ROBUST AIRLINE FLEET ASSIGNMENT

by

VENKATA L. PILLA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2006

To my Mother, Father, Brother and Sister.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere thanks to my advisors Dr.Victoria C.P.Chen and Dr.Jay M.Rosenberger for introducing me to this interesting research topic and patiently guiding me throughout the course of the research. They were always available to answer my queries either in person or through mail. The discussions I had with them provided a good insight into the research, improved my ability to think independently, and always kept me motivated to work. I would like to thank Dr.H.W.Corley, Dr.Jamie K.Rogers and Dr.Eli V.Olinick for graciously agreeing to be on my committee and giving some good input toward this research. I will never forget the classes I took under Dr.Chen, Dr.Rosenberger and Dr.Corley, which laid the foundation for me to pursue my research in the fields of statistics and optimization. I would also like to thank Dr.Barry Smith from Sabre-Holdings for providing the data and answering questions related to my thesis.

I would also like to extend my appreciation to the Graduate School, for providing financial support through a TPEG scholarship to pursue my Ph.D. program, and Dr.Chen, for her support financially for couple of semesters. I am grateful to Robin Melton in SOAR Learning Services for giving me a chance to be a part of great group of Supplemental Instruction to help students at UTA and for providing me extra support and ease my financial burden. The appreciations shown by the students and the immense satisfaction I received during teaching are priceless. I would also like to thank Shelly Hamilton (lecturer in Mathematics Department) for helping me deal with the pressure due to research and working at SOAR.

I would like to thank Sheela Siddappa, whose help and valuable suggestions towards my research and personal life are inestimable. I also thank my friends and COSMOS lab members Duraikannan Sundaramoorthi, Aihong Wen, Prattana Punnakitikashem (Sandy), Prashant Kumar Tarun, Heesu Hwang (Peter), Tai-kuan Sung, Dachuan Shih (Thomas), Siriwat Visoldilokpun (Pop), Panita Suebvisai, Jawahar Veera, and Huiyuan Fan for their help and making COSMOS lab a wonderful place to do research. Though each of them had their own problems, they never held themselves back to lift another person's spirits to do research. I am lucky to have such good friends.

Last but not the least, I am indebted to my Parents (Sri. P. Radhakrishna Rao and Smt. P. Suneeta), Brother (Pawan Kumar) and Sister (Varaha Lakshmi) and my relatives for their love, compassion, motivation, moral support and blessings without which none of this was possible. I would like to thank Kumar Mahesh Pilla, Ameeta Kishan Hemmadi, and Tamal Kishore Pilla, who never made me feel that I missed my family. The hard work and patience involved in pursuing this degree are nothing compared to the sacrifices made by my family, and I dedicate this to them.

May 17, 2006

ABSTRACT

ROBUST AIRLINE FLEET ASSIGNMENT

Publication No. ______

VENKATA L. PILLA, Ph.D.

The University of Texas at Arlington, 2006

Supervising Professor: Victoria C. P. Chen, Jay M. Rosenberger

The fleet assignment model allocates the fleet of aircraft types to the scheduled flight legs in the airline timetable published 90 days prior to the departure of the aircraft. The objective is to maximize profit. While costs associated with assigning a particular fleet type to a leg are easy to estimate, the revenues are based upon demand, which is realized close to departure. The uncertainty in demand makes it challenging to assign the right type of aircraft to each flight leg based on the forecasts taken 90 days prior to departure.

Therefore, in this dissertation a two-stage stochastic programming framework has been utilized to model the uncertainty in demand, along with the Boeing concept of demand driven dispatch to reallocate aircraft close to departure of the aircraft. In this method, crew-compatible families are allocated in the first stage (90 days prior to departure of aircraft), and in the second stage (two weeks prior to departure), when most of the demand is realized, specific fleet types within the families are assigned. The stochasticity of the demand is incorporated by considering different demand scenarios in the second stage, and an average over the scenarios is used to calculate the expected profit.

Traditionally, the two-stage stochastic programming framework problems are solved using a Benders' approach. Due to the slow convergence of the Benders' approach a novel statistics-based approach using design and analysis of computer experiments has been developed. The results obtained with our approach are compared to that of a Benders' approach. Finally future research is discussed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

As airline woes continue to persist, mainly due to rising fuel costs, major domestic airlines are battling to reduce costs, increase load factors, and improve revenues. The International Air Transport Association reports that North American carriers are expected to loose $8 billion dollars in the year 2005 [1]. One way to reduce costs and increase revenue is to strike a balance between supply (seats) and demand (passengers). Utilizing a smaller capacity aircraft would result in spill due to seat shortage, and usage of a larger capacity aircraft would fly empty seats. To avoid this, airline Fleet Assignment Models (FAMs) are used to assign aircraft to the scheduled flights in order to maximize profit (revenue - cost). FAMs have been credited with an annual savings of $100 million at Delta [2], $15 million annually at USAirways [3], and a 1.4% improvement in operating margins at American Airlines [4].

The quality of a FAM solution depends upon the accuracy of cost and revenue estimates. While cost estimates are relatively stable and well known, revenue estimates depend on demand forecasts. Other than the seasonal variations, airline passenger demand has been affected due to various reasons like the terrorist attacks on September 11, 2001, the outbreak of SARS, and recently rising fuel costs, forcing the airlines to increase fares and indirectly affecting demand [5]. This variability in demand makes it challenging to assign the right type of aircraft to each flight leg in the schedule, which is published 90 days prior to the departure of the flight. Consequently, modeling demand stochastically and delaying the fleet assignment decision closer to departure would likely improve profit.

In this regard, stochastic programming and the Boeing concept of Demand Driven Dispatch are very useful. A brief introduction to these two concepts are given in Sections 1.1 and 1.2.

1.1 Demand Driven Dispatch

Since most demand for flights is realized after the schedule is published, a *robust* FAM approach would include reallocation of aircraft much closer to departure. Berge and Hooperstad [6] introduced the concept of Demand Driven Dispatch (D^3), in order to match demand to aircraft close to the departure of the flight. In this approach, they use two stages of decision-making. The first stage occurs 90 days prior to the departure of the flight, when the flight schedules are published. During this stage crew-compatible families of aircraft are assigned to flights in the airline timetable. Two aircraft are said to be *crew-compatible*, if they have the same cockpit model; hence, the same crew could operate either aircraft type. The second stage occurs two weeks prior to the departure of aircraft when most of the demand is realized and individual aircraft assignments within the crew-compatible families are done based on the demand. Swapping can essentially take place with the assignment of specific aircraft in the second stage. For example, Boeing 757 and 767 models are crew-compatible. Suppose flights A and B are initially allocated a 757 and 767, respectively as shown in Figure 1.1. If two weeks before departure, flight A has realized a higher demand than expected, while demand is lower than expected for flight B, then the airline can swap the 757 and 767 without affecting the crew schedule. Higher profit is achieved via this swapping since more revenue is captured, and the cost of swapping is usually insignificant for the airline.

Using this concept, Berge and Hooperstad have reported 1% - 5% improvement in profits. The difficulty in implementing a D^3-FAM approach is assigning aircraft types to flights such that swapping is feasible within the practical constraints of the airline

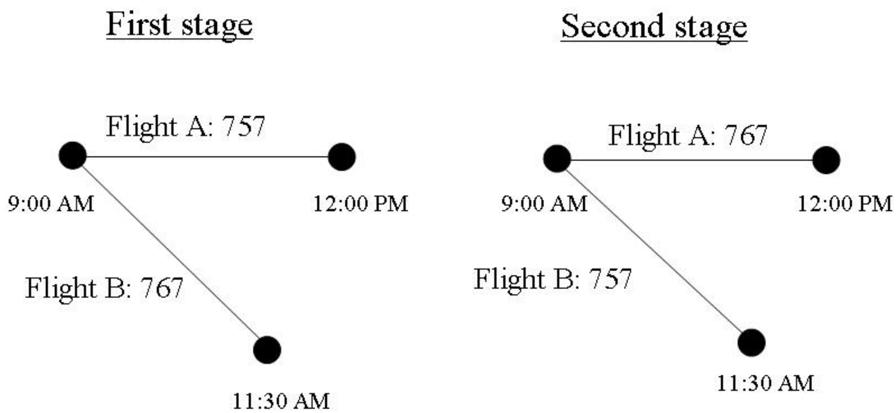

Figure 1.1. Demand driven dispatch example.

network. Most notably, two planes can only be swapped if they depart from the same airport and have similar departure times.

1.2 Stochastic Programming

The uncertainty in demand, which is realized close to departure of the aircraft, can be modeled using two-stage Stochastic Programming (SP). SP models use probability distributions for random events, and their goal is to maximize the expectation of some function of the decisions based on these random events. In two-stage SP, a decision is taken in the first stage, after which a random event with known probability distribution is realized, and a second-stage *recourse* decision is taken based on the random event and the decision from the first stage. The optimal decision consists of a single first-stage decision and a set of *recourse* decisions to be taken based on the random outcome.

As mentioned in Birge and Louveaux [7], the basic two-stage stochastic linear program with fixed recourse is given by:

$$\min z = c^T x + E_\xi \left[\min q(\omega)^T y(\omega)\right] \tag{1.1}$$

$$\text{s.t. } Ax = b, \tag{1.2}$$

$$T(\omega)x + Wy(\omega) = h(\omega), \tag{1.3}$$

$$x \geq 0, \quad y(\omega) \geq 0, \tag{1.4}$$

where $x \in R^{n_1}$ is the first-stage decision vector with linear costs $c \in R^{n_1}, y(\omega) \in R^{n_2}$ is the second-stage decision vector with linear costs $q(\omega) \in R^{n_2}, A$ is an m_1 x n_1 first-stage linear constraint matrix with right hand side $b \in R^{m_1}$ and $T(\omega)$ and W are, respectively m_2 x n_1 and m_2 x n_2 matrices specifying the second-stage linear constraints on x and y with right hand side $h(\omega) \in R^{m_2}$. For a given realization of the stochastic variables, $\omega \in \Omega$, the second-stage problem data, $q(\omega), h(\omega)$ and $T(\omega)$ become known, and the second-stage decision $y(\omega, x)$ can be obtained. Let Ξ denote a set of scenarios and the vector $\xi^T(\omega)$ represent a scenario with different components of the second stage, i.e., $q(\omega)^T, h(\omega)^T$ and $T(\omega)$, such that $\xi \in \Xi$. The objective function represented by (1.1) contains a deterministic term $c^T x$ and the expectation of the second-stage objective $q(\omega)^T y(\omega)$ taken over all realizations of the random event ω. For a given ω, we can write the second-stage value function as

$$Q(x, \xi(\omega)) = \min_y \{q(\omega)^T y | Wy = h(\omega) - T(\omega)x, y \geq 0\}. \tag{1.5}$$

Suppose, the expected second stage recourse function is defined as:

$$\Im(x) = E_\xi Q(x, \xi(\omega)), \tag{1.6}$$

then the deterministic equivalent of the two-stage stochastic linear program can be written as:

$$\begin{aligned} \min\ z &= c^T x + \Im(x) \\ \text{s.t. } Ax &= b, \\ x &\geq 0. \end{aligned} \tag{1.7}$$

One difficulty is determining the recourse function $\Im(x)$. If the stochastic variables are continuous, then a large number of scenarios may be needed to estimate the expectation of the second-stage value function $\Im(x)$. Thus, there is a high computational cost for evaluating $\Im(x)$ at just one value of x, making the problem defined in (1.7) even harder to solve. Consequently, the iterative approximation methods described by Birge and Louveaux [7] can be very slow to converge.

In order to control the computational requirement, Chen [8] proposed discretization of the x-space to a finite set of points and solving for $\Im(x)$ only at those points, followed by a function approximation technique to estimate the entire surface of $\Im(x)$. This approximation, $\hat{\Im}(x)$, will be computationally trivial to evaluate in (1.7). In selecting a discretization, it is important to select only those x values which result in a feasible solution in the second stage.

1.3 Contribution

The FAM problem is modeled in this dissertation using a two-stage SP framework. The two stages correspond to the stages of decision-making in the D^3 concept. Similar to Berge and Hooperstad [6], the FAM model presented here assigns crew-compatible aircraft to flights in the first stage. Given an initial Crew Compatible Allocation (CCA), the second stage obtains many equally likely scenarios using a probability distribution, and for each scenario, a combined FAM and passenger mix model, using the current

CCA, solves for the Linear Programming (LP) relaxation of assigning aircraft within the crew-compatible families. The estimate of the expected revenue is determined as the average over the scenarios. As mentioned in Section 1.2, the basic SP solution approach can be computationally intensive if an excessive number of scenarios is required for each initial CCA.

A computationally tractable solution method is presented in this dissertation by employing statistical methods from Design and Analysis of Computer Experiments (DACE) (see Chen et al [9], Sacks et al [10]). Our approach employs experimental design to generate the initial CCA (first-stage decision) and then for each CCA and for each scenario (demand realization) in the second stage, a *computer experiment* (in our case, an optimization problem) is solved. The average over the scenarios is used to estimate the expected revenue value of the recourse function for the two-stage SP model. A statistical model is fit to these data to provide a continuous approximation of expected revenue over the initial CCA space. In our case, the experimental design was based on a Latin hypercube, and the statistical model was based on Multivariate Adaptive Regression Splines (MARS). Assuming that the generated fit is concave, derivatives of the MARS approximation were used as optimality cuts to obtain a first-stage solution.

A Literature review regarding the FAM problem and its extensions is presented in Chapter 2. Chapter 3 discusses more on DACE and the steps taken to obtain a MARS approximation to the recourse function. Results of DACE step are included. The optimization of the two-stage FAM problem using MARS approximation of the recourse function and its comparison to traditional Benders' approach is presented in Chapter 4. Future work is presented in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

The fleet assignment problem has been a well researched topic for the past fifty years and has been credited for increasing profits at several airlines. Sherali et al [11], provided an extensive review of various fleet assignment models and algorithms. Ferguson and Dantzig formulated a combined FAM and aircraft routing problem with deterministic demand (1955) and later with stochastic demand (1956). After the deregulation of the US airline industry in 1978, major carriers increased their flight schedules significantly and developed hub-and-spoke networks. This allowed them to serve more destinations with a lot of traffic at the hubs making the fleet assignment problem much more complex to solve. Abara [4] published the first significant FAM application using an integer linear programming model. He defined *turn* as the successive assignment of an aircraft to two consecutive flights, and this decision variable causes practical limitations to the FAM problem. He first solved the FAM LP relaxation, fixed variables, and then solved the Mixed Integer Programming (MIP). To measure solution quality, Abara observed that the number of high demand legs that were covered by larger aircraft increased from 76% to 90% and increased operating margin by 1.4%.

Hane et al [12] presented a detailed description on solving a fleet assignment problem as a multi-commodity network flow problem, which formed the basis for a large portion of later FAM research. They modeled the FAM problem as an MIP model and formulated the constraints using a time line network for each airport and fleet type combination. Since these problems are often degenerate, they proposed different methods, which include an interior-point algorithm, dual steepest edge simplex, cost perturbation, model aggregation, branching on set-partitioning constraints, and prioritizing the order of branching, to reduce the time required to solve the problem. Some of the concepts addressed in Hane et al [12], like node aggregation and islands at stations with low activity, are employed in this dissertation to preprocess the FAM model before solving. Hane et al [12] reported run times twice as fast as the standard LP based branch and bound code.

Subramanian et al [2] implemented a coldstart model with many of the features described by Hane et al to solve the fleet assignment problem at Delta Airlines. They solved a daily fleet assignment problem using an MIP model, so as to minimize the combination of operating and passenger spill costs. The name *coldstart* was inspired by the fact that the optimization model did not require any initial fleeting and produced results based on a raw schedule. Talluri [13] presented algorithms to solve a warmstart model that took an existing daily fleet assignment, and then tried to improve it by using local swap opportunities. Theoretical properties of the FAM problem are presented in Gu et al [14]. They also present the behavior of the FAM solution obtained as a function of the number of fleets.

The solution of FAM affects subsequent planning decisions like maintenance requirements, aircraft routing, and crew scheduling. Consequently, extensions of FAM were considered in later research. Clarke et al [15], extended basic FAM to address both maintenance and crew. Their formulation avoids crew "lonely overnights." Lonely overnights occur when a crew arrives at a station (not the crew's base) late in the evening, and its aircraft leaves before the crew has sufficient rest time. Rushmeier and Kontogiorgis [3] presented a FAM formulation to deal with aircraft routing issues. They modeled FAM in a more flexible manner by using an event-activity network rather than a closed-loop space-time network in general. They partition the set of operations at each station into connecting complexes: each complex is a balanced subset of incoming and outgoing flights (similar to islands in [12], discussed later), and an incoming leg is allowed to connect to an outgoing leg in the same complex. They introduced soft constraints with penalties to make the solution more maintenance and crew friendly.

Barnhart et al [16] utilized *strings* to represent the assignment of a sequence of legs to a single aircraft. A string is a sequence of connected flights that begins and ends at a maintenance base. They presented a single model and solution approach to simultaneously solve the fleet assignment and aircraft routing problems. Because of the possibility of numerous strings, Barnhart et al proposed a delayed column generation technique to generate maintenance feasible routing solutions. Using a string based model, Rosenberger et al [17] proposed a robust FAM that creates *rotation cycles* (i.e., sequence of legs assigned to each aircraft) that are short and that involve as few hub stations as possible, so that flight cancellations or delays will have a smaller chance to disrupt the entire network.

An integrated fleet assignment and schedule decision was presented by Rexing et al [18], by allowing variability in scheduled departure times for improving flight connections through time windows. They add multiple copies of each flight, spaced at five-minute intervals through the window. Only one of those copies is assigned in the solution. They reduce the problem size using node aggregation, and they also eliminate arcs associated with the flight copies that become redundant after node aggregation. Ahuja and Orlin [19] developed a neighborhood search method to integrate fleet assignment and aircraft routing problems. Their approach starts with a feasible solution obtained by solving the FAM, not necessarily to optimality, and then determines the through-flights sequentially. Lettovsky et al [20] proposed a column generation approach to either improve an existing schedule or generate a schedule from scratch for an airline. Each column represents the scheduled service plan for an origin-destination pair.

An important component of a successful FAM is modeling the objective function. The objective function can be maximizing revenue or minimizing passenger spill (lost revenue due to assigning smaller aircraft) or minimizing the number of aircraft being used. For the objective, the FAM models addressed above assume demand for each flight leg is independent and deterministic. In practice, both the assumptions are invalid. As in a multi-leg itinerary, capacity of one flight leg affects the revenues of others and the demand forecasts made early in the planning process can have significant errors.

Farkas [21] demonstrated that Revenue Management (RM) has a significant impact on passenger volume and mix, and by ignoring these effects FAM produces sub-optimal solutions. Kniker [22] augmented FAM with the Passenger Mix Model (PMM) to capture multi-leg passenger itineraries. Given a schedule with known flight capacities and a set of passenger demands with known fares, the combined FAM and PMM determines optimal demand and fleet assignments to maximize revenue. This is also referred to as Origin Destination (O-D) FAM. PMM assumes that demand is deterministic and that the airline has complete knowledge and control of which passengers they accept. Kniker solved the problem using branch-and-price and utilized sophisticated preprocessing techniques to reduce the computation involved. Barnhart et al [23] developed an alternative model to solve the O-D FAM problem addressed by Kniker using decision variables that assign a subset of legs to a fleet. They showed that by carefully selecting subsets, the model is computationally tractable.

Jacobs et al [24] presented an O-D FAM formulation and solved it using Benders' decomposition. Given an assignment solution, the revenue is estimated in the RM sub-problem. The revenue function is approximated in the master problem with a series of Benders' cuts with each cut improving the accuracy. When a specified accuracy is reached in the relaxed master problem, the assignment variables are changed to integer and an MIP is solved. Though this approach addresses both passenger flows in the network and demand uncertainty, Smith [25] mentions that this approach can suffer from slow convergence and high fractionality.

The stochastic nature of passenger demand was recently addressed in Listes and Dekker [26] for determining an optimal airline fleet composition. Given an airline schedule and a set of aircraft types, the fleet composition problem determines the number of aircraft of each type the fleet requires in order to make the maximum profit. They developed a two-stage stochastic model to determine a single fleet composition that maximizes profit in the first stage across all demand scenarios generated in the second stage. In the second stage they solved a deterministic FAM model for each demand scenario allowing swapping to occur (similar to Berge and Hooperstad [6]) and employed a scenario aggregation approach (Rockafellar and Wets [27]) to reduce the computational complexity. They report a 90% runtime reduction and profit benefits up to 0.5 margin points. Even though the authors model the stochasticity of demand, the network effects are still ignored.

Recently, Sherali and Zhu [28] developed a two-stage stochastic mixed-integer programming approach in which the first stage makes only a higher-level family-assignment decision, while the second stage performs subsequent family-based type-level assignments according to forecasted market demand realizations. They conducted polyhedral analysis of the proposed model and developed suitable solution approaches.

In this dissertation, we have modeled combined FAM and PMM to account for the network effects and incorporate the stochastic nature of demand by utilizing a two-stage SP framework. As mentioned in Chapter 1, DACE has been utilized to reduce the computation involved. For convenience, the augmented FAM is just addressed as FAM.

CHAPTER 3

GENERATING RECOURSE FUNCTION

This chapter focuses on generating a recourse function for the two-stage SP FAM problem, which will be utilized for optimization. A brief introduction to DACE is presented in Section 3.1 followed by Sections 3.2 - 3.5, which explain in detail the different steps involved in DACE to solve the FAM problem. The results of the DACE phase are presented in Section 3.6.

3.1 Design and Analysis of Computer Experiments Approach

In statistics, a common problem is to model the relationship between n explanatory or input variables (vector x) and a *response variable* y, which is expressed generally as:

$$E\left[y|x\right] = f(x) \tag{3.1}$$

where $f(\cdot)$ is an unknown function. Many methods from statistical experimental design (DoE) and statistical modeling have been developed to address the estimation of the function $f(\cdot)$. A Design and Analysis of Computer Experiments (DACE) approach is useful when a computer experiment is the only means for representing a complex system (see Chen et al [9], Tsai and Chen [29]). DACE-related methods have been successfully employed for solving stochastic dynamic programming and Markov decision problems ([30] - [31]), and their use was first suggested for stochastic programming by Chen [8]. This dissertation represents the first implementation of these ideas. Typically, the computer experiment is a simulation model; however, here it is an optimization model.

The steps involved in DACE are:

- An optimization model (computer experiment) of system performance is constructed based on knowledge of how the system operates.
- Design of Experiments (DoE) is used to select the set of sample points as input to the optimization model, which then provides the corresponding responses.
- A statistical model is to fit these data.

Once a fit is obtained, the recourse function is optimized to generate a single first stage solution. Since a fit for the recourse function can be obtained well in advance of the 90 day period, we have utilized two phases to further reduce the computation involved, as shown in Figure 3.1. In the DACE phase, since the first stage requires only the crew-compatible allocation (CCA) of aircraft, a reduced state space corresponding to the crew group allocation is generated. Within this state space DoE can be used to select the discretization points at which the optimization will be solved. A more detailed explanation for generating the reduced state space and experimental design is discussed in Section 3.3.1. In particular, DoE points for the CCA space must be feasible, which is an issue not typically handled by DACE methods; thus, a new approach is devised and described in Section 3.3. In order to estimate the second-stage revenue function, scenarios are generated based on known probability distributions, and for each CCA, average revenues from the second-stage FAM optimization are collected as y (see Section 3.4 for details). A MARS statistical model, $\hat{y}(\text{CCA})$ is fit to these data to generate an approximate second-stage recourse function, which can then be employed for more efficient future evaluations in the Optimization phase of Figure 3.1 and optimized by dynamically generating revenue cuts.

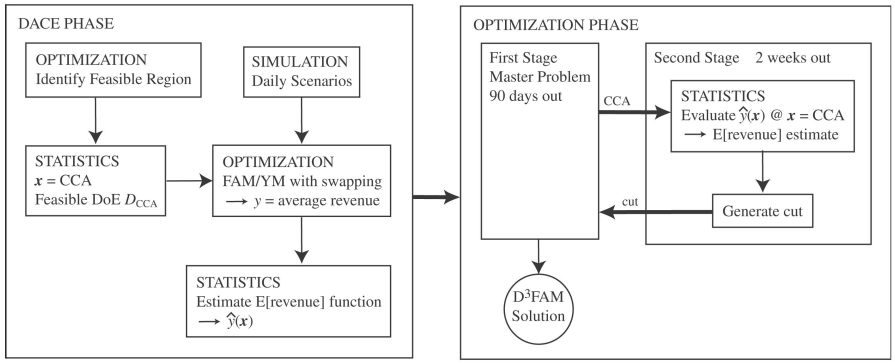

Figure 3.1. Two-Stage Stochastic Programming framework.

3.2 Two-Stage SP FAM Formulation

Given an airline schedule and a set of fleets of different aircraft that can fly each flight leg, the fleet assignment problem allocates the fleet of aircraft to the scheduled flights subject to the following operational constraints:

- Balance: Aircraft cannot appear or disappear from the network.
- Cover: Each flight in the schedule must be assigned to exactly one aircraft type.
- Plane Count: The total number of aircraft assigned cannot exceed the number of available aircraft in the fleet.

The objective is to find a feasible assignment that maximizes profit. In this dissertation, the fleet assignment problem is formulated as an integer multicommodity flow problem on a time line network similar to those of Berge and Hooperstad [6] and Hane et al [12]. A time line is a graph that represents the arrival and departure events occurring at each station over a specified time period as shown in Figure 3.2.

Flights below the time line indicate arrivals, flights above the time line indicate departures, and the numbers indicate the corresponding flight legs. Any flight, which arrives at a particular station, will not be available for departure immediately because of the time required for fueling, loading passengers/baggage etc. As such a *turn time* is added to all the arrivals before they are ready to take off. The turn time is dependent on

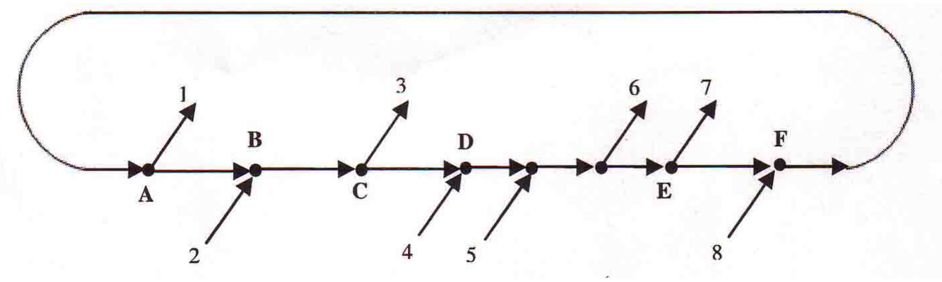

Figure 3.2. Time Line.

the particular fleet type and the station. A *node* in a time line begins with an arrival and ends before the next arrival with at least one departure in between. In Figure 3.2, BC, DE and FA represent nodes. The arcs that connect within these nodes are called *ground arcs*, and the arc that connects the last arrival on the time line to the first departure is called the *overnight arc*. These arcs denote at least one plane being on the ground at a station and are defined as continuous variables because once all flight variables are integral the values corresponding to these arcs will be integral as well. The sum of all the times corresponding to the arcs represent the *total ground time* of the planes at that particular station. The overnight arc includes a *plane count hour* (typically 4 A.M. EST) that is used for developing the plane count constraints. A detailed description of how to generate the constraints based on the time line is given in Section 3.3.1.

3.2.1 Model Assumptions

The following assumptions are taken to solve the two-stage SP FAM problem:

- The mean and variance for each itinerary-fare class are known.
- Spilled passengers are assumed to be lost by the carrier and are not recaptured.
- The decision variables are relaxed to be continuous in the DACE phase, but then integrality is maintained in the Optimization phase.

3.2.2 Stochastic Model

Let L be the set of flight legs (indexed by l). Let F denote the set of fleet types (indexed by f), and let G be the set of crew-compatible families (indexed by g), which can be used for each of the legs $l \in L$. Let e be a mapping from F to G ($e : F \rightarrow G$). Since we assign, crew-compatible families in the first stage, for each leg $l \in L$ and for each crew-compatible family type $g \in G$, let a binary variable x_{gl} be defined such that

$$x_{gl} = \begin{cases} 1 & \text{if crew-compatible family } g \text{ is assigned to flight leg } l, \\ 0 & \text{otherwise.} \end{cases}$$

In the second stage, we assign specific aircraft within the crew-compatible family. As such, for each leg $l \in L$, for each aircraft $f \in F$, and for scenario $\xi \in \Xi$, let a binary variable x_{fl}^{ξ} be defined such that

$$x_{fl}^{\xi} = \begin{cases} 1 & \text{if aircraft } f \text{ is assigned to the leg } l \text{ for scenario } \xi, \\ 0 & \text{otherwise.} \end{cases}$$

Since a combined FAM and PMM model is used, let the decision variable z_i^{ξ} represent the number of booked passengers for itinerary-fare class i for scenario ξ.

For basic FAM + PMM, consider the following additional parameters:

- S = set of stations, indexed by s,
- I = set of itinerary-fare classes, indexed by i,
- V = set of nodes in the entire network, indexed by v,
- $f(v)$ = fleet type associated with node v,
- A_v = set of flights arriving at node v,
- D_v = set of flights departing at node v,
- M_f = number of aircraft of type f,
- f_i = fare for itinerary-fare class i,
- C_{fl} = cost if aircraft type f is assigned to flight leg l,

- $a^{\xi}_{v^+}$ = value of ground arc leaving node v for scenario ξ,
- $a^{\xi}_{v^-}$ = value of ground arc entering node v for scenario ξ,
- O_f = set of arcs that include the plane count hour for fleet type f, indexed by o,
- L_0 = set of flight legs in air at the plane count hour,
- Cap_f = capacity of aircraft f,
- D^{ξ}_i = demand for itinerary-fare class i for scenario ξ.

The two-stage formulation can be represented as:

$$\max\ \theta\ = E\left[-\sum_{l\in L}\sum_{f\in F} C_{fl}(x^{\xi}_{fl}) + \sum_{i\in I} f_i z^{\xi}_i\right] \tag{3.2}$$

$$\text{s.t. } a^{\xi}_{v^-} + \sum_{l\in A_v} x^{\xi}_{f(v)l} - \sum_{l\in D_v} x^{\xi}_{f(v)l} - a^{\xi}_{v^+} = 0 \qquad \forall v\in V, \xi\in\Xi \tag{3.3}$$

$$\sum_{f\in g} x^{\xi}_{fl} = x_{gl} \qquad \forall l\in L, g\in G, \xi\in\Xi \tag{3.4}$$

$$\sum_{o\in O_f} a^{\xi}_o + \sum_{l\in L_0} x^{\xi}_{fl} \le M_f \qquad \forall f\in F, \xi\in\Xi \tag{3.5}$$

$$\sum_{i\ni I} z^{\xi}_i - \sum_{f\in F} Cap_f\ x^{\xi}_{fl} \le 0 \qquad \forall l\in L, \xi\in\Xi \tag{3.6}$$

$$0 \le z^{\xi}_i \le D^{\xi}_i \qquad \forall i\in I, \xi\in\Xi \tag{3.7}$$

$$x^{\xi}_{fl} \in \{0,1\} \qquad \forall l\in L, \xi\in\Xi \tag{3.8}$$

$$x_{gl} \in \{0,1\} \qquad \forall l\in L, g\in G \tag{3.9}$$

$$a^{\xi}_{v^+} \ge 0 \qquad \forall v\in V, \xi\in\Xi. \tag{3.10}$$

The objective is to maximize profit (revenue - cost) in the second stage by assigning aircraft within the crew-compatible allocation made in the first stage. The *block time* of a flight leg l is defined as the length of time from the moment the plane leaves the origin station until it arrives at the destination station. Let b_l be the scheduled block time for flight leg l. The cost for each flight leg is calculated as a function of block time and operating cost of a particular fleet type per block hour, and is given by:

$$C_{fl} = b_l * (\text{Operating cost per block hour})_f.$$

Constraints in set (3.3) represent the balance constraints needed to maintain the circulation of aircraft throughout the network. Cover constraints (3.4) guarantee that aircraft within the crew-compatible family (assigned in the first stage) are allocated. For formulating the plane count constraints (3.5), we need to count the number of aircraft of each fleet being used at any particular point of day (generally when there are fewer planes in the air). As such the *ground arcs* that cross the time line at the plane count hour and the flights in air during that time are summed to assure that the total number of aircraft of a particular fleet type do not exceed the number available. Constraints (3.6) impose the seat capacity limits, i.e., the sum of all the booked passengers on different itineraries for a flight l should not exceed the capacity of the aircraft assigned and constraint (3.7) to meet the forecasted demand. Binary constraints (3.8) are relaxed to model the decision variables as continuous. As in practice most crew-compatible families include only one or two aircraft types, so integer solutions result [6]. For families with more than two aircraft types, an upper bound on the objective function can be obtained.

Since the variables are relaxed, the resulting fit for the recourse function obtained using MARS provides an upper bound for the profit generated. A true recourse function can be estimated with MARS using integral values only at the expense of higher computational time. This is difficult in the case of solving the two-stage SP with traditional Benders' decomposition as there are integrality issues during the generation of cuts.

3.3 Design of Experiments Approach

The main objective of the first-stage solution is to assign an initial the crew-compatible allocation (CCA) using DoE. The design is generated within the experimental region formed by the first-stage FAM constraints. Since we are concerned with only the

CCA, the first-stage constraints are similar to the constraints described earlier except that they are based on crew-compatible family variables as defined below:

$$(Balance) \qquad a_{v^-} + \sum_{l \in A_v} x_{gl} - \sum_{l \in D_v} x_{gl} - a_{v^+} = 0 \qquad \forall v \in V \tag{3.11}$$

$$(Cover) \qquad \sum_{g \in G} x_{gl} = 1 \qquad \forall l \in L, g \in G \tag{3.12}$$

$$(PlaneCount) \qquad \sum_{o \in O_g} a_o + \sum_{l \in L_0} x_{gl} \leq M_g \qquad \forall g \in G \tag{3.13}$$

where O_g represents the set of arcs that include the plane count hour for crew-compatible family g, and M_g denotes the number of aircraft of crew group g.

In order to use DoE, the above constraints must be preprocessed such that they form a *polytope* defined by a system of linear inequalities $Ax \leq b$. Preprocessing is conducted by exploiting the explicit [32], as well as implicit equalities present in the constraints as defined in Section 3.3.1. Savelsbergh [33] demonstrated the use of preprocessing techniques to reduce MIP problems.

Let $P := \{x \in R | Ax \leq b\}$ be a nonempty convex *polytope* formed by the first-stage constraints. Discretized points within this *polytope* can be generated as shown in Section 3.3.2 to represent the initial CCA. Any infeasible points generated during the design can be projected onto the feasible polytope as described in detail in Section 3.3.2.2.

3.3.1 Generating Reduced State Space

It is important to reduce the first-stage FAM to the minimum number of variables as the precision of the estimates are adversely affected by multicollinearity (or correlations) among the input variables [34]. Reducing to the minimum number of variables in the first stage removes the redundancy, as well as dramatically improves the efficiency of the design. In order to generate the feasible polytope P, we need to pivot out variables using the equality constraints as demonstrated in Sections 3.3.1.1 and 3.3.1.2, and then remove the implicit equalities as discussed in Section 3.3.1.3. In particular, Section 3.3.1.1 discusses the application of the Boeing concept of D^3, and Section 3.3.1.2 takes advantage of the first-stage constraints to reduce the number of variables. Once the equalities are removed, we are left with a set of minimum variables that defines the first-stage decision space.

3.3.1.1 Demand Driven Dispatch

The robust FAM approach presented here incorporates the Boeing concept of D^3 as defined in Section 1.2. In the first stage (90 days prior to departure), before most demand is realized, crew-compatible families of aircraft are assigned to flights. As such, in the first stage, only one variable needs to be considered when there is more than one aircraft belonging to the same crew-compatible family that can be allocated to a flight leg $l \in L$. For example, suppose flight leg l can be assigned to six different fleet types; that is $F = \{1, 2, 3, 4, 5, 6\}$. The cover constraint for flight leg l is given by:

$$x_{1l} + x_{2l} + x_{3l} + x_{4l} + x_{5l} + x_{6l} = 1. \tag{3.14}$$

Suppose aircraft 1 belongs to crew-compatible family 1, aircraft 2 and 3 belong to crew-compatible family 2, and aircraft 4, 5, and 6 belong to crew-compatible family 3; that is $G = \{1, 2, 3\}$. In this case we substitute x_{2l} and x_{3l} with just x_{2l}, and we replace x_{4l}, x_{5l} and x_{6l} with x_{3l}. Then the cover constraint for flight leg l becomes:

$$x_{1l} + x_{2l} + x_{3l} = 1. \tag{3.15}$$

Thus, the number of variables have been reduced from six to three.

3.3.1.2 Constraints

The special structure of each of the constraints in the first stage can be exploited to reduce the number of variables over which the design has to be generated. Section

3.1.2.1 addresses the reduction due to cover constraints and Section 3.1.2.2 describes the reduction achieved due to balance constraints.

3.3.1.2.1 Cover Constraints

For every leg l, the cover constraint obtained after reduction through the Boeing concept can be manipulated to reduce the dimensionality by one. For example, for equation (3.15) we have:

$$x_{1l} + x_{2l} + x_{3l} = 1 \quad \Rightarrow \quad x_{3l} = 1 - x_{1l} - x_{2l}. \tag{3.16}$$

We also have the constraints:

$$x_{gl} \in \{0,\, 1\} \text{ and } g \in G = \{1,\, 2,\, 3\}. \tag{3.17}$$

Since we are modeling the variables as continuous, we have

$$0 \le x_{3l} \le 1 \quad \Rightarrow \quad 0 \le 1 - x_{1l} - x_{2l} \le 1 \quad \Rightarrow \quad 0 \le x_{1l} + x_{2l} \le 1. \tag{3.18}$$

3.3.1.2.2 Balance Constraints

The balance constraints maintain the circulation of aircraft throughout the entire flight network. Based on the ground time of the planes and the traffic intensity, stations are typically classified as hubs and spokes.

Balance Constraints for Spokes

Stations with low traffic intensity are called *spokes*. A spoke time line consists of sudden activity during a period of time where we can see equal numbers of arrivals and departures. Therefore, we can find time periods in a spoke time line when there are no planes on the ground. The ground arcs corresponding to these periods can be dropped. *Islands* can be defined as a set of nodes in which the incoming and outgoing ground arcs are zero, and a *complex island* has two or more nodes. Figure 3.3 represents a spoke time line.

In Figure 3.3, A, B, C, and E represent nodes; B and D represent islands while A forms an *overnight island*. Island D is a complex island with nodes C and E in it. The time line for each crew-compatible family assists us in framing the balance constraints at each station. Consider a spoke time line for crew-compatible family g as shown in Figure 3.3. The balance constraint for node B is:

$$x_{g3} - x_{g4} = 0 \qquad \Rightarrow \qquad x_{g3} = x_{g4}. \tag{3.19}$$

Thus, the crew group type of x_{g3} will determine that of x_{g4}. For node C, the balance constraints are:

$$x_{g5} + x_{g6} - x_{g7} - a_g = 0 \tag{3.20}$$

$$a_g \geq 0 \tag{3.21}$$

where a_g represents the ground arc from node C to E. For node E, the balance constraints are:

$$a_g + x_{g8} - x_{g9} - x_{g10} = 0. \tag{3.22}$$

From (3.20), (3.21) and (3.22), by dropping the ground arc variable a_g the balance constraints for the island D are:

$$x_{g5} + x_{g6} - x_{g7} \geq 0 \tag{3.23}$$

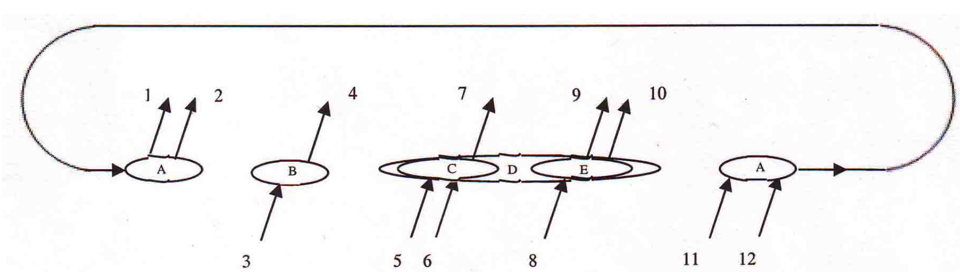

Figure 3.3. Spoke Time Line.

$$x_{g5} + x_{g6} - x_{g7} + x_{g8} - x_{g9} - x_{g10} = 0 \quad \Rightarrow \quad x_{g10} = x_{g5} + x_{g6} - x_{g7} + x_{g8} - x_{g9}. \tag{3.24}$$

Similarly, for node A:

$$x_{g11} + x_{g12} - x_{g1} - x_{g2} = 0 \quad \Rightarrow \quad x_{g2} = x_{g11} + x_{g12} - x_{g1}. \tag{3.25}$$

Thus, the total number of decision variables is reduced from twelve to nine. Similarly, the balance constraints are developed at other spoke stations.

Balance Constraints for Hubs

A *hub* consists of continuous activity with periodic flights coming from the spokes. Because of the high traffic intensity at hubs we cannot drop any ground arcs as we have done at spokes. Consider a hub time line for crew-compatible family g with nodes A, B and C as represented in Figure 3.4. The balance constraints for nodes A and B are given by:

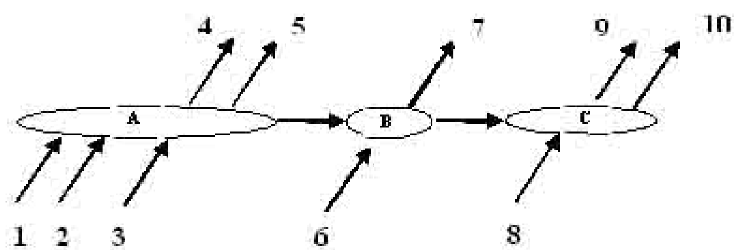

Figure 3.4. Hub Time Line.

$$a_g + x_{g1} + x_{g2} + x_{g3} - x_{g4} - x_{g5} \geq 0 \tag{3.26}$$

$$a_g \geq 0 \tag{3.27}$$

$$a_g + x_{g1} + x_{g2} + x_{g3} - x_{g4} - x_{g5} + x_{g6} - x_{g7} \geq 0. \tag{3.28}$$

Since the time line ends with the last departure at node C, the constraints for node C will determine the overall balance constraint for the hub. Therefore, we have:

$$x_{g1} + x_{g2} + x_{g3} - x_{g4} - x_{g5} + x_{g6} - x_{g7} + x_{g8} - x_{g9} - x_{g10} = 0$$

$$\Rightarrow \quad x_{g10} = x_{g1} + x_{g2} + x_{g3} - x_{g4} - x_{g5} + x_{g6} - x_{g7} + x_{g8} - x_{g9}. \tag{3.29}$$

For each of the hubs, there will be one equality constraint for the entire time line to maintain the balance of flights arriving and departing. Hence, the number of decision variables reduces by one, and there is one ground arc variable for each crew-compatible family.

3.3.1.3 Implicit Inequalities

In this section, we generate equations based on implicit opposing inequalities from the set of constraints $Ax \leq b$. This is done by formulating an LP problem that finds two implicit opposing inequality constraints using convex combinations of the explicit inequalities. Any feasible solution to the LP refers to an implied equation that can be generated.

An inequality $\alpha x \leq \beta$ from $Ax \leq b$ is called an implicit equality (in $Ax \leq b$), if $\alpha x = \beta$, for all x satisfying $Ax \leq b$. To check if there are any implicit equalities among the system of constraints defined by

$$\textstyle\sum_{j=1}^{n} a_{ij}x_j \leq b_i, \qquad \forall i = 1, 2, ..., m, \tag{3.30}$$

we formulate an LP. Let $z \geq 0$ be a multiplier satisfying equation (3.30), so

$$\textstyle\sum_{j=1}^{n} a_{ij}x_j z_i \leq b_i z_i, \qquad \forall i = 1, 2, ..., m.$$

P is full-dimensional if and only if $P' := \{z | A^T z = 0, b^T z = 0, z \geq 0\} = \{0\}$. The proof for this statement is detailed in Gao and Zhang [35], Sierksma and Tijssen [36], and Pilla et al [37]. The formulated LP is a maximization problem as we are looking for a non-zero solution. Since the constraints in P' form a cone, we need to add a constraint of the form $\sum_{i=1}^{m} z_i \leq M$, where M is a constant, to get a feasible solution. Thus, the LP can be represented as:

$$\max \sum_{i=1}^{m} z_i \tag{3.31}$$

$$\text{s.t. } A^T z = 0 \tag{3.32}$$

$$b^T z = 0 \tag{3.33}$$

$$\sum_{i=1}^{m} z_i \leq 1 \tag{3.34}$$

$$z \geq 0 \tag{3.35}$$

If nonzero solution exists for the LP, then an implicit equality can be generated.

The following simple example illustrates the above mentioned methodology. Consider the system of inequalities shown below:

$$x_1 + x_2 + x_3 \leq 3 \tag{3.36}$$

$$-x_1 \leq -1 \tag{3.37}$$

$$-x_2 \leq -1 \tag{3.38}$$

$$-x_3 \leq -1 \tag{3.39}$$

$$x_3 + x_4 \leq 2 \tag{3.40}$$

$$-x_4 \leq 0. \tag{3.41}$$

Consider z_i, for $i = 1, ..., 6$, multipliers for each inequality. In order to check for implicit equalities we can formulate our LP as:

$$\max \sum z_i \tag{3.42}$$

$$s.t.\ z_1 - z_2 = 0 \tag{3.43}$$

$$z_1 - z_3 = 0 \tag{3.44}$$

$$z_1 - z_4 + z_5 = 0 \tag{3.45}$$

$$z_5 - z_6 = 0 \tag{3.46}$$

$$3z_1 - z_2 - z_3 - z_4 + 2z_5 = 0 \tag{3.47}$$

$$z_1 + z_2 + z_3 + z_4 + z_5 + z_6 = 1 \tag{3.48}$$

Bounds

$$z_i \geq 0, \forall i = 1, ..., 6. \tag{3.49}$$

The constraint $\sum_{i=1}^{m} z_i \leq 1$ for $i = 1, \ldots, 6$ is added to get a solution in the unbounded cone formed by the remaining constraints. Solving with CPLEX, we get the solution as:

$$\begin{aligned} z_1 &= 0.25 \\ z_2 &= 0.25 \\ z_3 &= 0.25 \\ z_4 &= 0.25 \\ z_5 &= 0 \\ z_6 &= 0. \end{aligned} \tag{3.50}$$

Since the z_i are multipliers for each equation, we can write constraint (3.36) as,

$$\begin{aligned} 0.25 * (+x_1 + x_2 + x_3) &\leq 0.25 * 3 \\ \Rightarrow \qquad x_1 + x_2 + x_3 &\leq 3, \end{aligned} \tag{3.51}$$

and, combining constraints (3.37), (3.38) and (3.39), we have

$$\begin{aligned} 0.25*(-x_1)+0.25*(-x_2)+0.25*(-x_3) &\leq 0.25*(-1-1-1) \\ \Rightarrow -x_1-x_2-x_3 &\leq -3 \\ \Rightarrow x_1+x_2+x_3 &\geq 3. \end{aligned} \tag{3.52}$$

From equation (3.51) and equation (3.52) we have the implicit equality

$$x_1+x_2+x_3=3, \tag{3.53}$$

and can pivot out one variable.

A feasible FAM problem with 50 stations, 2358 legs, seven fleet types and four crew-compatible families was considered to test the methodology. The two implicit inequalities generated were:

$$\begin{aligned} &+y_{0\_1}+y_{5\_1}+y_{9\_1}+y_{10\_1}+y_{0\_3}+y_{5\_3}+y_{10\_3}+y_{16\_3}+y_{28\_3}+y_{0\_4}+y_{5\_4} \\ &+y_{9\_4}+y_{10\_4}+y_{16\_4}+y_{28\_4}-x_{262\_2}+x_{1061\_2}-x_{1692\_2}+x_{1978\_2} \\ &+x_{1980\_2}-x_{2140\_2}-x_{2141\_2}<=28 \end{aligned} \tag{3.54}$$

$$\begin{aligned} &-y_{0\_1}-y_{5\_1}-y_{9\_1}-y_{10\_1}-y_{0\_3}-y_{5\_3}-y_{10\_3}-y_{16\_3}-y_{28\_3}-y_{0\_4}-y_{5\_4} \\ &-y_{9\_4}-y_{10\_4}-y_{16\_4}-y_{28\_4}+x_{262\_2}-x_{1061\_2}+x_{1692\_2}-x_{1978\_2} \\ &-x_{1980\_2}+x_{2140\_2}+x_{2141\_2}<=-28. \end{aligned} \tag{3.55}$$

Basing on the two opposing inequalities from equation (3.54) and equation (3.55), we can conclude that:

$$\begin{aligned} &+y_{0\_1}+y_{5\_1}+y_{9\_1}+y_{10\_1}+y_{0\_3}+y_{5\_3}+y_{10\_3}+y_{16\_3}+y_{28\_3}+y_{0\_4}+y_{5\_4} \\ &+y_{9\_4}+y_{10\_4}+y_{16\_4}+y_{28\_4}-x_{262\_2}+x_{1061\_2}-x_{1692\_2}+x_{1978\_2} \\ &+x_{1980\_2}-x_{2140\_2}-x_{2141\_2}=28. \end{aligned} \tag{3.56}$$

Equation (3.56) represents a hyperplane and one variable can be pivoted out from this valid equality.

Figure 3.5 shows the flow chart used for reducing the dimensionality of the FAM problem.

3.3.2 Generating the Experimental Design

An underlying form for the expected revenue of a FAM model cannot be assumed; thus, our desired experimental design should adequately represent the CCA space. This is challenging for several reasons that will become apparent in the design generation process. After reducing dimensionality, the design is constructed in three steps:

- A Latin hypercube algorithm is devised to generate DoE points that satisfy the triangular FAM cover constraints. This will provide a set of well-distributed points in the CCA space that are close to the feasible region.
- The current DoE points are projected to extreme points of the (convex) feasible region.
- Another Latin hypercube is used to generate interior feasible points as convex combinations of the extreme points.

McKay et al [38] introduced Latin hypercube sampling, in which the continuous range of each variable is partitioned into n intervals, each interval for each variable is sampled exactly once, and the univariate sample values are randomly matched across all the variables to form the n sample points. A more detailed description of the various design approaches are presented in Chen et al [9]. If we project n points of a LH design onto any single dimension, the points would correspond to n distinct levels in that dimension. Because perfect correlation between two input variables is possible in a Latin hypercube design, our algorithm generates a larger design and columns are eliminated so that the final design has low input variable correlations.

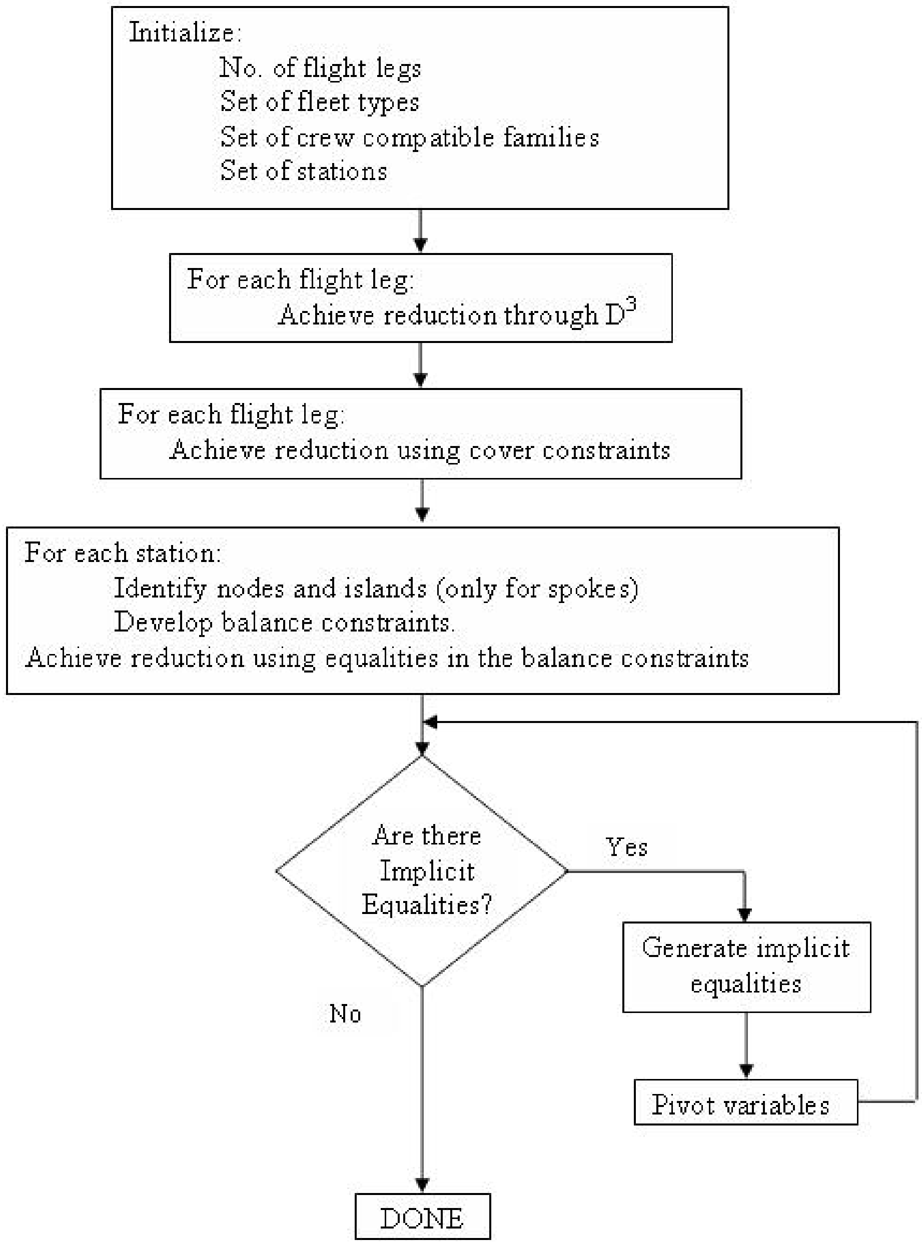

Figure 3.5. State Space Reduction Flow Chart.

3.3.2.1 Latin Hypercube Algorithm for the Reduced Cover Constraints

After dimensionality reduction, suppose m is the number of crew-compatible types, n is the number of legs, and N is the total number of design points, where $N = \lambda p > mn$, and λ is the frequency parameter of the Latin hypercube design.

- Draw a grid of p points in a m-dimensional hyper-triangle representing a reduced cover constraint over crew compatible types.
- Order the p points according to the distance from the origin. Randomly break ties $\rightarrow 1, 2, \ldots, p$ (each is a m-tuple).
- Generate a Latin hypercube with $N = \lambda p$ in n dimensions.
- Randomly assign the Latin hypercube levels for each dimension:

  $1, \ldots, \lambda \rightarrow 1$

  $\lambda + 1, \ldots, 2\lambda \rightarrow 2$

  $\vdots$

  $(i-1)\lambda + 1, \ldots, i\lambda \rightarrow i, \qquad \text{for } i = 1, 2, \ldots, p$

Since each of the reduced cover constraints has a different number of crew-compatible types (m), our design generates $N = \lambda p_1 p_2 \ldots p_m$ points.

For example, suppose, the constraints are:

$$x_{11} \leq 1$$

$$x_{21} \leq 1$$

$$x_{31} + x_{32} \leq 1.$$

We have two legs with one dimension and one leg with two dimensions. Consider four points or levels for the variable with one dimension $\Longrightarrow p_1 = 4$, and three levels for the variable with two dimensions $\Longrightarrow p_2 = 3$. Consider $\lambda = 1$, so $n = 3$ and $N = \lambda p_1 p_2 = 12$. Define, $p_1 = \{0, 0.25, 0.5, 1\}$ (corresponding to four levels) and
$p_2 = \{(0,0), (0,1), (1,0)\}$ (corresponding to three levels).

Suppose the following design is generated as shown in Table 3.1.

Table 3.1. LH Design Levels

Design Point	Leg Number		
	1	2	3
1	0	3	0
2	1	1	1
3	2	0	2
4	1	2	1
5	0	1	0
6	3	3	2
7	3	0	2
8	0	1	1
9	2	2	0
10	1	0	1
11	2	3	2
12	3	2	0

The obtained LH design can be converted to a set of design points by mapping each of these numbers in the design to their respective levels as defined earlier. For example, the values in Table 3.1 can be represented as in Table 3.2.

The levels and the corresponding points are selected such that the points are evenly spread across the design space. The method that we have used here is basically to maximize the minimum Euclidean distance between points. For a constraint with a single variable, we need to spread the points evenly across the line, but for a constraint with more than one variable (two variables it is a triangle, three is a hyper-triangle, etc.), we need to spread them evenly across the design space.

Table 3.2. Design Points

Design Point	Leg Number			
	1	2	3	
1	0	1	0	0
2	0.25	0.25	0	1
3	0.5	0	1	0
4	0.25	0.5	0	1
5	0	0.25	0	0
6	1	1	1	0
7	1	0	1	0
8	0	0.25	0	1
9	0.5	0.5	0	0
10	0.25	0	0	1
11	0.5	1	1	0
12	1	0.5	0	0

A quadratic program that maximizes the minimum pairwise distance t_{ij} between N points in a set K can be represented as:

$$\max_{x_i, y_i} \quad \min_{(i,j)\in K} \quad t_{ij} \tag{3.57}$$

$$s.t. \quad (x_i - x_j)^2 + (y_i - y_j)^2 \;=\; t_{ij} \quad \forall (i,j) \in K \tag{3.58}$$

$$x_i + y_i \;\leq\; 1 \forall i = 1, 2, \ldots, N \tag{3.59}$$

$$x_i \;\geq\; 0 \tag{3.60}$$

$$y_i \;\geq\; 0 \tag{3.61}$$

The problem of maximizing the minimum pairwise distance among n points in a unit square is equivalent to the problem of finding the maximum diameter of equal non-overlapping circles contained in a unit square [39]. For example, a constraint with two variables can be represented in the form a triangle and it six equidistant points can be found as shown in Figure 3.6. It is preferable to have most of the points on the boundary, as the solution to equations (3.11) - (3.13) will lie at the corner points.

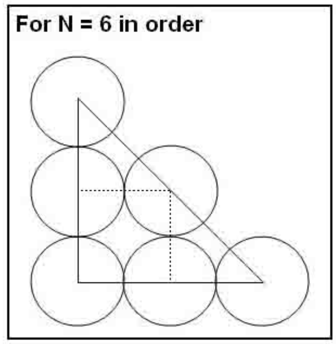

Figure 3.6. Overlapping circles for generating design.

3.3.2.2 Projecting to the Feasible Region

It is critical that the final design points lie within the (convex) feasible region. Different options were explored, including the omission of any infeasible points and the projection of points into the feasible region. The most promising option was projection of points to nearby extreme points of the feasible region. Given that the original set of design points is well-distributed, the resulting set of extreme points should provide good coverage. However, it should be noted that it is not computationally practical to identify all extreme points, so some regions of the feasible region will not be covered. The infeasible points (x^0) are projected onto the feasible region using the L1 norm as represented below:

$$\min \|x - x^0\| \tag{3.62}$$

$$s.t. \quad Ax \leq b \tag{3.63}$$

$$x \geq 0. \tag{3.64}$$

Suppose the solution to the above LP is ($\tilde{x}$). The solution ($\tilde{x}$) actually lies on the face of the feasible polytope $P := \{x | Ax \leq b, x \geq 0\}$. Finding a nearby extreme point to a point on the face of the polytope is a difficult task.

We optimized in the direction of the gradient of the L2 norm given by $\|x^0 - x\|^2$, and exploited the features in ILOG CPLEX 8.0. Since every iteration of primal simplex provides a basic feasible solution that is an extreme point, we can optimize in the direction of the gradient of the L2 norm by providing $\tilde{x}$ as the starting information and stopping the optimization at the first iteration. This will not guarantee the solution to the nearest extreme point, but by providing $\tilde{x}$ as the initial primal values and a single call to an optimization routine, we assume that an extreme point close to $\tilde{x}$ is obtained. The LP to optimize the gradient of the L2 norm at the point $\tilde{x}$ is represented as:

$$\min(x^0 - \tilde{x})^T x \tag{3.65}$$

$$s.t. \quad Ax \leq b \tag{3.66}$$

$$x \geq 0. \tag{3.67}$$

Nearby extreme points could also be obtained using neighborhood search methods like *breadth first search method* [40] or *interior point methods*.

3.3.2.3 Interior Feasible Points

Although a MARS approximation could be fit to the extreme points alone, interior points would enable better representation of the shape of the recourse function. By Minkowski's finite basis theorem [41], interior points can be most easily obtained by using convex combinations of the extreme points.

3.4 Optimization

Once the crew-compatible allocations (CCA) in the first stage are obtained, random scenarios are generated, and a deterministic second-stage FAM problem was solved for each CCA and scenario. The two-stage optimization model was presented in Section 3.2. An estimate of the revenue was calculated as an average over the scenarios. The objective of the second stage is to maximize the profit (Revenue - Cost) of assigning aircraft to each flight leg in the schedule. For calculating the revenue, average fares per passenger are considered and multiplied with the minimum of demand and capacity for each scenario. Passengers spilled (customers who cannot be accommodated due to insufficient capacity) are assumed to be lost revenue and are not captured.

The solution of FAM can be improved by incorporating the effects of Revenue Management (RM), and using the average passenger revenues tends to cause FAM to under-estimate revenues for flights with high nominal load factors. Integrating FAM with RM will be considered for future research, as the objective here is to demonstrate the use of DACE and to fit a recourse function for the two-stage SP FAM problem.

3.4.1 Scenarios

The purpose of scenarios is to provide estimation of an expected value, which is equivalent to numerical integration. The demand scenarios for each flight are generated based on a normal distribution assuming that the mean and variance are known. Since an average of the scenarios is taken in the second stage as the estimate, one would expect the performance of the estimation in the second stage to improve as the number of scenarios is increased. Even though the computation time required to solve the problem is not of particular interest, we can observe that the solution time required to solve the second stage grows rapidly with the number of scenarios.

One way to mitigate this problem is to use confidence level information to calculate an upper bound for the number of scenarios. Suppose $\bar{y}$ is used as an estimate of average revenue (μ). Then we can be 100(1 - α)% confident that the error ($|\bar{y} - \mu|$) will not exceed a specified amount E when the sample size is

$$\left\lceil \left(\frac{2\ z_{\alpha/2}\ \sigma}{E} \right)^2 \right\rceil ,$$

where $2E$ is the length of the resulting confidence interval and α is the significance level.

Algorithm (1) shows the approach to optimize the second-stage FAM problem.

Algorithm 1 Optimization of second stage

```
Initialize the number of design points ← N
Initialize the number of scenarios ← Ξ
Let G be the number of crew-compatible families
for all n_d ∈ N design points do
   Get Design Point (first-stage Solution – Initial CCA).
   for all ξ ∈ Ξ scenarios do
      Get Demand Values and Calculate Revenue.
      for all g ∈ G crew groups do
         Read Problem.
         Solve Problem.
         Get Objective value.
      end for
      Total Revenue ← (Total Revenue + Objective value).
   end for
   E[Revenue] ← (Total Revenue / Number of scenarios).
end for
```

3.5 MARS

MARS (Multivariate Adaptive Regression Splines) was developed by Friedman [42] as a statistical modeling method for estimating a completely unknown relationship between a single response variable and several covariates. The MARS model is selected in a two-phase process. In the first phase, a model is grown by adding basis functions (new main effects, knots, or interactions) until an overly fitted model is obtained. In the second phase, basis functions are deleted in order of least contribution to the model until a balance of bias and variance is found. Both the variables to use and the end points of the intervals for each variable (referred to as knots) are found via greedy search procedures. Variables, knots, and interactions are considered simultaneously by evaluating a “Lack of Fit" (LOF) criterion. MARS chooses the LOF that most improves the model at each step.

The run time for MARS is dependent upon the number of basis functions that the user specifies with the parameter M_{max}. A variant of MARS that automatically selects M_{max}, developed by Tsai et al [29], is used to decrease the time required to fit the approximation.

3.6 RESULTS

We applied the methodology described in the above sections on a real time airline carrier with a weekly schedule containing 50 stations, 2358 legs, and 6537 variables. Seven fleet types and four crew-compatible families were considered. The turn time was taken as 29 minutes and was assumed constant irrespective of the crew-compatible family. This assumption can be relaxed to make the problem more realistic but will make the problem more complex. Based on the ground time for each station and the traffic intensity, six hubs were identified.

In order to use design for generating the first-stage crew-compatible allocation, initially the reduced state space was obtained using the approach presented in Section 3.3.1. As discussed in Section 3.3.1.1, the Boeing concept of D^3 was used to create the cover constraints with only crew-compatible families as variables. The obtained cover constraints were used to pivot one variable for each flight as mentioned in Section 3.3.1.2.1. The balance constraints were generated for each station using a time line as discussed in Section 3.3.1.2.2 and the equalities obtained were used to pivot more variables. The plane count constraints were added to the model to maintain feasibility of the first-stage solution. Finally, after reducing the model to the least number of variables using the equalities in the operational constraints, the model was checked for implicit equalities.

In our first-stage problem, we did not find any implicit equalities. The results of the first-stage dimensionality reduction are presented in Table 3.3.

Table 3.3. Computation Results

Total number of variables	6537
Variable reduction due to	
Demand Driven Dispatch (D^3)	2441
Cover Constraint	2358
Balance Constraint	474
Implicit Equalities	0
Remaining variables	1264

The obtained 1264 variables represented 972 legs, with the corresponding cover constraints forming a polytope. A Latin hypercube design was generated based on the cover constraints, similar to the methodology discussed in Section 3.3.2. Out of the 972 legs, the cover constraints of 680 legs had one variable and the remaining 292 legs had two variables. Since $n = 972$, $m = 1$ or 2, and N should be greater than 1264 (680*1 + 292*2). We considered $N = 1980$ design points, with three levels for 680 legs, and six levels for 292 legs as shown below:

$p_1 = \{0, 0.5, 1\}$ (corresponding to three levels)

$p_2 = \{(0,0), (0,1), (1,0), (0, 0.5), (0.5, 0), (0.5, 0.5)\}$ (corresponding to six levels). These six levels correspond to the six centers of the circles as shown in Figure (3.6).

The generated Latin hypercube design was mapped to the corresponding levels, which resulted in 1980 first-stage decision values. But none of these satisfied the first-stage operational constraints. As discussed in 3.3.2.2, the 1980 infeasible points (x^0) were projected on to the feasible polytope formed by the cover constraints which resulted in 141 points (more than one infeasible point getting projected to the same point, $\tilde{x}$). Since the optimal solution lies at an extreme point, nearby extreme points were obtained by considering the gradient of the L2 norm and using one iteration of CPLEX. This resulted in 1525 extreme points (x^*). The spread of these points with respect to the original infeasible DoE points was measured using the ratio

$$\frac{||x^0 \; - \; x^*||}{||x^*||} \tag{3.68}$$

and the graph is shown in Figure 3.7. There are chances of some corners in the polytope being left behind during this process as there are a total of $\binom{3695}{1264}$ combinations of extreme points, and it is impossible to take into account all of these. Considering that there are no breaks in the graph, we expect the points to be evenly spread across the polytope.

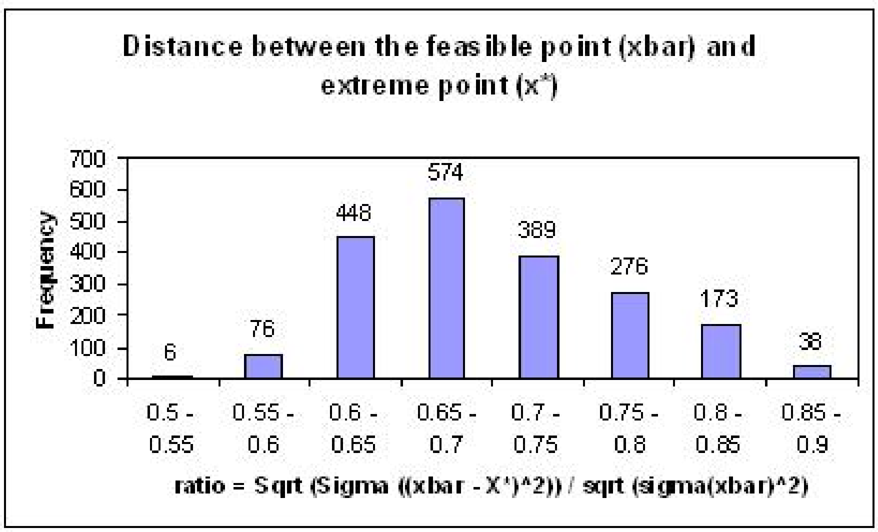

Figure 3.7. Histogram to check the spread of points.

Interior points were obtained using convex combinations of these 1525 extreme points. Thus, around 3562 total first-stage assignments were obtained, which included 141 feasible projections, 1525 extreme points and 1896 interior points.

In order to solve the second stage, demand scenarios were generated using a normal distribution with known mean and standard deviation values for each itinerary-fare class. To calculate the number of scenarios required to estimate the expected revenue value, initially 20 scenarios were considered and the revenue values were obtained for five different design points. A box plot comparing the revenue values for the five design points is presented in Figure 3.8. As discussed in Section 3.4, using the confidence interval information for each design point, and with the assumption that the error should be approximately 0.1% of the average revenue, we obtained the sample sizes for each of the design points in the range of 35 to 39. An upper bound for the sample size of 40 was finally taken. For each design point and for each scenario, the second stage LP, as discussed in Section 3.2, was solved, and the expected revenue was calculated as the average over the scenarios.

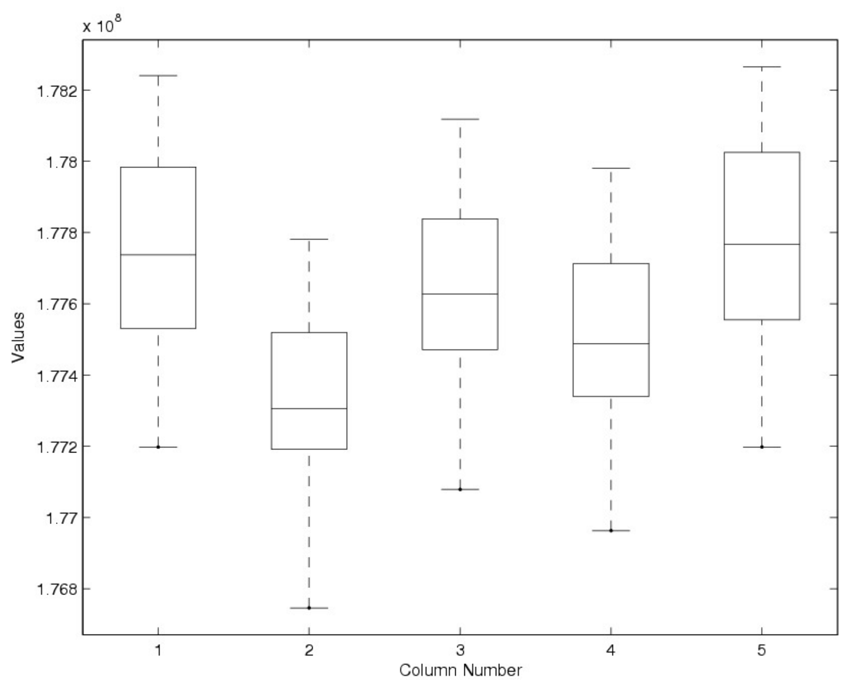

Figure 3.8. Boxplot comparing revenue values for five design points.

A MARS function with an automatic stopping rule as discussed in Tsai and Chen [29] was used, with average revenue as "y" values and the initial CCA as "x" values.

The fit resulted in 84 basis functions with an R^2 value equal to 99.459. A new set of "x" values was generated (utilizing the convex combinations of 1525 extreme points) to evaluate the MARS function, and relative errors were calculated using the formula

$$\frac{|y - \hat{y}|}{|y|}. \tag{3.69}$$

A box plot representing the relative errors is shown in Figure 3.9. The maximum relative error was obtained as 4.63 x 10^{-5} and the median error was 9 x 10^{-6}, which are really small. A new set of extreme points were generated to evaluate the MARS function and the relative error was of the order of 10^{-3}.

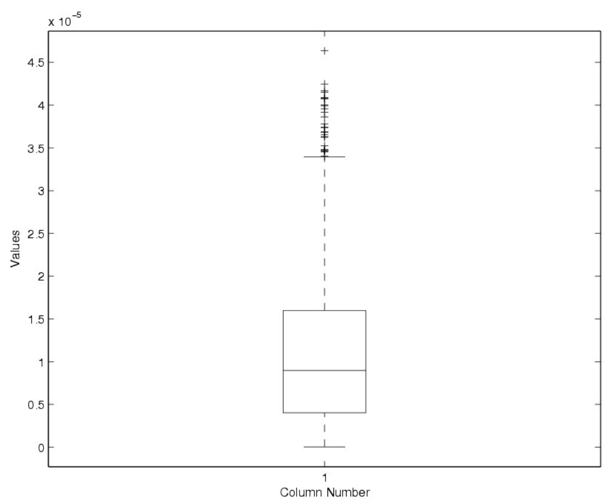

Figure 3.9. Boxplot for checking the variance of relative errors.

CHAPTER 4

MARS OPTIMIZATION

Chapter 4 focuses on optimizing the recourse function obtained in the previous chapter so as to generate a single first-stage solution. In this aspect, gradients of the MARS function are utilized to generate revenue cuts. This is similar to cutting plane algorithms, which develop a series of ever-improving approximating linear programs, whose solutions converge to the solution of the original problem. The revenue generated from using our approach is compared to the traditional Benders' approximation method utilized to solve a two-stage SP problem. An introduction to cutting plane methods is given in Section 4.1 followed by the L-shaped method in Section 4.2. Section 4.2.2 illustrates the methodology to solve the two-stage SP FAM problem using a Benders' approach, and Section 4 illustrates the variant of the cutting plane method used to solve the two-stage FAM problem with a MARS approximation to the recourse function.

4.1 Cutting Plane Methods

Cutting plane methods are applied to solve convex and quasi-convex optimization problems of the form:

$$\begin{aligned} &\min\ c^T \mathrm{x} \\ &\text{s.t.}\ \ \mathrm{x} \in S, \end{aligned} \tag{4.1}$$

where $S \subset \{\mathrm{x} \in R^n\text{: } \mathrm{ax} \leq \mathrm{b}\}$ is a closed convex set. Cutting-planes are hyperplanes that separate the current point from the optimal points and eliminate the half-space from our search. These are very useful for problems with a very large number of constraints. Algorithm 2 represents the general form of a cutting-plane algorithm for the problem defined in (4.1). The process is represented in Figure 4.1, where we can see that the hyperplane

Algorithm 2 Cutting Plane Algorithm

Given a polyhedron $P_k \supset S$
Step 1: Minimize $\mathrm{c}^T\mathrm{x}$ over P_k to obtain a point x_k in P_k.
if $\mathrm{x}_k \in S$ **then**
  Stop. x_k is optimal.
else
  Go to Step 2.
end if
Step 2: Find a hyperplane H_k separating the point x_k from S, that is, find $\mathrm{a}_k \in R^n, b_k \in R^1$
  such that $S \subset \{\mathrm{x}{:}\mathrm{a}_k^T\mathrm{x} \leq b_k\}$, $\mathrm{x}_k \in \{\mathrm{x}{:}\mathrm{a}_k^T\mathrm{x} > \mathrm{b}_k\}$.
  Update $P_{k+1} \leftarrow P_k \cap \{\mathrm{x}{:}\mathrm{a}_k^T\mathrm{x} \leq b_k\}$.
  Go to Step 1.

H_1 eliminates the point x_1 to generate the solution x_2, followed by the hyperplane H_2 to eliminate the point x_2 to generate the solution x_3. This process continues until an optimal point $\mathrm{x}_k \in S$ is obtained.

Different cutting plane methods have been described in Luenberger [43], Boyd and Vandenberghe [44]. These algorithms mainly differ in the manner in which the hyperplane that separates the current point x_k from the constraint set S is selected. This selection is important, as it defines the effectiveness of the cut, i.e., how much improvement there is in the objective value, and hence defines the convergence of the method. Specific algorithms also differ with respect to the manner by which the polyhedron is updated once the new hyperplane is determined. A simple procedure is to adjoin the linear inequality associated with the hyperplane to the ones determined previously. This yields the best possible updated approximation to the constraint set, but requires a large number of iterations because of the huge number of inequalities expressing the approximation. Thus, in some algorithms, older inequalities which are not binding at the current point are discarded.

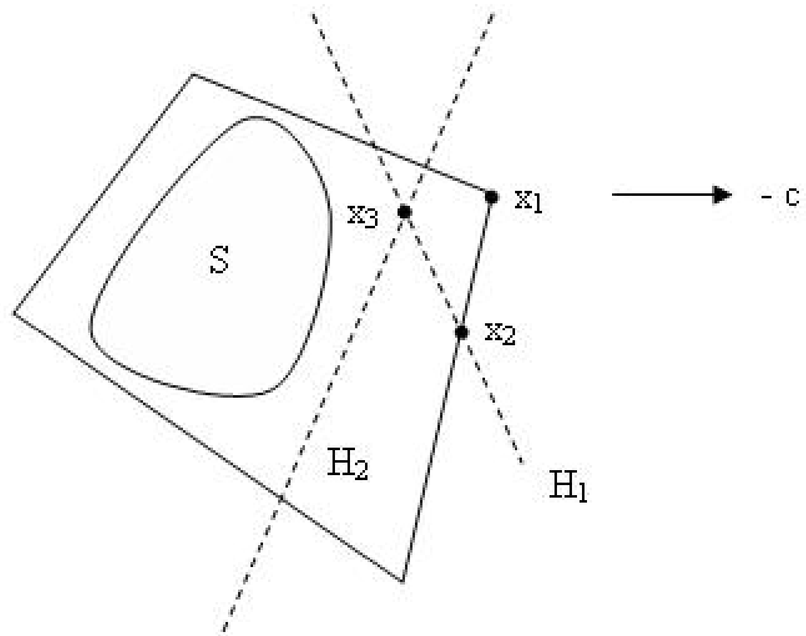

Figure 4.1. Cutting Plane Method.

If the closed convex set S in equation 4.1 is represented by $g(\mathrm{x}) \leq 0$, where $g(\mathrm{x}) \in R^p$ is convex and differentiable, then we can write equation 4.1 as:

$$\min \; c^T x \tag{4.2}$$

$$\text{s.t. } g(x) \leq 0.$$

Using Taylor series expansion, we have

$$g_i(x)|_{x=w} = g_i(w) + (x-w)^T \; \nabla g_i(w) + O(||x-w||^2). \tag{4.3}$$

Since $g_i(x)$ is convex and differentiable, we have

$$g_i(x) \geq g_i(w) + (x-w)^T \; \nabla g_i(w) \quad \forall w. \tag{4.4}$$

Equation (4.4) represents the cutting plane that will be added to the constraint set $g(x)$, and the optimization problem (4.2) is solved. This method is called *Kelley's cutting plane method* [45], and it is presented in Algorithm 3. Figure 4.2 illustrates the method.

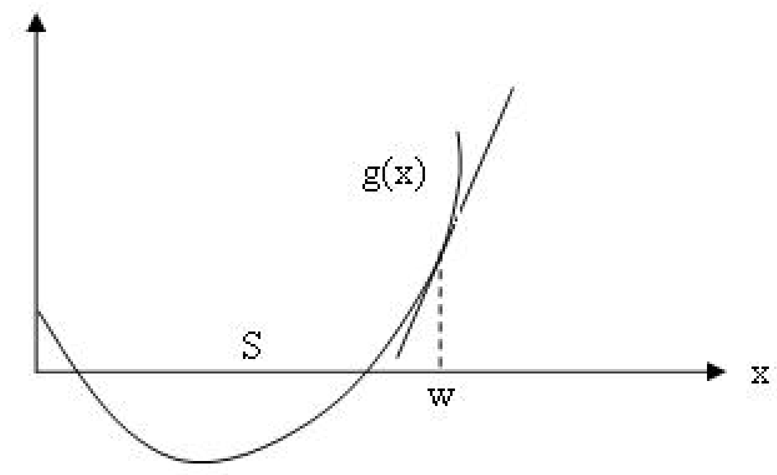

Figure 4.2. Convex Cutting Plane.

Algorithm 3 Kelley's Cutting Plane Algorithm

Let $S = \{\text{x}:g_i(\text{x}) \leq 0\}$ and let polyhedron $P \supset S$, such that c^Tx is bounded on P.
Step 1: Minimize c^Tx over P to obtain a point x = w.
if $g_i(\text{w}) \leq 0$ **then**
  Stop. w is an optimal solution.
else
  Go to Step 2.
end if
Step 2: Let i be an index maximizing $g_i(\text{w})$. Define the new polyhedron as: $P \leftarrow P \cap \{\text{x}:g_i(\text{w}) + \nabla g_i^T(\text{w})(\text{x - w}) \leq 0\}$. Go to Step 1.

Although cutting plane algorithms do not have good convergence properties, they are easy to implement. As mentioned in Luenberger [43], these algorithms converge arithmetically.

i.e., if x^* is optimal, then they converge at the rate of $||x_k - x^*||^2 \leq c/k$ for some constant c.

4.2 L-Shaped Method

The L-shaped method can be interpreted as a cutting plane method. In Chapter 1, we discussed the basics of two-stage SP and its formulation. As defined, the deterministic equivalent of two-stage stochastic linear problem with recourse ($SLPwR$) is:

$$\max\ z = c^T x + \Im(x) \tag{4.5}$$

$$\text{s.t } Ax = b \tag{4.6}$$

$$x \geq 0 \quad . \tag{4.7}$$

where $\Im(x)$ represents the recourse function defined by

$$\Im(x) = E_\xi[Q(x, \xi(\omega))], \tag{4.8}$$

and $Q(x, \xi(\omega))$ is obtained from the second-stage problem given by:

$$Q(x, \xi(\omega)) = \min_y \{q(\omega)^T y | Wy = h(\omega) - T(\omega)x, y \geq 0\}. \tag{4.9}$$

where q is a coefficient vector and W, h and T are coefficient matrices which depend on the random variable $\xi(\omega)$. The matrix W is known as the recourse matrix. We can see that the two-stage formulation defined in equations (4.5) - (4.7) has linear constraints and a convex objective function. Consequently, it can be solved using algorithms with the cutting plane methods discussed earlier. However the $SLPwR$ problems are difficult to solve because their recourse functions ($\Im(x)$) and sub-gradients are very difficult to evaluate. Some of the difficulties can be avoided when the recourse structure is simple, as in problems with fixed recourse and complete recourse. Fixed recourse means that the recourse matrix is independent of random vector ξ, whereas complete recourse means that any set of values that we choose for the first-stage decisions, x, leaves us with a feasible second-stage problem.

The method most frequently used in building an outer linearization of the recourse function and finding a solution to the first-stage problem is called the L-shaped method. As defined in Birge and Louveaux [7], the basic idea of the L-Shaped method is to approximate the recourse term in the objective of these problems. Since the value of the recourse function is a solution to second-stage recourse linear programs, we can just solve it as a subproblem. To make this approach possible, we assume that the random vector ξ has finite realizations. Let $k = 1, ..., K$ be the index for possible realizations and let p_k be their probabilities. Under this assumption, we can write the deterministic equivalent program in the extensive form. This form is created by associating one set of second-stage decisions, say, y_k, to each realization ξ, i.e., to each realization of q_k, h_k, and T_k. It is a large scale linear problem that we can define as the *extensive form* (EF):

$$\text{(EF)} \min c^T x + \sum_{k=1}^{K} p_k q_k^T y_k \tag{4.10}$$

$$\text{s.t } Ax = b, \tag{4.11}$$

$$T_k x + W y_k = h_k, k = 1, ..., K \tag{4.12}$$

$$x \geq 0, y_k \geq 0, k = 1, ..., K \tag{4.13}$$

The block structure of the extensive form appears in Figure 4.3. Taking the dual of the extensive form, one obtains a dual block angular structure, as in Figure 4.4. Therefore it seems natural to exploit this dual structure by performing a Dantzig-Wolfe decomposition. This method has been extended in SP to take care of feasibility questions and is known as Van Slyke and Wets's L-shaped method, and is presented in Algorithm 4.

Algorithm 4 L-Shaped Algorithm

Step 0: Set $r = s = v = 0$.
Step 1: Set $v = v + 1$. Solve the linear program

$$\min z = c^T x + \theta \tag{4.14}$$
$$\text{s.t. } Ax = b, \tag{4.15}$$
$$D_l x \geq d_l, \qquad l = 1, ..., r, \tag{4.16}$$
$$E_l x + \theta \geq e_l, \qquad l = 1, ..., s, \tag{4.17}$$
$$x \geq 0, \theta \in \Re. \tag{4.18}$$

Let (x^v, θ^v) be an optimal solution. For the first iteration, we do not have any optimality cut. As such we do not require constraint set (4.16) or (4.17), and θ^v is set to $-\infty$ and not considered for the calculation of x^v.
Step 2. For $k = 1, ..., K$ solve the linear program

$$\min w^{'} = e^T v^+ + e^T v^- \tag{4.19}$$
$$\text{s.t. } Wy + Iv^+ - Iv^- = h_k - T_k x^v, \tag{4.20}$$
$$y \geq 0, v^+ \geq 0, v^- \geq 0. \tag{4.21}$$

where e^T= (1,...1), until, for some k, the optimal value $w^{'} > 0$. In this case, let σ^v be the associated simplex multipliers and define

$$D_{r+1} = (\sigma^v)^T T_k \quad \text{and} \quad d_{r+1} = (\sigma^v)^T h_k \tag{4.22}$$

to generate a constraint (called a feasibility cut) of type (4.16). Set $r = r + 1$, add to the constraint set (4.16), and return to Step 1. If for all k, $w^{'} = 0$, go to Step 3.
Step 3. For $k = 1, ..., K$ solve the linear program

$$\begin{aligned} \min w &= q_k^T y \\ \text{s.t. } Wy &= h_k - T_k x^v, \\ y &\geq 0. \end{aligned} \tag{4.23}$$

Let π_k^v be the simplex multipliers associated with the optimal solution of Problem k of type (4.23). Define

$$E_{s+1} = \sum_{k=1}^{K} p_k.(\pi_k^v T_k) \quad \text{and} \quad e_{s+1} = \sum_{k=1}^{K} p_k.(\pi_k^v h_k). \tag{4.24}$$

Let $w^v = e_{s+1} - E_{s+1} x^v$. If $\theta^v \geq w^v$, stop; x^v is an optimal solution. Otherwise, set $s = s + 1$, add to the constraint set (4.17) and return to Step 1.

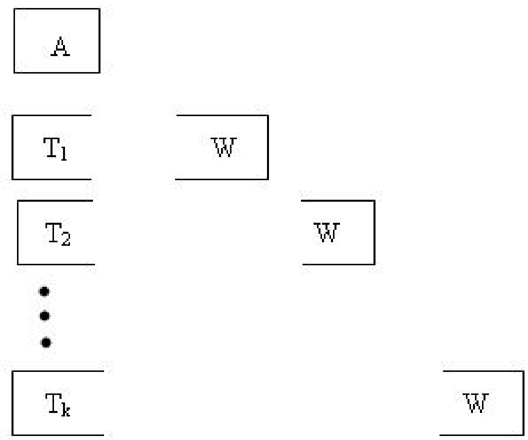

Figure 4.3. Block structure of the two-stage extensive form.

4.2.1 Feasibility and Optimality Cuts

There are two types of cuts which are added during the solution procedure: feasibility cuts and optimality cuts. A feasibility cut is a linear constraint that ensures that a first-stage decision is second-stage feasible. A complete recourse problem does not need any feasibility cuts i.e., any decision taken during the first stage will leave us with a second-stage problem that is feasible.

$$\begin{aligned} \min \ & c^T x + Q(x) \\ \text{s.t. } & x \in (K_1 \cap K_2), \end{aligned} \tag{4.25}$$

where the set K_1 contains possible decisions that satisfy the constraints of the first stage, and the set K_2 contains those decisions that are second-stage feasible, i.e., $K_1 = \{x|Ax = b, x \geq 0\}$ and $K_2 = \{x|Q(x) \leq \infty\}$.

On the other hand, an optimality cut is a linear approximation of $Q(x)$ on its domain of finiteness, and is determined based on the dual of the second-stage problem.

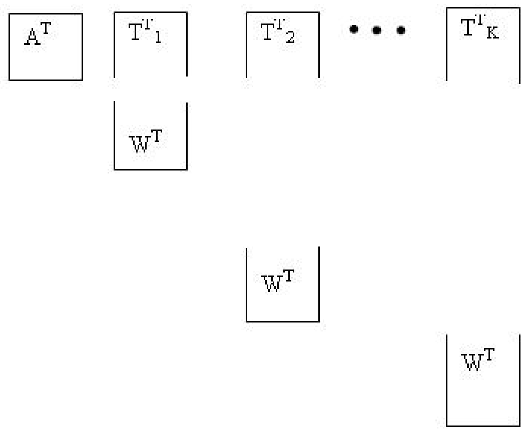

Figure 4.4. Block angular structure of the two-stage dual.

As such, each optimality cut provides a lower bound on $Q(x)$. The dual of the second-stage problem defined by equation 4.9 is given by:

$$\max \pi^T(h - Tx) \tag{4.26}$$

$$\text{s.t. } \pi^T W \leq q, \tag{4.27}$$

where the dual variable π corresponds to the Lagrange multipliers of the problem given by equation (4.9). Sections 4.2.1.1 and 4.2.1.2 show how to generate optimality cuts and feasibility cuts, respectively.

4.2.1.1 Optimality Cuts

For generating the l-th optimality cut, using the L-shaped method, consider the v-th iteration of the algorithm at the k-th realization of the random variable for $SLPwR$. Using the duality theorem, the value of the objective function of the second-stage problem would be given by:

$$Q(x^v, \xi(\omega)^k) = (\pi_k^v)^T(h_k - T_k x^v) \tag{4.28}$$

and therefore, because of convexity:

$$Q(x, \xi(\omega)^k) \geq (\pi_k^v)^T (h_k - T_k x). \tag{4.29}$$

If we consider now a discrete distribution for $\xi(\omega)$ and assume that the probability for the k-th realization of $\xi(\omega)$ is p_k, then the expected value of the objective function given by equation (4.28) is:

$$\Im(x^v) = E_\xi[Q(x^v, \xi(\omega)^k)] = E_\xi[(\pi_k^v)^T (h_k - T_k x^v)] = \sum_{k=1}^{K} p_k[(\pi_k^v)^T (h_k - T_k x^v)]. \tag{4.30}$$

Hence, also because of convexity:

$$Q(x) \geq \sum_{k=1}^{K} p_k[(\pi_k^v)^T (h_k - T_k x^v)]. \tag{4.31}$$

Finally, by defining

$$e_l = \sum_{k=1}^{K} p_k (\pi_k^v)^T h_k \tag{4.32}$$

$$E_l = \sum_{k=1}^{K} p_k (\pi_k^v)^T T_k, \tag{4.33}$$

and substituting e and E in equation (4.31), we get the optimality cut as:

$$E_l x + \theta \geq e_l \tag{4.34}$$

where $\theta = Q(x)$ is the objective of the recourse function.

4.2.1.2 Feasibility Cuts

A first-stage decision x^v is second-stage feasible if a finite vector v exists such that the constraints of the second stage are satisfied. As mentioned before, the role of a feasibility cut is to ensure that a first-stage decision is second-stage feasible. For this we can solve the phase 1 simplex problem as shown below:

$$z = \min e^T(v^+ + v^-) \tag{4.35}$$

$$\text{s.t. } Wv + v^+ - v^- = h - Tx^v, \tag{4.36}$$

$$v \geq 0, v^+ \geq 0, v^- \geq 0, \tag{4.37}$$

or its equivalent dual:

$$z = \max \ \sigma^T(h - Tx_v) \tag{4.38}$$

$$\text{s.t. } \sigma^T W \leq 0 \tag{4.39}$$

$$|\sigma| \leq e, \tag{4.40}$$

where e is a vector of ones. Clearly, $z \geq 0$. If $z = 0$, it means that the second-stage constraints are satisfied and, therefore, x^v is second-stage feasible. However , if $z > 0$, then the original second-stage problem is infeasible (and its dual is unbounded). Therefore, in order to ensure the feasibility of the second-stage problem, the constraint:

$$(\sigma^v)^T(h - Tx) \leq 0, \tag{4.41}$$

must be added. If for some k-th realization of the random variable the second-stage problem is infeasible, then, according to equation (4.41), we define:

$$D_l = (\sigma^v)^T T_x, \quad d_l = (\sigma^v)^T h_k, \tag{4.42}$$

in order to obtain the l - th feasibility cut:

$$D_l x \geq d_l. \tag{4.43}$$

Figure 4.5 [46] shows a pictorial representation on how the cuts are generated.

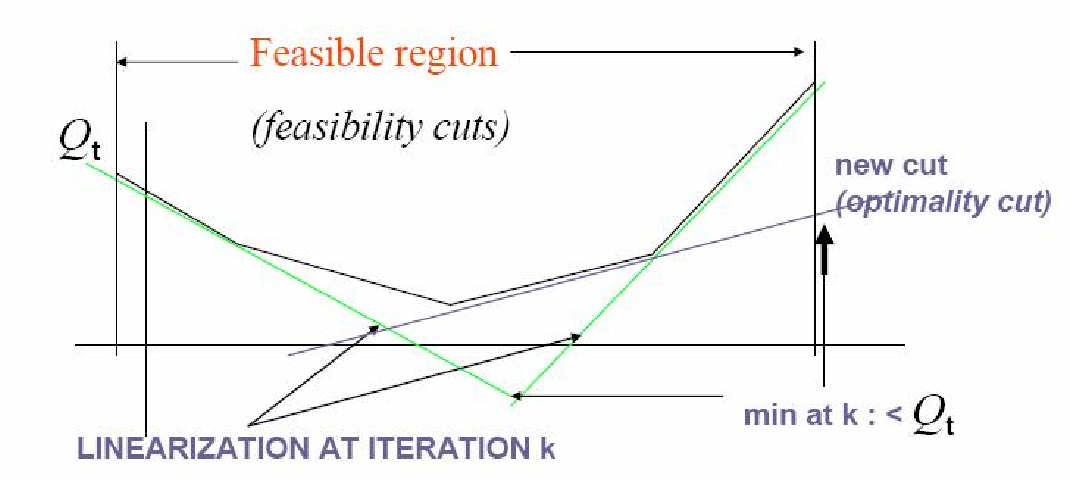

Figure 4.5. Traditional Benders' Approach.

4.2.2 Benders' Approach for Two-Stage FAM Problem

The two-stage SP framework for the fleet assignment problem was presented in Chapter 3. The objective function and the constraints form a block angular structure, and the standard Benders' approach can be applied to solve the problem. The master problem makes a crew-compatible assignment, and each recourse problem solves for a particular assignment within the family assigned in the first stage.

The restricted master problem (RMP) for the two-stage FAM is represented as:

$$\max\ \eta \tag{4.44}$$

$$\text{s.t. } a_{v^-} + \sum_{l \in A_v} x_{gl} - \sum_{l \in D_v} x_{gl} - a_{v^+} = 0 \qquad \forall v \in V, \tag{4.45}$$

$$\sum_{g \in G} x_{gl} = 1 \qquad \forall g \in G, \forall l \in L \tag{4.46}$$

$$\sum_{o \in O_g} a_o + \sum_{l \in L_0} x_{gl} \leq M_g \qquad \forall g \in G, \tag{4.47}$$

$$x_{gl} \in \{0, 1\} \qquad \forall l \in L, g \in G \tag{4.48}$$

$$a_{v^+} \geq 0 \qquad \forall v \in V \tag{4.49}$$

$$\eta \leq \text{Recourse Value}(x_g), \tag{4.50}$$

where η represents the objective value with respect to the first-stage problem and θ represents the second-stage recourse function value. Constraints (4.45) - (4.47) are the first-stage constraints. Integer assignments in the first stage are represented by constraint (4.48) and constraint (4.49) make sure that the ground arcs are non-negative. Constraint (4.50) is the optimality condition which has to be satisfied to reach the optimal solution and has to be replaced with linear Benders' cut.

The primal subproblem for the two-stage FAM (SPFAM) is represented as:

$$\max\ \theta\ = E\left[-\sum_{l\in L}\sum_{f\in F} C_{fl}(x_{fl}^{\xi}) + \sum_{i\in I} f_i z_i^{\xi}\right] \tag{4.51}$$

$$\text{s.t. } a_{v^-}^{\xi} + \sum_{l\in A_v} x_{f(v)l}^{\xi} - \sum_{l\in D_v} x_{f(v)l}^{\xi} - a_{v^+}^{\xi} = 0 \qquad \forall v \in V, \xi \in \Xi \tag{4.52}$$

$$\sum_{f\in g} x_{fl}^{\xi} = \bar{x}_{gl} \qquad \forall l \in L, g \in G, \xi \in \Xi \tag{4.53}$$

$$\sum_{o\in O_f} a_o^{\xi} + \sum_{l\in L_0} x_{fl}^{\xi} \le M_f \qquad \forall f \in F, \xi \in \Xi \tag{4.54}$$

$$\sum_{i\ni I} z_i^{\xi} - \sum_{f\in F} Cap_f\ x_{fl}^{\xi} \le 0 \qquad \forall l \in L, \xi \in \Xi \tag{4.55}$$

$$0 \le z_i^{\xi} \le D_i^{\xi} \qquad \forall i \in I, \xi \in \Xi \tag{4.56}$$

$$x_{fl}^{\xi} \in \{0,1\} \qquad \forall l \in L, f \in F, \xi \in \Xi \tag{4.57}$$

$$a_{v^+}^{\xi} \ge 0 \qquad \forall v \in V, \xi \in \Xi. \tag{4.58}$$

The subproblem can be decomposed by each scenario and solved similar to the standard L-shaped method, and finally an average over the scenarios is calculated to get the recourse function value θ. Let γ_v^{ξ}, π_{gl}^{ξ}, ρ_f^{ξ}, δ_l^{ξ} and μ_i^{ξ} be the duals corresponding to the constraints (4.52) – (4.56) for a random realization ξ, respectively. For a given ξ and $\bar{x}_{gl}$, the dual of the subproblem is:

$$\min\ \sum_{l\in L}\sum_{g\in G}\sum_{\xi\in\Xi} \pi_{gl}^{\xi}\bar{x}_{gl} + \sum_{f\in F}\sum_{\xi\in\Xi} \rho_f^{\xi} M_f + \sum_{i\in I}\sum_{\xi\in\Xi} \mu_i^{\xi} D_i^{\xi} \tag{4.59}$$

$$\text{s.t. } \Delta, \tag{4.60}$$

where Δ represents the polyhedron formed by the dual constraints.

The following inequality has to be satisfied at optimality as defined by constraint (4.50):

$$\eta \le \sum_{l\in L}\sum_{g\in G}\sum_{\xi\in\Xi} \pi_{gl}^{\xi}\bar{x}_{gl} + \sum_{f\in F}\sum_{\xi\in\Xi} \rho_f^{\xi} M_f + \sum_{i\in I}\sum_{\xi\in\Xi} \mu_i^{\xi} D_i^{\xi}; \tag{4.61}$$

otherwise, the master problem must be re-solved with an added Benders' optimality cut. The optimality cut is represented as:

$$\eta \le \sum_{l \in L} \sum_{g \in G} \sum_{\xi \in \Xi} \pi^{\xi}_{gl} x_{gl} + \sum_{f \in F} \sum_{\xi \in \Xi} \rho^{\xi}_{f} M_f + \sum_{i \in I} \sum_{\xi \in \Xi} \mu^{\xi}_{i} D^{\xi}_{i}. \tag{4.62}$$

The last two terms on the right hand side are constant, and they can be determined for any particular ξ, once the dual values $\pi^{\xi}_{gl}, \rho^{\xi}_{f}$ and μ^{ξ}_{i} are known. Let $\bar{\theta}$ be the average recourse function value for all scenarios (ξ), then the last two terms can be calculated as:

$$\sum_{f \in F} \sum_{\xi \in \Xi} \rho^{\xi}_{f} M_f + \sum_{i \in I} \sum_{\xi \in \Xi} \mu^{\xi}_{i} D^{\xi}_{i} = \bar{\theta} - \sum_{l \in L} \sum_{g \in G} \sum_{\xi \in \Xi} \pi^{\xi}_{gl} \bar{x}_{gl}. \tag{4.63}$$

Now the Benders' optimality cut represented by equation (4.62) can be modified as:

$$\eta - \sum_{l \in L} \sum_{g \in G} \sum_{\xi \in \Xi} \pi^{\xi}_{gl} x_{gl} \le \bar{\theta} - \sum_{l \in L} \sum_{g \in G} \sum_{\xi \in \Xi} \pi^{\xi}_{gl} \bar{x}_{gl}. \tag{4.64}$$

Given this, the original RMP can be reformulated as:

$$\max \ \eta \tag{4.65}$$

$$\text{s.t. } a_{v^-} + \sum_{l \in A_v} x_{gl} - \sum_{l \in D_v} x_{gl} - a_{v^+} = 0 \qquad \forall v \in V, \tag{4.66}$$

$$\sum_{g \in G} x_{gl} = 1 \qquad \forall g \in G, \forall l \in L \tag{4.67}$$

$$\sum_{o \in O_g} a_o + \sum_{l \in L_0} x_{gl} \le M_g \qquad \forall g \in G, \tag{4.68}$$

$$\eta - \sum_{l \in L} \sum_{g \in G} \sum_{\xi \in \Xi} \pi^{\xi}_{gl} x_{gl} \le \bar{\theta} - \sum_{l \in L} \sum_{g \in G} \sum_{\xi \in \Xi} \pi^{\xi}_{gl} \bar{x}_{gl} \tag{4.69}$$

$$x_{gl} \in \{0, 1\} \qquad \forall l \in L, g \in G \tag{4.70}$$

$$a_{v^+} \ge 0 \qquad \forall v \in V. \tag{4.71}$$

In our approach, all the first-stage constraints have been added to the RMP. As such, the complete recourse of the problem is maintained, and no feasibility cuts are needed.

If not all the first-stage constraints are added to the RMP, then the first-stage decision obtained might not be second-stage feasible. In that case, feasibility cuts can be added to the RMP. In general, for a random realization ξ_i, the feasibility cut for the two-stage FAM is represented as:

$$\sum_{l\in L}\sum_{g\in G}\pi_{gl}^{\xi_i}x_{gl} + \sum_{f\in F}\rho_f^{\xi_i}M_f + \sum_{i\in I}\mu_i^{\xi_i}D_i^{\xi} \geq 0. \tag{4.72}$$

The L-shaped method for the two-stage FAM with complete recourse is presented in Algorithm 5.

Algorithm 5 Benders' decomposition algorithm for Two-stage FAM

Step 0: Set $s = v = 0$, $\eta^v \leftarrow \infty$ and let x^v be an intital feasible first-stage assignment. Let $\xi \in \Xi$ be the set of scenarios indexed by k. Go to Step 2.

Step 1: Set $v = v + 1$. Solve the RMP and let (x^v, η^v) be the solution.

Step 2:

for all $k = 1, ..., K$ **do**

Solve the subproblem SPFAM to obtain the recourse function value θ_k^v and the simplex multiplier $\pi_{gl}^{\xi_k}$.

end for

Calculate:

$$\theta^v = \sum_{k\in K} p_k\theta_k^v \tag{4.73}$$

$$\pi_{gl}^{\xi} = \sum_{k\in K}\sum_{f\in g} p_k\pi_{fl}^{\xi_k} \tag{4.74}$$

where p_k is the probability associated with the k^{th} scenario.

if $\eta^v \leq \theta^v$ **then**

Stop. x^v is the optimal first-stage solution.

else

Generate Optimality cut (4.64) and add to RMP. $s = s + 1$. Go to Step 1.

end if

4.2.2.1 Scenario Generation

Discrete distributions must be used to represent the scenarios in the SP problems. The stochasticity of the demand is incorporated by generating demand scenarios for each itinerary-fare class, similar to method discussed in Section 3.4.1. There are two major issues:

- The number of scenarios must be small enough for the stochastic program to be solvable.
- The number of scenarios must be large enough to represent the underlying distribution or data adequately.

The number of scenarios can be determined based upon the convergence of the recourse function value. This can be illustrated based upon the optimality cut that is added in each iteration of the L-shaped method. For a given first-stage decision x^0_{gl} and RMP objective value η^0, we can modify equation (4.64) as:

$$\eta^0 - \sum_{l\in L}\sum_{g\in G}\sum_{\xi\in\Xi}\pi^{\xi}_{gl}x^0_{gl} \le \bar{\theta} - \sum_{l\in L}\sum_{g\in G}\sum_{\xi\in\Xi}\pi^{\xi}_{gl}x^0_{gl} \tag{4.75}$$

$$\Rightarrow \eta^0 \le \bar{\theta}. \tag{4.76}$$

4.3 Convex Cutting Plane Method for MARS

For the two-stage SP fleet assignment problem, the constraints are all linear, and under the assumption that the MARS approximation ($\hat{y}(x)$) obtained in Chapter 3 is concave, a variant of Kelley's cutting plane method can be used to solve the problem. As defined in Section 4.2.2, the two-stage FAM problem is represented as:

$$\max\ \eta \tag{4.77}$$

$$\text{s.t. } x \in P, \tag{4.78}$$

$$\eta \le \hat{y}(x), \tag{4.79}$$

where P represents the polytope formed by the first-stage constraints defined by equations (4.45) – (4.49), and constraint (4.79) represents the optimality criteria that has to be met to obtain the first-stage solution. As mentioned in Chen et al [47], the MARS function is continuous and differentiable at different values of x.

If the MARS function is concave, then a Taylor series may be used at any point $\bar{x}$ to represent equation (4.79) as:

$$\eta \leq \hat{y}(\bar{x}) + (x - \bar{x})^T \; \nabla \; \hat{y}(\bar{x}), \tag{4.80}$$

where, $\nabla\hat{y}(\bar{x})$ represents the gradient of the MARS function at the point $\bar{x}$. Equation (4.80) is similar to the optimality cut in Benders' approximation method, which is added in every iteration. Algorithm 6 presents the methodology to optimize the two-stage FAM problem using a MARS recourse function.

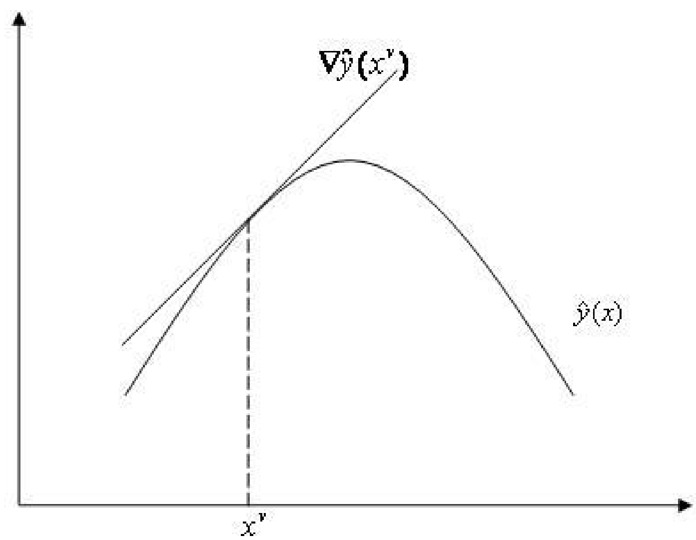

Figure 4.6. Cutting Plane Method using MARS approximation.

Algorithm 6 Cutting Plane Algorithm for MARS

Step 0: Set $s = v = 0$, $\eta^v \leftarrow \infty$ and let x^v be an initial feasible first-stage assignment. Go to Step 2.
Step 1: Set $v = v + 1$. Solve the restricted master problem represented by equations (4.77) - (4.78) & (4.80).
Let (x^v, η^v) be the solution.
Step 2: Evaluate the MARS approximation $\hat{y}(x)$ at current solution x^v.
if $\eta^v \leq \hat{y}(x^v)$ **then**
Stop. x^v is the optimal first-stage solution.
else
Generate Optimality cut (4.80) and add it to the master problem. $s = s + 1$. Go to Step 1.
end if

4.4 RESULTS

The three primary objectives for this research were to quantify:

- The importance of using D^3.
- The value of a two-stage model with respect to a single-stage model.
- How a computer experiments approach developed in this dissertation compares to Benders'.

The methodology described in Section 4.3 was implemented on the MARS approximation obtained in Chapter 4, and then compared with the L-shaped method described in Section 4.2. Initially the MARS approximation was tested for its convexity by generating 1000 random points and checking the inequality

A two-stage problem was constructed similar to the method discussed in Section 4.2.2 and solved by relaxing the second-stage LP. Scenarios for solving the FAM using Benders' were generated similar to the ones described in Section 3.4.1 (for obtaining the recourse function values for MARS approximation). The number of scenarios for the Benders' approach is debatable, as it is problem-specific and depends on the convergence of the recourse function value. For this research it was taken as 40, same as the ones required for MARS approximation, but Benders' might require more. Increasing the number of scenarios will increase the time required for obtaining the Benders' solution. In addition, some numerical issues were encountered with Benders' that caused infeasibility in the recourse problem. In order to maintain complete recourse, nearly integer numbers were rounded while generating solutions in the first stage.

Since the first-stage assignment is supposed to be binary, MIP cuts were generated from the second stage and added to the first stage in both the methods. The stopping criteria used was:

$$\text{(RMP objective - 0.025 * RMP objective)} \leq \text{Recourse function value} \tag{4.81}$$

The initial 10 - 15 MIP cuts were very effective, but later the cuts were less valuable as can be seen in Figure 4.7. In order to decrease the time required to solve the problem, we implemented two methods. The first method was to relax the first stage to allow fractional values, and LP cuts were added from the second stage. Once the criteria

$$\text{(RMP LP objective - 0.001 * RMP LP objective)} \leq \text{Recourse function value} \tag{4.82}$$

was met, MIP cuts were again added. Using this method the time required was reduced from days to hours for Benders', and was only minutes when using the MARS recourse function. The second method was to increase the node limit in each iteration linearly and once the stopping criteria for the MIP objective was reached, the problem was solved without any node limit. Another criteria that can be used to speed up the process of solving the two-stage FAM model is not solving the current master problem to optimality, but to instead terminate it as soon as a feasible solution is produced that has a value below (upper bound - ϵ); thus this incumbent is an ϵ-optimal solution [48, 49].

The results for the two methods are shown in Tables 4.1 and 4.2. The first-stage MIP assignment obtained using the two approaches was used to get the recourse function values, and they happen to be very similar (difference was within 0.1%). This can be attributed to the fact that the objective function of an SP problem is typically flat, giving rise to similar objective values, but with different solutions, as mentioned in Survajeet and Higle [50]. Table 4.1 and 4.2 shows that there is a significant reduction of time

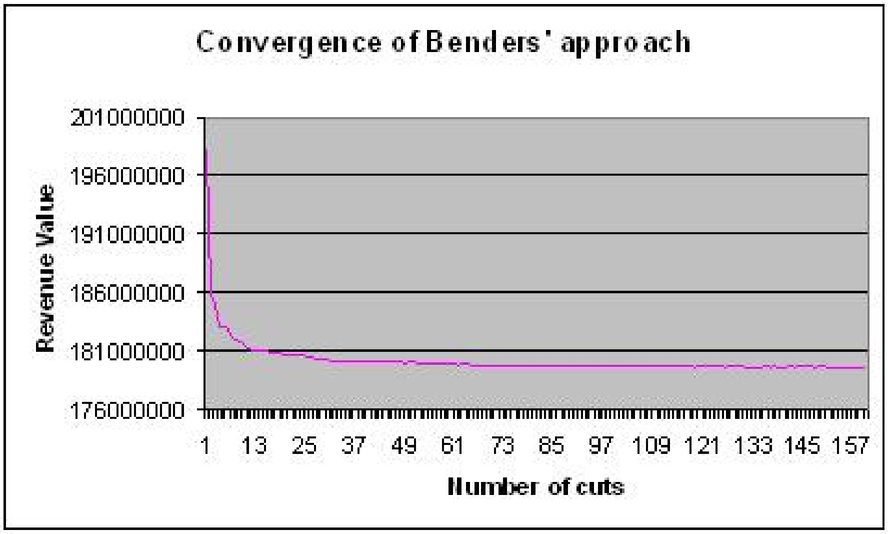

Figure 4.7. Convergence of the objective.

Table 4.1. Comparison of MARS and Benders' approaches by relaxing initially.

	MARS	Benders'	Benders'
Number of Scenarios	40	40	50
Number of LP cuts	65	512	625
Time required for LP cuts	85.6 sec	10.58 hours	15.14 hours
LP objective	178,652,847.88	178,296,016.56	178,286,461.13
Number of MIP cuts	2	0	0
Time required for MIP cuts	18.13 min	10.60 hours	15.18 hours
MIP objective	178,555,380.65	178,278,679.00	178,270,492.00
Recourse function value	177,842,438.16	177,523,946.11	177,499,444.04

in the optimization phase by using the MARS approximation to the recourse function.

Table 4.2. Comparison of MARS and Benders' approaches by increasing node limit.

	MARS	Benders'
Number of Scenarios	40	40
Number of MIP cuts	163	483
Time required for MIP cuts	3.64 hours	3.26 days
MIP objective	178,615,156.69	178,377,994.42
Recourse function value	177,613,214.82	177,459,628.86

This can be attributed to the fact that the second-stage recourse evaluation is based upon a closed-form formula as opposed to solving LP's. However, the trade-off is that the recourse function has to be generated earlier to do the optimization. Table 4.3 summarizes the different steps for obtaining the solution using a MARS recourse function. The

Table 4.3. Steps to generate first-stage solution using MARS approximation.

Step No.	Description	Time required
1	Dimensionality Reduction	1.5 hours
2	Generate LH design using MATLAB	4.5 hours
3	Map the LH design to points	0.32 hours
4	Check for feasibility	0.43 hours
5	Use the L1 norm for projection	0.97 hours
6	To find the proximate extreme point	1.1 hours
7	Generate interior points	5 min
8	Generate recourse function values	2.5 days
9	Fit a MARS approximation	0.75 hours
10	Optimize to obtain a first-stage solution	0.3 hours

time required for generating the recourse function values can be decreased by considering fewer variables and fewer design points. Some directions for future work are mentioned in Chapter 5. Using the node limit method, the total time required to solve the two-stage fleet assignment problem is around 3.07 days using our method compared to the 3.26 days for Benders'.

In order to quantify the value of D^3 and the two-stage model, a single-stage FAM plus the PMM was constructed as shown below:

$$\max\ \kappa\ = \left[-\sum_{l\in L}\sum_{f\in F} C_{fl}(x_{fl}) + \sum_{i\in I} f_i z_i\right] \tag{4.83}$$

$$\text{s.t. } a_{v^-} + \sum_{l\in A_v} x_{fl} - \sum_{l\in D_v} x_{fl} - a_{v^+} = 0 \qquad \forall v \in V, \tag{4.84}$$

$$\sum_{f\in F} x_{fl} = 1 \qquad \forall l \in L, f \in F, \tag{4.85}$$

$$\sum_{o\in O_f} a_o + \sum_{l\in L_0} x_{fl} \leq M_f \qquad \forall f \in F, \tag{4.86}$$

$$\sum_{i\ni I} z_i \leq \sum_{f\in F} Cap_f\ x_{fl} \qquad \forall l \in L, \tag{4.87}$$

$$0 \leq z_i \leq D_i \qquad \forall i \in I, \tag{4.88}$$

$$x_{fl} \in \{0,1\} \qquad \forall l \in L, f \in F \tag{4.89}$$

$$a_{v^+} \geq 0 \qquad \forall v \in V, \tag{4.90}$$

where κ is the objective value of the single-stage model. Let x^*_{ss} be the MIP solution to the single-stage model, x^*_{2sm} be the MIP solution to the two-stage model using MARS recourse function, and x^*_{2sb} be the MIP solution to the two-stage model using a Benders' approach. Let Pfam(X) and Prec(X) be the profit from FAM single-stage objective function and from the recourse function, respectively. Then the value of D^3 can be calculated as: Prec(x^*_{ss}) - Pfam(x^*_{ss}), and the value of the two-stage model using MARS recourse function is: Prec(x^*_{2sm}) - Prec(x^*_{ss}). Similarly the value of the two-stage model using Benders' approach is given as: Prec(x^*_{2sb}) - Prec(x^*_{ss}).

For our airline example, Pfam and Prec values are shown in Table 4.4.

Therefore, the value of D^3 = \$177,037,186.71 - \$164,389,021.49 = \$12,648,165.22; which is 7.7% improvement. The value of the two-stage model for MARS approach = \$177,842,438.16 - \$177,037,186.71 = \$805,251.45, which is 0.45% increase. The reason

Table 4.4. Solutions.

	Profit
Pfam(x^*_{ss})	\$164,389,021.49
Prec(x^*_{ss})	\$177,037,186.71
Prec(x^*_{2sm})	\$177,842,438.16
Prec(x^*_{2sb})	\$177,523,946.11

for such low increase in the profit in the two-stage model is because of the low standard deviation value assumed during generation of scenarios ($\sigma = 0.05\ \mu$). Such low variance is representative of good forecast in demand. A higher ratio ($\sigma = 0.25\ \mu$) would give better results.

CHAPTER 5

FUTURE WORK

This research has laid out the process and conducted a first attempt for solving the airline fleet assignment problem using a statistics-based computer experiments approach. The results of this research is promising, and this chapter presents some future directions that are expected to enhance the practicality of the approach. The method can be applied to other two-stage SP problems, since the comparison to Benders' will be application specific.

The most important research direction is further preprocessing of the decision space to reduce the number of dimensions, and consequently reduce the size of the experimental design. In the methodology presented in this research, the most time consuming part was evaluating the FAM optimization model at 3562 design points, which took nearly two and a half days. This time is actually proportional to the number of design points and the number of scenarios. Since the number of scenarios was based on confidence level information, this can only be reduced by decreasing the confidence level or increasing the error (confidence interval width). However, the number of design points required to generate the MARS approximation was based upon the number of variables remaining after removing various explicit and implicit equalities in the problem.

For the airline example employed in this research, the MARS approximation had 84 basis functions and an R^2 value $= 99.459$ (Refer Chapter 3). These 84 basis functions were based on only 42 variables out of the 1264 variables remaining after dimension reduction. Clearly, not all 1264 variables were needed for determining an excellent approximation to the recourse function. A quick study of the 3562 design points used in this research showed that a particular column had a value of zero throughout, and one column had a value of one. In these cases, we can simply assign the variable corresponding to these columns to zero and one, respectively, and eliminate these two variables.

To further reduce the dimensionality, a data mining approach called false discovery rate (FDR) can be applied to identify the potentially important variables from the remaining 1262 variables based on multiple comparisons test. This method was proposed by Benjamini and Hochberg [51] and sets a cutoff on the p-values of individual tests by controling the expected proportion of false positives in a set of non-null hypotheses. Each of these individual tests determines the significance of the variable in generating the response.

Another approach to reduce the number of dimensions is to create a simple heuristic procedure to find the high demand and high capacity legs that drive the revenue for the FAM problem. After that, a design can be constructed on the variables corresponding to these legs (which might not require as high as 1262 variables), and the methodology mentioned in this dissertation can be followed.

Other than reducing the number of points at which the second-stage problem is solved, the time required to generate the recourse function values can be decreased by utilizing the power of parallel computing. Specifically, the LP's for the different design points can all be run in parallel by partitioning the set as evenly as possible according to how many processors are available. A parallel version of MARS can also be used (refer Tsai [52]), to further reduce the time required to generate the MARS approximation. In theory, Benders' can also be parallelized, by running the second stage for different scenarios on different processors, but these are very small tasks. With regard to parallel computing, it is much more efficient to partition out large groups of tasks to the "slave" processors once (as would be the case for the design points) than to repeatedly send small tasks to the "slave" processors. This is because of the computational effort required to pass information between the "master" and "slave" processors. By partitioning the design points into two sets and using two processors, the time required to generate the recourse function values can be reduced by half and so on.

One of the assumptions made in this research is that the recourse function MARS approximation is convex. As mentioned in Chapter 4, only 18 points of the 1000 points randomly chosen to test the convexity of the recourse function proved to be non-convex. However, convexity has to be assured to obtain the global optimum for the two-stage FAM problem. In order to ensure MARS convexity, two major modifications can be made: (1) constrain basis function coefficients such that pairs of basis functions are guaranteed to jointly form convex functions; (2) alter the form of interaction terms to a new convex form. Finally, MARS convexity can be achieved by the fact that the sum of convex functions is convex.

The other area which is wide open for research is generating the design points within the polytope formed by the reduced set of cover constraints. Experimental design criteria combined with numerical optimization methods may enable a viable approach to spread the points evenly across the constrained design space. The construction of so-called "optimal space-filling" designs is a nonlinear, non-convex optimization problem, which will consequently be highly computational (see Sacks et al. [10] for example).

REFERENCES

[1] IATA, "Us$7.4 billion losses for global aviation in 2005," http://www.iata.org/pressroom/pr/2005-09-12-01.htm, 2005.

[2] R. Subramanian, R. Scheff, J. Quillinan, D. Wiper, and R. Marsten, "Coldstart: fleet assignment at delta air lines," *Interfaces*, vol. 24, no. 1, pp. 104 – 120, 1994.

[3] R. Rushmeier and S. Kontogiorgis, "Advances in the optimization of airline fleet assignment," *Transportation Science*, vol. 31, no. 2, pp. 159 – 169, 1997.

[4] J. Abara, "Applying integer linear programming to the fleet assignment problem," *Interfaces*, vol. 19, no. 4, pp. 20 – 28, 1989.

[5] IATA, "Passenger growth slows in august - fuel prices begin to dampen demand," http://www.iata.org/pressroom/pr/2005-09-30-02.htm, 2005.

[6] M. Berge and C. Hopperstad, "Demand driven dispatch: A method of dynamic aircraft capacity assignment, models and algorithms," *Operations Research*, vol. 41, no. 1, pp. 153 – 168, 1993.

[7] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming.* New York, New York: Springer, 1997.

[8] V. Chen, "Measuring the goodness of orthogonal array discretizations for stochastic programming and stochastic dynamic programming," *SIAM Journal of Optimization*, vol. 12, no. 2, pp. 322 – 344, 2001.

[9] V. Chen, K. Tsui, R. Barton, and J. Allen, "A review of design and modeling in computer experiments," *Handbook of Statistics*, vol. 22, pp. 231 – 261, 2003.

[10] J. Sacks, S. Schiller, and W. Welch, "Designs for computer experiments," *Technometrics*, vol. 31, pp. 41 – 47, 1989.

[11] H. D. Sherali, E. K. Bish, and X. Zhu, "Airline fleet assignment concepts, models, and algorithms," *European Journal of Operational Research*, vol. 172, no. 1, pp. 1 – 30, 2006.

[12] C. Hane, C. Barnhart, E. Johnson, G. Nemhauser, and G. Sigismondi, "The fleet assignment problem: Solving a large scale integer program," *Mathematical Programming*, vol. 70, pp. 211 – 232, 1995.

[13] K. T. Talluri, "Swapping applications in a daily fleet assignment," *Transportation Science*, vol. 30, no. 3, pp. 237 – 248, 1996.

[14] Z. Gu, E. Johnson, G. Nemhauser, and Y. Wang, "Some properties of the fleet assignment problem," *Operations Research Letters*, vol. 15, no. 2, pp. 59 – 71, 1994.

[15] L. Clarke, C. Hane, E. Johnson, and G. Nemhauser, "Maintenance and crew considerations in fleet assignment," *Transportation Science*, vol. 30, no. 3, pp. 249 – 260, 1996.

[16] C. Barnhart, N. Boland, L. Clarke, E. Johnson, G. Nemhauser, and R. Shenoi, "Flight string models for aircraft fleeting and routing," *Transportation Science*, vol. 32, no. 3, pp. 208 – 220, 1998.

[17] J. Rosenberger, E. Johnson, and G. Nemhauser, "A robust fleet assignment model with hub isolation and short cycles," *Transportation Science*, vol. 38, no. 3, pp. 357 – 368, 2003.

[18] B. Rexing, C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamurthy, "Airline fleet assignment with time windows," *Transportation Science*, vol. 34, no. 1, pp. 1 – 20, 2000.

[19] R. Ahuja and J. Orlin, "Very large scale neighborhood search in airline fleet scheduling," *SIAM News*, vol. 35, no. 9, 2002.

[20] L. Lettovsky, E. Johnson, and B. Smith, "Schedule generation model," in *AGIFORS Symposium Proceedings, 39*, New Orleans, LA, USA, 1999.

[21] A. Farkas, "The influence of network effects and yield management on airline assignment decisions," Ph.D. dissertation, Massachusetts Institute of Technology, 1996.

[22] T. Kniker, "Itinerary-based airline fleet assignment," Ph.D. dissertation, Massachusetts Institute of Technology, 1998.

[23] C. Barnhart, T. Kniker, and M. Lohatepanont, "Itinerary-based airline fleet assignment," *Transportation Science*, vol. 36, no. 2, pp. 199 – 217, 2002.

[24] T. Jacobs, E. Johnson, and B. Smith, "O-d fam: Incorporating passenger flows into the fleeting process," in *AGIFORS Symposium Proceedings, 39*, New Orleans, LA, USA, 1999.

[25] C. Barry, "Robust airline fleet assignment," Ph.D. dissertation, Georgia Institute of Technology, 2004.

[26] O. Listes and R. Dekker, "A scenario aggregation based approach for determining a robust airline fleet composition," Erasmus University Rotterdam, The Netherlands, Tech. Rep. Econometric Institute Report EI 2002-17, 2002.

[27] R. Rockafellar and R. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of Operations Research*, vol. 16, no. 1, pp. 119 – 147, 1991.

[28] H. D. Sherali and X. Zhu, "Two-stage fleet assignment model considering stochastic passenger demands," *Working Paper*, 2006.

[29] J. Tsai and V. Chen, "Flexible and robust implementations of multivariate adaptive regression splines within a wastewater treatment stochastic dynamic program," *Quality and Reliability Engineering International*, vol. 21, no. 7, pp. 689 – 699, 2005.

[30] P. Werbos, *A Brain-Like Design To Learn Optimal Decision Strategies in Complex Environments.* Springer, London: Springer, 1998.

[31] C. Cervellera, V. Chen, and A. Wen, "Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization," *European Journal of Operational Research, to appear*, 2005.

[32] V. Pilla, V. Chen, J. Rosenberger, and B. Boardman, "Dimensionality reduction in fleet assignment modeling," in *IERC Annual Conference Proceedings*, Houston, TX, USA, 2004.

[33] M. Savelsbergh, "Preprocessing and probing techniques for mixed integer programming problems," *ORSA Journal on Computing*, vol. 6, no. 4, pp. 445 – 454, 1994.

[34] R. Myers and D. Montgomery, *Response Surface Methodology.* New York, New York: John Wiley and Sons Inc, 1986.

[35] L. Gao and Y. Zhang, "Computational experience with lenstra's algorithm," Rice University, Department of Computational and Applied Mathematics, Tech. Rep. TR02-12, 2002.

[36] G. Sierksma and G. Tijssen, "Degeneracy degrees of constraint collections," *Mathematical Methods of Operations Research*, vol. 57, pp. 437 – 448, 2003.

[37] V. Pilla, J. Rosenberger, and V. Chen, "An lp formulation for implicit dimensionality reduction," University of Texas at Arlington, Department of Industrial Engineering, Tech. Rep. COSMOS-05-01, 2005.

[38] M. Mckay, R. Beckman, and W. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239 – 245, 1979.

[39] C. Maranas, C. Floudas, and P. Pardalos, "New results in the packing of equal circles in a square," *DMATH: Discrete Mathematics*, vol. 142, 1995.

[40] V. Chvatal, *Linear Programming.* New York, New York: W. H. Freeman and Company, 1983.

[41] H. Minkowski, *Geometrie der Zahlen.* Teubner Leipzig, 1896.

[42] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, pp. 1 – 141, 1991.

[43] D. G. Luenberger, *Linear and Nonlinear Programming.* Reading, Massachusetts: Addison-Wesley, 1975.

[44] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, New York: Springer, 2003.

[45] J. E. Kelley, "The cutting-plane method for solving convex programs," *SIAM journal on applied mathematics*, vol. 8, pp. 703 – 712, 1960.

[46] Birge, "computational methods for large-scale stochastic dynamic programs," http://ice.uchicago.edu/slides/Birge.pdf, 2005.

[47] V. Chen, D. Ruppert, and C. Shoemaker, "Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming," *Operations Research*, vol. 47, pp. 38 – 53, 1999.

[48] A. M. Geoffrion and G. Graves, "Multicommodity distribution system design by benders decomposition," *Management Science*, vol. 20, no. 5, pp. 822 – 844, 1974.

[49] W. P. Adams and H. Sherali, "Mixed-integer bilinear programming problems," *Mathematical Programming*, vol. 59, no. 3, pp. 279 – 305, 1993.

[50] S. Survajeet and J. Higle, *Stochastic Decomposition.* Boston, MA: Kluwer Academic Publishers, 1996.

[51] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the royal statistical society, series B*, vol. 57, no. 1, pp. 289 – 300, 1995.

[52] J. Tsai, "Statistical modeling of the value function in high-dimensional, continuous-state stochastic dynamic programming," Ph.D. dissertation, Georgia Institute of Technology, 2002.

BIOGRAPHICAL STATEMENT

Venkata L. Pilla was born in Visakhapatnam, India, in 1978. He received his B.S. degree in Mechanical, Production and Industrial Engineering from Gandhi Institute of Technology and Management (Andhra University), India, in 1999. He recieved his M.S. and Ph.D. degrees from The University of Texas at Arlington in 2001 and 2006, respectively, both in Industrial Engineering. His research interests include applied probability and statistics, and optimization.