \mathbf{q}_m^{t2}(k) = \mathbf{q}_m^{t2}(\xi_2) \quad \forall \xi_2 \in \Xi_b \quad (62)$$

$$\text{constraint (56)} \Rightarrow \quad \mathbf{p}_m^{tu2}(1) = \mathbf{p}_m^{tu2}(\xi_2) \quad \forall \xi_2 \in \Xi_a \quad (63)$$

$$\text{constraint (56)} \Rightarrow \quad \mathbf{p}_m^{tu2}(k) = \mathbf{p}_m^{tu2}(\xi_2) \quad \forall \xi_2 \in \Xi_b \quad (64)$$

$$\text{constraint (57)} \Rightarrow \quad \mathbf{x}_m^{st2}(1) = \mathbf{x}_m^{st2}(\xi_2) \quad \forall \xi_2 \in \Xi_a \quad (65)$$

$$\text{constraint (57)} \Rightarrow \quad \mathbf{x}_m^{st2}(k) = \mathbf{x}_m^{st2}(\xi_2) \quad \forall \xi_2 \in \Xi_b \quad (66)$$

$$\text{constraint (58)} \Rightarrow \quad \mathbf{y}_m^2(1) = \mathbf{y}_m^2(\xi_2) \quad \forall \xi_2 \in \Xi_a \quad (67)$$

$$\text{constraint (58)} \Rightarrow \quad \mathbf{y}_m^2(k) = \mathbf{y}_m^2(\xi_2) \quad \forall \xi_2 \in \Xi_b \quad (68)$$

If we replace decision variables by (53)-(58) and add constraints (59)-(68) to stochastic program (39)-(51), then the nonanticipativity constraint (59)-(68) are the only linking



constraints among scenarios. Therefore, the three-stage model can be decomposed by scenario  $\xi_2$  if we relax the nonanticipativity constraints. Lagrangian multipliers of  $\lambda_m^{st1}(\xi_2)$ ,  $\forall \xi_2 \neq 1$  is introduced for not satisfying constraint (59) and  $\theta_m^1(\xi_2)$ ,  $\forall \xi_2 \neq 1$  is placed for not satisfying constraint (60). In addition, for all  $\xi_2 \in \Xi_a$ ,  $\pi_m^{t2a}(\xi_2)$ ,  $\mu_m^{tu2a}(\xi_2)$ ,  $\lambda_m^{st2a}(\xi_2)$ , and  $\theta_m^{2a}(\xi_2)$  are the Lagrange multipliers introduced for not satisfying constraints (61), (63), (65), and (67), respectively. For all  $\xi_2 \in \Xi_b$ ,  $\pi_m^{t2b}(\xi_2)$ ,  $\mu_m^{tu2b}(\xi_2)$ ,  $\lambda_m^{st2b}(\xi_2)$ , and  $\theta_m^{2b}(\xi_2)$  are the Lagrange multipliers introduced for not satisfying constraints (62), (64), (66), and (68), respectively. Note that the constraints (59)-(68) we relaxed are equalities, therefore, the corresponding Lagrange multipliers are unrestricted in sign. Furthermore, (69)-(78) are added to the objective function of the three-stage model.

$$\lambda_m^{st1}(\xi_2) [\mathbf{x}_m^{st1}(1) - \mathbf{x}_m^{st1}(\xi_2)] \quad \forall \xi_2 \neq 1 \quad (69)$$

$$\theta_m^1(\xi_2) [\mathbf{y}_m^1(1) - \mathbf{y}_m^1(\xi_2)] \quad \forall \xi_2 \neq 1 \quad (70)$$

$$\pi_m^{t2a}(\xi_2) [\mathbf{q}_m^{t2}(1) - \mathbf{q}_m^{t2}(\xi_2)] \quad \forall \xi_2 \in \Xi_a \quad (71)$$

$$\pi_m^{t2b}(\xi_2) [\mathbf{q}_m^{t2}(k) - \mathbf{q}_m^{t2}(\xi_2)] \quad \forall \xi_2 \in \Xi_b \quad (72)$$

$$\mu_m^{tu2a}(\xi_2) [\mathbf{p}_m^{tu2}(1) - \mathbf{p}_m^{tu2}(\xi_2)] \quad \forall \xi_2 \in \Xi_a \quad (73)$$

$$\mu_m^{tu2b}(\xi_2) [\mathbf{p}_m^{tu2}(k) - \mathbf{p}_m^{tu2}(\xi_2)] \quad \forall \xi_2 \in \Xi_b \quad (74)$$

$$\lambda_m^{st2a}(\xi_2) [\mathbf{x}_m^{st2}(1) - \mathbf{x}_m^{st2}(\xi_2)] \quad \forall \xi_2 \in \Xi_a \quad (75)$$

$$\lambda_m^{st2b}(\xi_2) [\mathbf{x}_m^{st2}(k) - \mathbf{x}_m^{st2}(\xi_2)] \quad \forall \xi_2 \in \Xi_b \quad (76)$$

$$\theta_m^{2a}(\xi_2) [\mathbf{y}_m^2(1) - \mathbf{y}_m^2(\xi_2)] \quad \forall \xi_2 \in \Xi_a \quad (77)$$

$$\theta_m^{2b}(\xi_2) [\mathbf{y}_m^2(k) - \mathbf{y}_m^2(\xi_2)] \quad \forall \xi_2 \in \Xi_b \quad (78)$$

Let  $L(x^1, y^1, q^2, p^2, x^2, y^2, q^3, p^3)$  be the resulting objective in the Lagrangian relaxation problem, then the relaxed problem can be decomposed by scenarios so that

$$\begin{aligned}
 & L(x^1, y^1, q^2, p^2, x^2, y^2, q^3, p^3) \\
 = & \sum_{\xi_2 \in \Xi} L_{\xi}(x^1(\xi_2), y^1(\xi_2), q^2(\xi_2), p^2(\xi_2), x^2(\xi_2), y^2(\xi_2), q^3(\xi_2), p^3(\xi_2)), \quad (79)
 \end{aligned}$$

$$\begin{aligned}
& \text{where } L(x^1(1), y^1(1), q^2(1), p^2(1), x^2(1), y^2(1), q^3(1), p^3(1)) \\
&= \sum_m \left\{ \sum_s \sum_t \left[ (g+c) \cdot p_1 + \sum_{\xi_2 \neq 1} \lambda_m^{st1}(\xi_2) \right] \cdot \mathbf{x}_m^{st1}(1) + \left[ can \cdot p_1 + \sum_{\xi_2 \neq 1} \theta_m^1(\xi_2) \right] \cdot \mathbf{y}_m^1(1) \right. \\
&+ \sum_t \left[ d \cdot p_1 + \sum_{\xi_2 \in \Xi_a} \pi_m^{t2a}(\xi_2) \right] \cdot \mathbf{q}_m^{t2}(1) + \sum_t \sum_u \left[ a \cdot p_1 + \sum_{\xi_2 \in \Xi_a} \mu_m^{tu2a}(\xi_2) \right] \cdot \mathbf{p}_m^{tu2}(1) \\
&+ \sum_s \sum_t \left[ (g+c) \cdot p_1 + \sum_{\xi_2 \in \Xi_a} \lambda_m^{st2a}(\xi_2) \right] \cdot \mathbf{x}_m^{st2}(1) + \left[ can \cdot p_1 + \sum_{\xi_2 \in \Xi_a} \theta_m^{2a}(\xi_2) \right] \cdot \mathbf{y}_m^2(1) \\
&\left. + \sum_t \left[ d \cdot \mathbf{q}_m^{t3}(1) + \sum_u a \cdot \mathbf{p}_m^{tu3}(1) \right] \cdot p_1 \right\},
\end{aligned}$$

$$\begin{aligned}
& L_\xi(x^1(\xi_2), y^1(\xi_2), q^2(\xi_2), p^2(\xi_2), x^2(\xi_2), y^2(\xi_2), q^3(\xi_2), p^3(\xi_2)) \\
&= \sum_m \left\{ \sum_s \sum_t \left[ (g+c) \cdot p_{\xi_2} - \lambda_m^{st1}(\xi_2) \right] \cdot \mathbf{x}_m^{st1}(\xi_2) + \left[ can \cdot p_{\xi_2} - \theta_m^1(\xi_2) \right] \cdot \mathbf{y}_m^1(\xi_2) \right. \\
&+ \sum_t \left[ d \cdot p_{\xi_2} - \pi_m^{t2a}(\xi_2) \right] \cdot \mathbf{q}_m^{t2}(\xi_2) + \sum_t \sum_u \left[ a \cdot p_{\xi_2} - \mu_m^{tu2a}(\xi_2) \right] \cdot \mathbf{p}_m^{tu2}(\xi_2) \\
&+ \sum_s \sum_t \left[ (g+c) \cdot p_{\xi_2} - \lambda_m^{st2a}(\xi_2) \right] \cdot \mathbf{x}_m^{st2}(\xi_2) + \left[ can \cdot p_{\xi_2} - \theta_m^{2a}(\xi_2) \right] \cdot \mathbf{y}_m^{2a}(\xi_2) \\
&\left. + \sum_t \left[ d \cdot \mathbf{q}_m^{t3}(\xi_2) + \sum_u a \cdot \mathbf{p}_m^{tu3}(\xi_2) \right] \cdot p_{\xi_2} \right\}, \quad \forall \xi_2 \in \Xi_a,
\end{aligned}$$

$$\begin{aligned}
& L(x^1(k), y^1(k), q^2(k), p^2(k), x^2(k), y^2(k), q^3(k), p^3(k)) \\
&= \sum_m \left\{ \sum_s \sum_t \left[ (g+c) \cdot p_k - \lambda_m^{st1}(k) \right] \cdot \mathbf{x}_m^{st1}(k) + \left[ can \cdot p_k - \theta_m^1(k) \right] \cdot \mathbf{y}_m^1(k) \right. \\
&+ \sum_t \left[ d \cdot p_k + \sum_{\xi_2 \in \Xi_b} \pi_m^{t2b}(\xi_2) \right] \cdot \mathbf{q}_m^{t2}(k) + \sum_t \sum_u \left[ a \cdot p_k + \sum_{\xi_2 \in \Xi_b} \mu_m^{tu2b}(\xi_2) \right] \cdot \mathbf{p}_m^{tu2}(k) \\
&+ \sum_s \sum_t \left[ (g+c) \cdot p_k + \sum_{\xi_2 \in \Xi_b} \lambda_m^{st2b}(\xi_2) \right] \cdot \mathbf{x}_m^{st2}(k) + \left[ can \cdot p_k + \sum_{\xi_2 \in \Xi_b} \theta_m^{2b}(\xi_2) \right] \cdot \mathbf{y}_m^2(k) \\
&\left. + \sum_t \left[ d \cdot \mathbf{q}_m^{t3}(k) + \sum_u a \cdot \mathbf{p}_m^{tu3}(k) \right] \cdot p_k \right\},
\end{aligned}$$

$$\begin{aligned}
& L_\xi(x^1(\xi_2), y^1(\xi_2), q^2(\xi_2), p^2(\xi_2), x^2(\xi_2), y^2(\xi_2), q^3(\xi_2), p^3(\xi_2)) \\
&= \sum_m \left\{ \sum_s \sum_t \left[ (g+c) \cdot p_{\xi_2} - \lambda_m^{st1}(\xi_2) \right] \cdot \mathbf{x}_m^{st1}(\xi_2) + \left[ can \cdot p_{\xi_2} - \theta_m^1(\xi_2) \right] \cdot \mathbf{y}_m^1(\xi_2) \right. \\
&+ \sum_t \left[ d \cdot p_{\xi_2} - \pi_m^{t2b}(\xi_2) \right] \cdot \mathbf{q}_m^{t2}(\xi_2) + \sum_t \sum_u \left[ a \cdot p_{\xi_2} - \mu_m^{tu2b}(\xi_2) \right] \cdot \mathbf{p}_m^{tu2}(\xi_2) \\
&+ \sum_s \sum_t \left[ (g+c) \cdot p_{\xi_2} - \lambda_m^{st2b}(\xi_2) \right] \cdot \mathbf{x}_m^{st2}(\xi_2) + \left[ can \cdot p_{\xi_2} - \theta_m^{2b}(\xi_2) \right] \cdot \mathbf{y}_m^2(\xi_2) \\
&\left. + \sum_t \left[ d \cdot \mathbf{q}_m^{t3}(\xi_2) + \sum_u a \cdot \mathbf{p}_m^{tu3}(\xi_2) \right] \cdot p_{\xi_2} \right\}, \quad \forall \xi_2 \in \Xi_b.
\end{aligned}$$

The Lagrangian relaxation problem of the three-stage model can now be described as objective function (79) subjects to (80)-(91), which are described below.

$$\min \quad (79)$$

$$s.t. \quad \sum_m \sum_t [\mathbf{p}_m^{tu2}(\xi_2) + \mathbf{p}_m^{tu3}(\xi_2)] \leq C_{\xi_u}^u \quad \forall u, \xi_2 \quad (80)$$

$$\mathbf{q}_m^{t2}(\xi_2) + \sum_{u=t}^{t+\Delta h_m} \mathbf{p}_m^{tu2}(\xi_2) = \sum_s \mathbf{x}_m^{st1}(\xi_2) \quad \forall t, m, \xi_2 \quad (81)$$

$$\mathbf{q}_m^{t3}(\xi_2) + \sum_{u=t}^{t+\Delta h_m} \mathbf{p}_m^{tu3}(\xi_2) = \sum_s \mathbf{x}_m^{st2}(\xi_2) \quad \forall t, m, \xi_2 \quad (82)$$

$$\sum_{s=b_m}^{b_m+\Delta s_m} \sum_{t=s+\Delta k_m-\Delta t_m^+}^{s+\Delta k_m+\Delta t_m^-} [\mathbf{x}_m^{st1}(\xi_2) + \mathbf{x}_m^{st2}(\xi_2)] + \mathbf{y}_m^1(\xi_2) + \mathbf{y}_m^2(\xi_2) = 1 \quad \forall m, \xi_1 \quad (83)$$

$$\mathbf{x}_m^{st1}(\xi_2) \in \{0, 1\} \quad \forall s, t, m, \xi_2 \quad (84)$$

$$\mathbf{x}_m^{st2}(\xi_2) \in \{0, 1\} \quad \forall s, t, m, \xi_2 \quad (85)$$

$$\mathbf{y}_m^1(\xi_2) \in \{0, 1\} \quad \forall m, \xi_2 \quad (86)$$

$$\mathbf{y}_m^2(\xi_2) \in \{0, 1\} \quad \forall m, \xi_2 \quad (87)$$

$$\mathbf{p}_m^{tu2}(\xi_2) \in \{0, 1\} \quad \forall t, u, m, \xi_2 \quad (88)$$

$$\mathbf{p}_m^{tu3}(\xi_2) \in \{0, 1\} \quad \forall t, u, m, \xi_2 \quad (89)$$

$$\mathbf{q}_m^{t2}(\xi_2) \in \{0, 1\} \quad \forall t, m, \xi_2 \quad (90)$$

$$\mathbf{q}_m^{t3}(\xi_2) \in \{0, 1\} \quad \forall t, m, \xi_2 \quad (91)$$

Problem (79)-(91) provides lower bounds for the three-stage model (39)-(51). If we let the optimal value of (79)-(91) be  $Z^3(LR)$ , then we form the Lagrangian dual problem as

$$\max \quad Z^3(LR) \quad (92)$$

to obtain a tighter lower bound. Since the Lagrange multiplies are unrestricted in sign, problem (92) subjects to nothing.

### 4.3.1 Computational Results

We solve the Lagrangian dual problem (92) by subgradient method. Given initial values of  $\lambda^{1,0}$ ,  $\theta^{1,0}$ ,  $\pi^{2a,0}$ ,  $\pi^{2b,0}$ ,  $\mu^{2a,0}$ ,  $\mu^{2b,0}$ ,  $\lambda^{2a,0}$ ,  $\lambda^{2b,0}$ ,  $\theta^{2a,0}$ , and  $\theta^{2b,0}$ , a set of  $\lambda^{1,s}$ ,  $\theta^{1,s}$ ,  $\pi^{2a,s}$ ,  $\pi^{2b,s}$ ,  $\mu^{2a,s}$ ,  $\mu^{2b,s}$ ,  $\lambda^{2a,s}$ ,  $\lambda^{2b,s}$ ,  $\theta^{2a,s}$ , and  $\theta^{2b,s}$  can be updated by (93)-(102), respectively.

$$\lambda^{1,s+1}(\xi_2) = \lambda^{1,s}(\xi_2) + \alpha^{1,s} \cdot [\mathbf{x}^{1,s}(1) - \mathbf{x}^{1,s}(\xi_2)], \quad \forall \xi \neq 1, \quad (93)$$

$$\theta^{1,s+1}(\xi_2) = \theta^{1,s}(\xi_2) + \beta^{1,s} \cdot [\mathbf{y}^{1,s}(1) - \mathbf{y}^{1,s}(\xi_2)], \quad \forall \xi \neq 1, \quad (94)$$

$$\pi^{2a,s+1}(\xi_2) = \pi^{2a,s}(\xi_2) + \gamma^{2a,s} \cdot [\mathbf{q}^{2,s}(1) - \mathbf{q}^{2,s}(\xi_2)], \quad \forall \xi \in \Xi_a, \quad (95)$$

$$\pi^{2b,s+1}(\xi_2) = \pi^{2b,s}(\xi_2) + \gamma^{2b,s} \cdot [\mathbf{q}^{2,s}(k) - \mathbf{q}^{2,s}(\xi_2)], \quad \forall \xi \in \Xi_b, \quad (96)$$

$$\mu^{2a,s+1}(\xi_2) = \mu^{2a,s}(\xi_2) + \eta^{2a,s} \cdot [\mathbf{p}^{2,s}(1) - \mathbf{p}^{2,s}(\xi_2)], \quad \forall \xi \in \Xi_a, \quad (97)$$

$$\mu^{2b,s+1}(\xi_2) = \mu^{2b,s}(\xi_2) + \eta^{2b,s} \cdot [\mathbf{p}^{2,s}(k) - \mathbf{p}^{2,s}(\xi_2)], \quad \forall \xi \in \Xi_b, \quad (98)$$

$$\lambda^{2a,s+1}(\xi_2) = \lambda^{2a,s}(\xi_2) + \alpha^{2a,s} \cdot [\mathbf{x}^{2,s}(1) - \mathbf{x}^{2,s}(\xi_2)], \quad \forall \xi \in \Xi_a, \quad (99)$$

$$\lambda^{2b,s+1}(\xi_2) = \lambda^{2b,s}(\xi_2) + \alpha^{2b,s} \cdot [\mathbf{x}^{2,s}(k) - \mathbf{x}^{2,s}(\xi_2)], \quad \forall \xi \in \Xi_b, \quad (100)$$

$$\theta^{2a,s+1}(\xi_2) = \theta^{2a,s}(\xi_2) + \beta^{2a,s} \cdot [\mathbf{y}^{2,s}(1) - \mathbf{y}^{2,s}(\xi_2)], \quad \forall \xi \in \Xi_a, \text{ and} \quad (101)$$

$$\theta^{2b,s+1}(\xi_2) = \theta^{2b,s}(\xi_2) + \beta^{2b,s} \cdot [\mathbf{y}^{2,s}(k) - \mathbf{y}^{2,s}(\xi_2)], \quad \forall \xi \in \Xi_b. \quad (102)$$

$\alpha^{1,s}$ ,  $\beta^{1,s}$ ,  $\gamma^{2a,s}$ ,  $\gamma^{2b,s}$ ,  $\eta^{2a,s}$ ,  $\eta^{2b,s}$ ,  $\alpha^{2a,s}$ ,  $\alpha^{2b,s}$ ,  $\beta^{2a,s}$ , and  $\beta^{2b,s}$  are the stepsizes at iteration  $s$ , while  $\mathbf{x}^{1,s}(\xi_2)$ ,  $\mathbf{y}^{1,s}(\xi_2)$ ,  $\mathbf{q}^{2,s}(\xi_2)$ ,  $\mathbf{p}^{2,s}(\xi_2)$ ,  $\mathbf{x}^{2,s}(\xi_2)$ , and  $\mathbf{y}^{2,s}(\xi_2)$  are the optimal solutions for the decomposed-by-scenario problems of (79)-(91). The initial values of  $\lambda^{1,0}$ ,  $\theta^{1,0}$ ,  $\pi^{2a,0}$ ,  $\pi^{2b,0}$ ,  $\mu^{2a,0}$ ,  $\mu^{2b,0}$ ,  $\lambda^{2a,0}$ ,  $\lambda^{2b,0}$ ,  $\theta^{2a,0}$ , and  $\theta^{2b,0}$  are set to be zeros.

As we did for the two-stage cases, the stepsize rules of (103)-(112) are used. Note that  $\kappa^1$ ,  $\nu^1$ ,  $\tau^{2a}$ ,  $\tau^{2b}$ ,  $\nu^{2a}$ ,  $\nu^{2b}$ ,  $\kappa^{2a}$ ,  $\kappa^{2b}$ ,  $\nu^{2a}$ , and  $\nu^{2b}$  are small numbers so that series

of (103)-(112) converge.

$$\alpha^{1,0} \in \mathbb{R}^+, \quad \alpha^{1,k+1} = \frac{\alpha^{1,k}}{\kappa^1} \quad (103)$$

$$\beta^{1,0} \in \mathbb{R}^+, \quad \beta^{1,k+1} = \frac{\beta^{1,k}}{\nu^1} \quad (104)$$

$$\gamma^{2a,0} \in \mathbb{R}^+, \quad \gamma^{2a,k+1} = \frac{\gamma^{2a,k}}{\tau^{2a}} \quad (105)$$

$$\gamma^{2b,0} \in \mathbb{R}^+, \quad \gamma^{2b,k+1} = \frac{\gamma^{2b,k}}{\tau^{2b}} \quad (106)$$

$$\eta^{2a,0} \in \mathbb{R}^+, \quad \eta^{2a,k+1} = \frac{\eta^{2a,k}}{\nu^{2a}} \quad (107)$$

$$\eta^{2b,0} \in \mathbb{R}^+, \quad \eta^{2b,k+1} = \frac{\eta^{2b,k}}{\nu^{2b}} \quad (108)$$

$$\alpha^{2a,0} \in \mathbb{R}^+, \quad \alpha^{2a,k+1} = \frac{\alpha^{2a,k}}{\kappa^{2a}} \quad (109)$$

$$\alpha^{2b,0} \in \mathbb{R}^+, \quad \alpha^{2b,k+1} = \frac{\alpha^{2b,k}}{\kappa^{2b}} \quad (110)$$

$$\beta^{2a,0} \in \mathbb{R}^+, \quad \beta^{2a,k+1} = \frac{\beta^{2a,k}}{\nu^{2a}} \quad (111)$$

$$\beta^{2b,0} \in \mathbb{R}^+, \quad \beta^{2b,k+1} = \frac{\beta^{2b,k}}{\nu^{2b}} \quad (112)$$

The number of combinations to set the initial values of stepsizes and the denominators to update the stepsizes are huge. Therefore, in the implementations of subgradient method, all the initial stepsizes are set the same and the denominators are set to a single number. It is reasonable to do this since our goal is to obtain a good lower bound in order to justify the rolling horizon method.

Table 30 lists the different sets of parameters tested for the subgradient method for 22-flight case running for 6,000 iterations. In order to obtain good lower bounds for the three-stage stochastic programming model, the selection of parameters is very important. Large denominators for updating the stepsizes can result faster convergence but possibly to a bad lower bound. Moreover, large stepsizes do not guarantee good lower bounds. Take Figure 21 for example, it is worthy noting that the 0.01-curve produces better lower bound in the beginning, but 0.09-curve outperforms the 0.01-curve later on while producing better overall lower bound, which approaches the

optimal value.

**Table 30:** List of different parameters in subgradient method for three-stage model, flight count=22

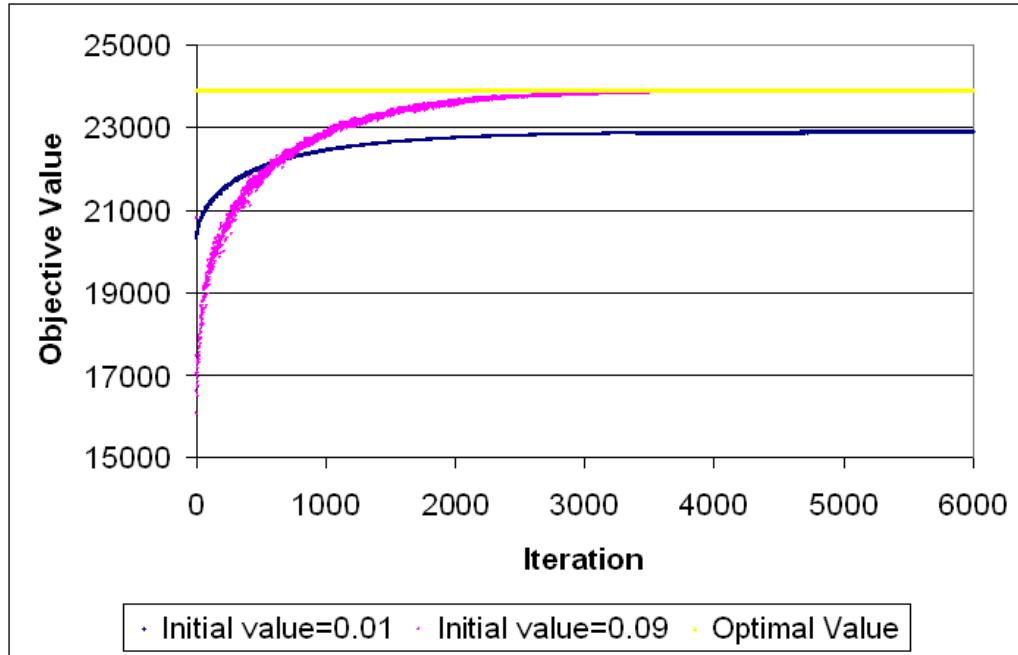
stepsizes	denominators	lower bound
0.01	1.001	22,897.19
0.05	1.001	23,851.87
0.10	1.001	23,907.96
0.15	1.001	23,905.93
0.20	1.001	23,903.51
0.30	1.001	23,899.04
0.40	1.001	23,893.78
0.50	1.001	23,889.69
1.00	1.001	23,867.60

The subgradient method implementations with stepsize rules (103)-(112) are presented in Table 31, with optimal objective values directly from three-stage model and the solutions derived from rolling horizon method for comparison.

**Table 31:** Computational comparison among optimality, rolling horizon method, and Lagrangian relaxation for three-stage model

flight count	OPT value	RHM (UB)	LR (LB)	% from optimal/UB
22	23,916.18	23,916.18	23,908.31	0.03%
27	29,402.30	29,402.30	29,389.69	0.04%
28	29,404.99	29,404.99	29,391.43	0.05%
31	30,008.38	30,011.09	29,979.71	0.10%
35	30,050.45	30,054.24	29,970.43	0.27%
44	n/a	30,663.20	30,490.59	0.56%

The subgradient method produces good lower bounds for each case, but it is time-consuming to obtain these bounds. The length of the computational time heavily depends on the number of iterations we run - longer time produces better bounds.



**Figure 21:** Comparison of paths between two parameters for three-stage model, flight count=22

The computational time spent to obtain the lower bounds shown in Table 31 are presented in Table 32.

**Table 32:** Computational time of the subgradient method for three-stage model

flight count	number of scenario	# of iteration	total time (sec)	second/iteration	second /scenario
22	2,048	6,000	27,844	4.64	0.002266
27	2,048	6,000	31,687	5.28	0.002579
28	4,096	6,000	65,186	10.86	0.002652
31	16,384	6,000	300,277	50.05	0.003055
35	32,768	6,000	735,994	122.67	0.003743
44	32,768	6,000	916,770	152.80	0.004663

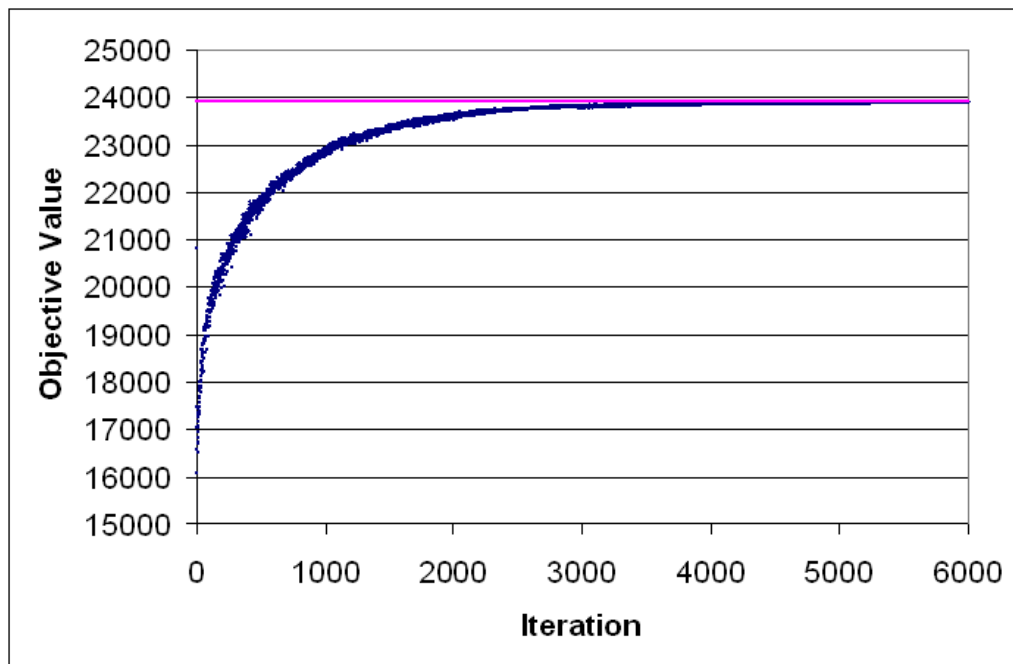
Table 33 lists the parameters that produce the best lower bounds for the three-stage model. We also plot the lower bound calculation paths for subgradient method associated with different flight counts in Figure 22-27. Note that except for forty-four-flight case that the horizontal line is depicted by the objective produced by the



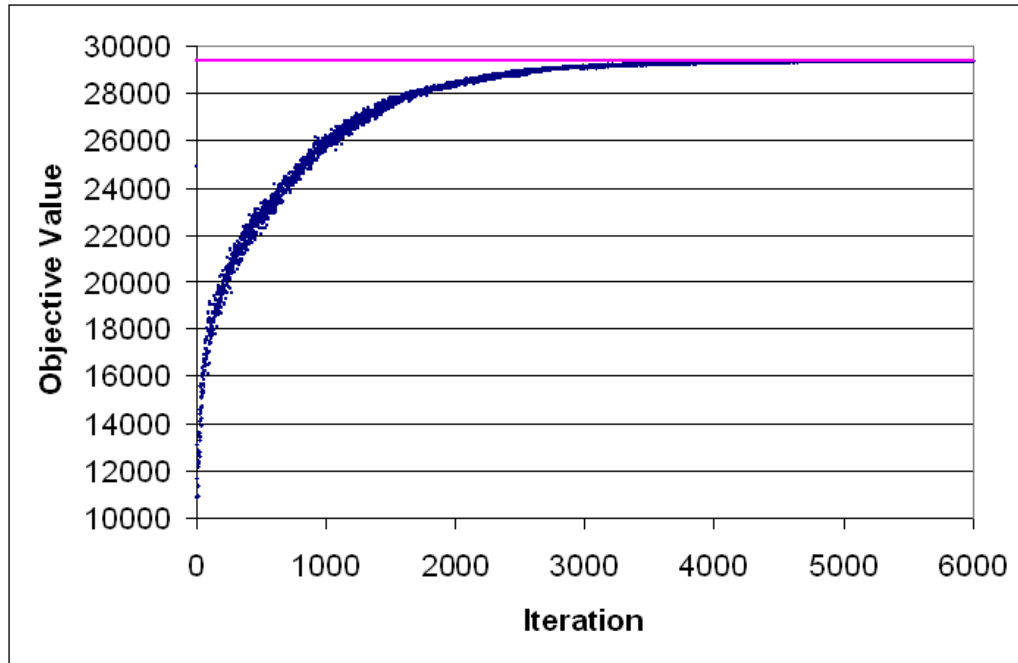
rolling horizon method, other cases are illustrated by the optimal objective value.

**Table 33:** List of parameters in subgradient method producing the best lower bounds for three-stage model

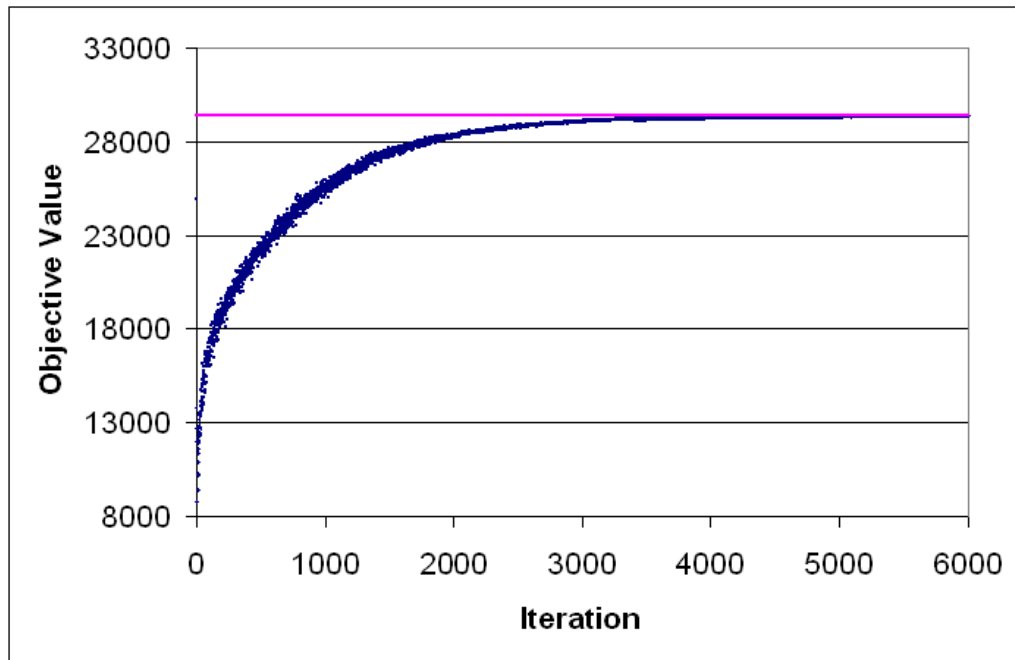
flight count	stepsizes	denominators
22	0.09	1.001
27	0.22	1.001
28	0.11	1.001
31	0.03	1.001
35	0.02	1.001
44	0.02	1.001



**Figure 22:** Path of subgradient method for three-stage model, flight count=22



**Figure 23:** Path of subgradient method for three-stage model, flight count=27



**Figure 24:** Path of subgradient method for three-stage model, flight count=28

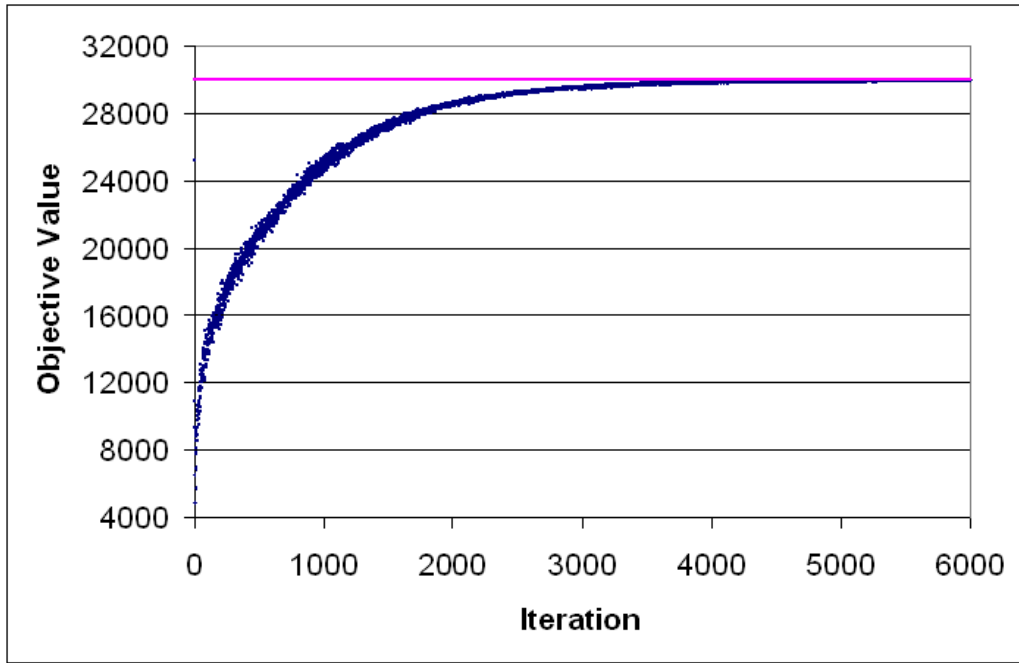


Figure 25: Path of subgradient method for three-stage model, flight count=31

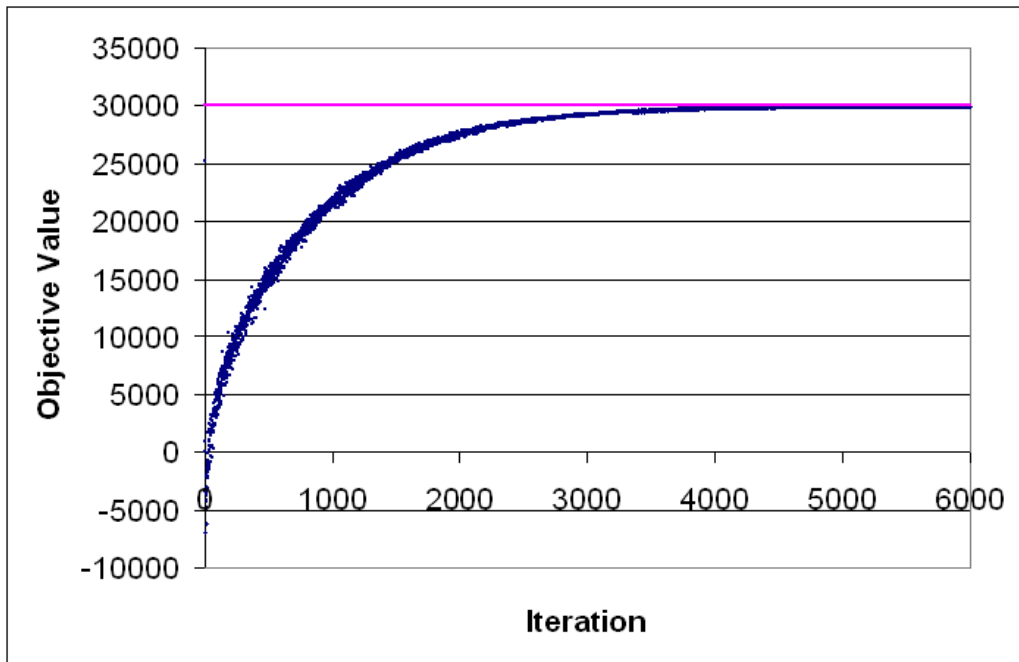
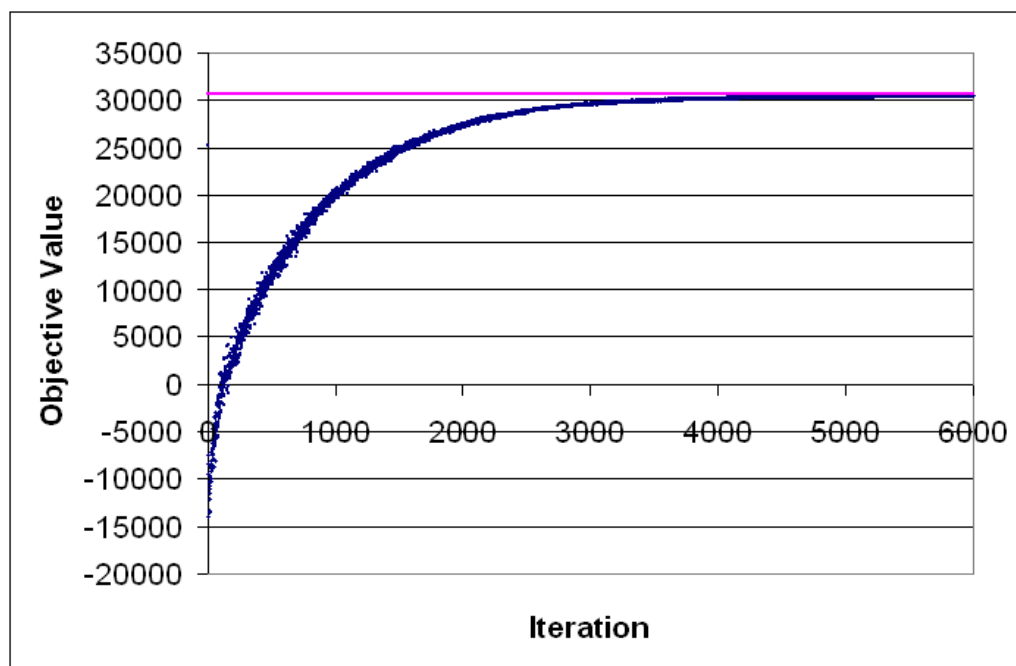


Figure 26: Path of subgradient method for three-stage model, flight count=35



**Figure 27:** Path of subgradient method for three-stage model, flight count=44

## CHAPTER V

### CONCLUSION AND FUTURE WORK

In this study, we have introduced stochastic programming approaches to the air traffic flow management problems. As uncertainties involve, the question as how aircraft are sent toward a sector and what their cruise speeds are during the inclement weather has been studied through two- and three-stage stochastic integer programming models. To the best of our knowledge, this problem has not yet been extensively studied and the models described in this thesis are the most comprehensive ones that involve several flight activities during inclement weather. In this chapter, we draw the conclusions and contributions of this study, and discuss the possible extensions in the area of air traffic flow management.

#### *5.1 Conclusion*

Stochastic programming technique has been applied to various areas when uncertainties involve. Likewise, in the area of traffic flow management while weather is a major concern, stochastic programs are proposed to approach the problems. In this study, we considered the only limiting resource in the airspace. For analysis purpose, we then divide the airspace into airspace sectors so that we can focus on a particular sector with aircraft coming from nationwide airports. We first introduce a two-stage stochastic programming model so that the ground-delay, cancellation, and the cruise speed decisions are made in the first stage, while air-holding and diversion decisions are made after the weather realizations in the second-stage. Different from studies in the ground holding program or other more general efforts in the air traffic flow management problems that dealt with only ground-delay and air-holding, our model is the most comprehensive one that handles additional cancellation, cruise speed, and

diversion decisions.

Due to the intractability of the presented stochastic program, a heuristic we referred as "rolling horizon method" has been introduced to approach the model. The rolling horizon method solves the presented two-stage model in a rolling fashion so that all the involved flights are considered at least once. If some flights are considered more than once, then the most present decisions dominate. We find that the rolling horizon method provides near-optimal solutions as the optimization model does, and we justify the rolling horizon method by testing flight schedule with ten sets of different parameters. The performance of this heuristic is excellent in solution quality as well as in the computational time. Within one minute, the rolling horizon method can solve the optimization model, which can take up to one hour to solve.

As the optimization model can not always solve the problem due to computational issues, we are not able to judge the performance of the rolling horizon method in some cases. We then apply the Lagrangian relaxation techniques so that the nonanticipativity constraints are relaxed in Lagrangian fashion. We implement the Lagrangian relaxation by subgradient method. In the two-stage stochastic program, the subgradient method provides nice lower bounds, which are within 0.01% of the optimal value produced by the optimization model. For the case which the optimal value is not available, the gap between the upper bound produces by the rolling horizon method and the lower bound generated by the subgradient method is only 0.44%. Therefore, we believe that the success of the rolling horizon method can lead it to be applied to dynamic solution procedures which generate great decisions in terms of minimal costs in real time.

Multi-stage stochastic programs can easily be extended by two-stage stochastic models. However, the number of decision variables and constraints can be huge if the number of stages increase which will cause long computational time. As a result, we start the multi-stage models by looking at three stages.

A three-stage stochastic programming model has additional decision point compared with a two-stage stochastic program. The additional decision point at time  $k$  is constructed between current time we run the model (time zero) and future time the flights are scheduled to enter the sector. As time evolves, the weather forecast is more accurate at time  $k$  compared with time zero. The decisions can depend on weather forecast at either time zero or time  $k$ . Therefore, the additional decision point at time  $k$  provides opportunities to see future weather more clearly. The three-stage model is constructed so that the ground-delay, cancellation, and speed-change decisions are made before any realizations in the first-stage. Then in the second-stage, after the first weather realization, the recourse actions including diversion and air-holding decisions corresponding to the first-stage decisions, as well as the ground-delay, cancellation, and speed-change decisions for those flights that are waiting to see the updated weather forecast, are made. Finally, after the second weather realization, the diversion and air-holding decisions corresponding to the second-stage departure decisions are made in the third stage.

Solving the three-stage model to optimality takes more time than the two-stage program because it generates more than doubled decision variables and constraints. Nevertheless, it provides better solutions in terms of lower costs. We see the benefits by adding the additional stage, however, to leverage between better decisions and the computational time is an important issue.

Since the rolling horizon method performs well for the two-stage model, we test it to the three-stage stochastic program as well. Although the computational time is longer that is reasonable because the number of variables and constraints increase, the rolling horizon method does provide near-optimal solutions within 1.5 minutes.

For the case that the optimization model is not able to provide optimal solution, we can not see how rolling horizon method performs. Therefore, we apply the Lagrangian relaxation techniques on the nonanticipativity constraints. Implemented by

subgradient method, lower bounds are obtained within 0.27% of optimal. For the case which the optimal value is not available, the gap between the upper bound produces by the rolling horizon method and the lower bound generated by the subgradient method is only 0.56%.

To conclude, the rolling horizon method can be applied to either two- or three-stage stochastic integer programming models and the method can produce near-optimal solutions within reasonable computational time. The performance of the rolling horizon method is excellent so that we can apply it as a dynamic algorithm which can deal with the problem as how aircraft are sent towards an airspace sector under inclement weather, in real time.

## ***5.2 Future Work***

Several possible extensions of our models can be made. In this study, we focus on the sector capacities under the inclement weather. As studied in the ground holding program, airport capacities can also be one of the limiting resources. Incorporating airport capacities into the model can make the model even more comprehensive. Another possibility is to extend the model from single-sector to multi-sectors or NAS-wide which can be challenging as inter-sector activities need to be included. Extension to four-stage, five-stage, and so on is also possible. The overall objective value can be expected to be better (lower) than the two- and three- stage models presented in this study. However, as the number of stages increase, the problem will become even more intractable.

Reduction of the number of scenarios can be a reasonable modification. In our testing example, we assume that the number of weather types is two at each period. Weather can be so bad that the sector capacity in a particular period is determined to be zero. Under this situation, we do not need to distinguish good and bad weather, which can decrease the number of scenarios by a factor of two. If "G" and "B"



represent good and bad weather, respectively, weather forecast such as "G-B-G-B-G-B-G-B" in some consecutive periods is unlikely. The purpose to reduce the number of scenarios is to shrink the problem size and therefore decrease the computational time.

Another issue can be the justification of our presented two- and three-stage models. As described in the content of this study, the sector capacities are randomly generated data. In the near future as the sector information is available to public in a readable form, testing models by the real sector capacities can make the construction of mathematical programming models more valuable as a comparison between the objective value from a theoretical model and the real-incurred costs from a practical experience can become available.

## REFERENCES

- [1] “Bureau of transportation statistics.” <http://www.transtats.bts.gov/>.
- [2] “Howstuffworks.” <http://www.howstuffworks.com/air-traffic-control.htm>.
- [3] “Evaluating the true cost to airlines of one minute of airborne or ground delay,” tech. rep., Transport Studies Group, University of Westminster, April 2004.
- [4] AHMED, S., KING, A. J., and PARIJA, G., “A multi-stage stochastic integer programming approach for capacity expansion under uncertainty,” *Journal of Global Optimization*, vol. 26, pp. 3–24, 2003.
- [5] AHMED, S., TAWARMALANI, M., and SAHINIDIS, N. V., “A finite branch-and-bound algorithm for two-stage stochastic integer programs,” *Mathematical Programming*, vol. 100, pp. 355–377, 2004.
- [6] ALONSO, A., ESCUDERO, L. F., and ORTUÑO, M. T., “A stochastic 0-1 program based approach for the air traffic flow management problem,” *European Journal of Operational Research*, vol. 120, pp. 47–62, 2000.
- [7] ANDREATTA, G. and ROMANIN-JACUR, G., “Aircraft flow management under congestion,” *Transportation Science*, vol. 21, no. 4, pp. 249–253, 1987.
- [8] BALL, M. O., HOFFMAN, R., ODONI, A. R., and RIFKIN, R., “A stochastic integer program with dual network structure and its application to the ground-holding problem,” *Operations Research*, vol. 51, no. 1, pp. 167–171, 2003.
- [9] BARNIER, N. and BRISSET, P., “Graph coloring for air traffic flow management,” *Annals of operations research*, vol. 130, pp. 163–178, 2004.
- [10] BAYEN, A. M., GRIEDER, P., MEYER, G., and TOMLIN, C. J., “Lagrangian delay predictive model for sector-based air traffic flow,” *Journal of guidance, control, and dynamics*, vol. 28, no. 5, pp. 1015–1026, 2005.
- [11] BERTSIMAS, D., LULLI, G., and ODONI, A., “The air traffic flow management problem: An integer optimization approach,” *Lecture Notes In Computer Science*, vol. 5035, pp. 34–46, 2008.
- [12] BERTSIMAS, D. and PATTERSON, S. S., “The air traffic flow management problem with enroute capacities,” *Operations Research*, vol. 46, no. 3, pp. 406–422, 1998.
- [13] BERTSIMAS, D. and PATTERSON, S. S., “The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach,” *Transportation Science*, vol. 34, no. 3, pp. 239–255, 2000.

- [14] BERTSIMAS, D. and SHIODA, R., “Restaurant revenue management,” *Operations Research*, vol. 51, no. 3, pp. 472–486, 2003.
- [15] BERTSIMAS, D. and TSITSIKLIS, J. N., *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [16] BIRGE, J. R., “Decomposition and partitioning methods for multistage stochastic linear programs,” *Operations Research*, vol. 33, no. 5, pp. 989–1007, 1985.
- [17] BIRGE, J. R. and LOUVEAUX, F. V., “A multicut algorithm for two-stage stochastic linear programs,” *European Journal of Operational Research*, vol. 34, pp. 384–392, 1988.
- [18] BIRGE, J. R. and LOUVEAUX, F. V., *Introduction to stochastic programming*. Springer, 1997.
- [19] CARØE, C. C. and SCHULTZ, R., “Dual decomposition in stochastic integer programming,” *Operations Research Letter*, vol. 24, pp. 37–45, 1999.
- [20] CARØE, C. C. and TIND, J., “L-shaped decomposition of two-stage stochastic programs with integer recourse,” *Mathematical Programming*, vol. 83, pp. 451–464, 1998.
- [21] CASTRO, J., “A stochastic programming approach to cash management in banking,” *European Journal of Operational Research*, vol. 192, pp. 963–974, 2009.
- [22] CHEN, N. and SRIDHAR, B., “Estimation of air traffic delay using three dimensional weather information,” in *The 26th Congress of ICAS and 8th AIAA ATIO*, 2008.
- [23] DANTZIG, G. B., “Linear programming under uncertainty,” *Management Science*, vol. 1, no. 3-4, pp. 197–206, 1955.
- [24] EVANS, J., ROBINSON, M., CHO, J., and EVANS, M., “An integrated strategic-tactical approach to managing traffic flows to the northeast during severe convective weather.” Technical interchange meeting between Georgia Tech Air Transportation Laboratory and MIT Lincoln Laboratory, 2010.
- [25] FISHER, M. L., “The lagrangian relaxation method for solving integer programming problems,” *Management Science*, vol. 27, no. 1, pp. 1–18, 1981.
- [26] HELME, M. P., “Reducing air traffic delay in a space-time network,” in *IEEE international conference on systems, man, and cybernetics*, pp. 236–242, 1992.
- [27] HELME, M. P. and LINDSAY, K., “Optimization of traffic flow to minimize delay in the national airspace system,” in *IEEE conference on control application*, pp. 435–437, 1992.
- [28] HIGLE, J. L., “Stochastic programming: Optimization when uncertainty matters,” tech. rep., *Tutorials in Operations Research*, 2005.

- [29] HIGLE, J. L. and SEN, S., “Stochastic decomposition: an algorithm for two-stage linear programs with recourse,” *Mathematics of operations research*, vol. 16, no. 3, pp. 650–669, 1991.
- [30] HIGLE, J. L. and SEN, S., “Statistical approximations for stochastic linear programming problems,” *Annals of operations research*, vol. 85, pp. 173–192, 1999.
- [31] HOFFMAN, R., BURKE, J., LEWIS, T., FUTER, A., and BALL, M., “Resource allocation principles for airspace flow control,” in *AIAA Guidance, navigation and control conference and exhibit*, 2005.
- [32] HOFFMAN, R., KROZEL, J., DAVIDSON, G., and KIERSTEAD, D., “Probabilistic scenario-based event planning for traffic flow management,” in *AIAA Guidance, navigation and control conference and exhibit*, 2007.
- [33] HOFFMAN, R. L., *Integer programming models for ground-holding in air traffic flow management*. PhD thesis, University of Maryland at College Park, 1997.
- [34] HUANG, K. and AHMED, S., “The value of multistage stochastic programming in capacity planning under uncertainty,” *Operations Research*, vol. 57, no. 4, pp. 893–904, 2009.
- [35] HUNTER, G. and RAMAMOORTHY, K., “National airspace strategies for future demand scenarios in inclement weather,” in *24th Digital Avionics Systems Conference*, 2005.
- [36] KLEYWEGT, A. J., SHAPIRO, A., and DE MELLO, T. H., “The sample average approximation method for stochastic discrete optimization,” *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 479–502, 2002.
- [37] LAPORTE, G. and LOUVEAUX, F. V., “The integer l-shaped method for stochastic integer programs with complete recourse,” *Operations Research letter*, vol. 13, pp. 133–142, 1993.
- [38] LINDEROTH, J., SHAPIRO, A., and WRIGHT, S., “The empirical behavior of sampling methods for stochastic programming,” *Annals of Operations Research*, vol. 142, pp. 215–241, 2006.
- [39] LISTES, O. and DEKKER, R., “A scenario aggregation-based approach for determining a robust airline fleet composition for dynamic capacity allocation,” *Transportation Science*, vol. 39, no. 3, pp. 367–382, 2003.
- [40] LIU, P.-C. B., HANSEN, M., and MUKHERJEE, A., “Scenario-based air traffic flow management: From theory to practice,” *Transportation Research part B*, vol. 42, pp. 685–702, 2008.

- [41] LIUM, A.-G., CRAINIC, T. G., and WALLACE, S. W., “A study of demand stochasticity in service network design,” *Transportation Science*, vol. 43, no. 2, pp. 144–157, 2009.
- [42] LULLI, G. and ODONI, A., “The european air traffic flow management problem,” *Transportation Science*, vol. 41, no. 4, pp. 431–443, 2007.
- [43] MAATMAN, A., SCHWEIGMAN, C., RUIJS, A., and VAN DER VLERK, M. H., “Modeling farmers’ response to uncertain rainfall in Burkina Faso: a stochastic programming approach,” *Operations Research*, vol. 50, no. 3, pp. 399–414, 2002.
- [44] MARTIN, B. D., EVANS, J., and DELAURA, R., “Exploration of a model relating route availability in en route airspace to actual weather coverage parameters,” in *12th Conference on Aviation Range and Aerospace Meteorology*, 2006.
- [45] MUKHERJEE, A., *Dynamic Stochastic Optimization Models for Air Traffic Flow Management*. PhD thesis, University of California, Berkeley, 2004.
- [46] MUKHERJEE, A. and HANSEN, M., “A dynamic stochastic model for the single airport ground holding problem,” *Transportation Science*, vol. 41, no. 4, pp. 444–456, 2007.
- [47] MUKHERJEE, A. and HANSEN, M., “A dynamic rerouting model for air traffic flow management,” *Transportation Research Part B*, vol. 43, pp. 159–171, 2009.
- [48] MULVEY, J. M. and RUSZCZYŃSKI, A., “A new scenario decomposition method for large-scale stochastic optimization,” *Operations Research*, vol. 43, no. 3, pp. 477–490, 1995.
- [49] NAVAZIO, L. and ROMANIN-JACUR, G., “The multiple connections multi-airport ground holding problem: models and algorithms,” *Transportation Science*, vol. 32, no. 3, pp. 268–276, 1998.
- [50] NILIM, A., EL GHAOUI, L., and DUONG, V., “Algorithms for multi-aircraft routing under uncertainty,” *RIVF*, pp. 21–32, 2004.
- [51] NORKIN, V. I., PFLUG, G. C., and RUSZCZYŃSKI, A., “A branch and bound method for stochastic global optimization,” *Mathematical Programming*, vol. 83, pp. 425–450, 1998.
- [52] ODONI, A. R., “The flow management problem in air traffic control,” in *Flow control of congested networks* (ODONI, A. R., BIANCO, L., and SZEGÖ, G., eds.), pp. 269–288, Springer Verlag, 1987.
- [53] OUSSEDIK, S. and DELAHAYE, D., “Reduction of air traffic congestion by genetic algorithms,” *Lecture Notes In Computer Science*, vol. 1498, pp. 855–864, 1998.

- [54] PENNANEN, T. and KALLIO, M., “A splitting method for stochastic programs,” *Annals of Operations Research*, vol. 142, pp. 259–268, 2006.
- [55] RICHETTA, O. and ODONI, A. R., “Solving optimally the static ground-holding policy problem in air traffic control,” *Operations Research Society of America*, vol. 27, no. 3, pp. 228–238, 1993.
- [56] RICHETTA, O. and ODONI, A. R., “Dynamic solution to the ground-holding problem in air traffic control,” *Transportation Research -A*, vol. 28, no. 3, pp. 167–185, 1994.
- [57] RIKFIN, R. M., “The single airport static stochastic ground holding problem,” Master’s thesis, Massachusetts Institute of technology, 1994.
- [58] ROBELIN, C.-A., SUN, D., WU, G., and BAYEN, A. M., “MILP control of aggregate eulerian network airspace models,” *American Control Conference*, June 14-16 2006.
- [59] ROCKAFELLAR, R. T. and WETS, R. J.-B., “Scenarios and policy aggregation in optimization under uncertainty,” *Mathematics of Operations Research*, vol. 16, no. 1, pp. 119–147, 1991.
- [60] ROSA, C. H. and RUSZCZYŃSKI, A., “On augmented lagrangian decomposition methods for multistage stochastic programs,” *Annals of Operations Research*, vol. 64, pp. 289–309, 1996.
- [61] SANTOSO, T., AHMED, S., GOETSCHALCKX, M., and SHAPIRO, A., “A stochastic programming approach for supply chain network design under uncertainty,” *European Journal of Operational Research*, vol. 167, pp. 96–115, 2005.
- [62] SCHULTZ, R., “Stochastic programming with integer variables,” *Mathematical Programming*, vol. 97, pp. 285–309, 2003.
- [63] SCHULTZ, R., STOUGIE, L., and VAN DER VLERK, M., “Solving stochastic programs with integer recourse by enumeration: A framework using gröbner basis reductions,” *Mathematical Programming*, vol. 83, pp. 229–252, 1998.
- [64] SEN, S. and SHERALI, H. D., “Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming,” *Mathematical Programming*, vol. 106, pp. 203–223, 2006.
- [65] SHAPIRO, A., “Inference of statistical bounds for multistage stochastic programming problems,” *Mathematical Methods of Operations Research*, vol. 58, pp. 57–68, 2003.
- [66] TERRAB, M. and ODONI, A. R., “Strategic flow management for air traffic control,” *Operations Research*, vol. 41, no. 1, pp. 138–152, 1993.

- [67] VAN DEN AKKER, J., VAN KEMENADE, C., , and KOK, J., “Evolutionary 3D-air traffic flow management,” in *Handbook of evolutionary computation*, 1997.
- [68] VAN KEMENADE, C., VAN DEN AKKER, J., and KOK, J., “Evolutionary air traffic flow management for large 3D-problems,” tech. rep., Parallel Problem solving from nature IV, 1996.
- [69] VAN SLYKE, R. and WETS, R., “L-shaped linear programs with applications to optimal control and stochastic programming,” *SIAM Journal on Applied Mathematics*, vol. 17, no. 4, pp. 638–663, 1969.
- [70] VRANAS, P. B., BERTSIMAS, D. J., and ODONI, A. R., “Dynamic ground-holding policies for a network of airports,” *Transportation Science*, vol. 28, no. 4, pp. 275–291, 1994.
- [71] VRANAS, P. B., BERTSIMAS, D. J., and ODONI, A. R., “The multi-airport ground-holding problem in air traffic control,” *Operations Research*, vol. 42, no. 2, pp. 249–261, 1994.
- [72] WANG, H., “A dynamic programming framework for the global flow control problem in air traffic management,” *Transportation Science*, vol. 25, no. 4, pp. 308–313, 1991.
- [73] YEN, J. W. and BIRGE, J. R., “A stochastic programming approach to the airline crew scheduling problem,” *Transportation Science*, vol. 40, no. 1, pp. 3–14, 2006.
- [74] ZOBELL, S., WANKE, C., and SONG, L., “Probabilistic airspace congestion management,” in *12th Conference on Aviation Range and Aerospace Meteorology*, 2006.