5-2018

Computational optimization of networks of dynamical systems under uncertainties: application to the air transportation system

Jun Chen
Purdue University

COMPUTATIONAL OPTIMIZATION OF NETWORKS OF DYNAMICAL SYSTEMS UNDER UNCERTAINTIES: APPLICATION TO THE AIR TRANSPORTATION SYSTEM

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Jun Chen

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Dengfeng Sun, Chair

School of Aeronautics and Astronautics

Dr. Daniel DeLaurentis

School of Aeronautics and Astronautics

Dr. Inseok Hwang

School of Aeronautics and Astronautics

Dr. Jianghai Hu

School of Electrical and Computer Engineering

Approved by:

Dr. Weinong Chen

Head of the School Graduate Program

To my family

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude and appreciation to my advisor, Professor Dengfeng Sun, for helping me define research problems, for honest and helpful feedback, for supporting and encouraging my research path, and all the proof-reading and comments of research write-ups. He is an inspiring mentor and a caring friend. I think I learned more than just academic knowledge through working with him. I am grateful for every discussion I have had with him, which extended to not only interesting engineering problems but also unscientific daily things. His enthusiasm, support and encouragement will continue to motivate me to go forward in the future.

I am also indebted to my committee members, Professor Daniel DeLaurentis, Professor Jianghai Hu and Professor Inseok Hwang for their support, invaluable insight and constructive comments on my dissertation. Their expertise in the air transportation system, control theory and multi-agent systems are extremely valuable sources that have enriched my knowledge and helped me think problems from different respects. I am lucky to TA for Professor Martin Corless, and am thankful for his advice in academic life and research, also for his sense of humor.

I would also like to thank Professor Lijian Chen (University of Dayton) for the discussion on convex approximation, which inspired the work in Chapter 3. My gratitude also goes to some former members in Professor Sun's group, Dr. Peng Wei (Iowa State University), Dr. Yi Cao (Apple) and Dr. Christabelle Bosson (NASA Ames). I learned a lot of valuable knowledge of air transportation system and academic career from several conversations with Dr. Wei. Dr. Cao guided me to the big data platform and helped me a lot for building the Spark cluster. Dr. Christabelle Bosson is very helpful in finding proper data sources and giving valuable suggestions.

It is a enjoyable journey to work with the talent people at Purdue. I am thankful to my colleges Teng-yao Yang, Jiazhen Zhou and Meng Li for their supports and constructive suggestions during these years. Special thanks go to my friend Chiyu Zhang for exchanging research ideas and providing supports during daily life. I also would like to thank my friends during my Purdue years, providing support and entertainment: Waterloo Tsutsui, Michael Fruhnert, Ning Liu, Zhenbo Wang, Mingyu Cai, Yanran Cui, Hao wang, Yufei Long, Liang Zhang and Banghua Zhao.

Finally, my deepest gratitude goes to my family. I cannot finish my study in Purdue without their encouragement and endless love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABBREVIATIONS

AAR	Airport Acceptance Rate
ARTCSCC	Air Traffic Control System Command Center
ARTCC	Air Route Traffic Control Center
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
ATC	Air Traffic Control
ASDI	Aircraft Situation Distributed to Industry
BTS	Bureau of Transportation Statistics
B&B	Branch and Bound
CDM	Collaborative Decision Making
CCM	Chance Constrained Model
CDF	Cumulative Distribution Function
DST	Decision Support Tool
ETMS	Enhanced Traffic Management System
EWR	Newark Liberty International Airport
FAA	Federal Aviation Administration
GDP	Ground Delay Program
GHP	Ground Holding Problem
HDFS	Hadoop Distributed File System
IP	Integer Programming
JFK	John F. Kennedy International Airport
KDE	Kernel Density Estimation
LGA	LaGuardia Airport
LTM	Link Transmission Model
MIT	Miles-in-Trail

MAGHP	Multiple Airport Ground Holding Problem
MILP	Mixed Integer Linear Programming
NextGen	Next Generation Air Transportation System
NAS	National Airspace System
NYC	New York City
PAARs	Planned Airport Acceptance Rates
PDF	Probability Density Function
RDD	Resilient Distributed Dataset
SAGHP	Single Airport Ground Holding Problem
SFO	San Francisco International Airport
SAA	Sample Average Approximation
TMI	Traffic Management Initiatives
TFM	Traffic Flow Management
ZAU	Chicago Air Route Traffic Control Center
ZID	Indianapolis Air Route Traffic Control Center

ABSTRACT

Chen, Jun Ph.D., Purdue University, May 2018. Computational Optimization of Networks of Dynamical Systems under Uncertainties: Application to the Air Transportation System. Major Professor: Dengfeng Sun Professor.

To efficiently balance traffic demand and capacity, optimization of air traffic management relies on accurate predictions of future capacities, which are inherently uncertain due to weather forecast. This dissertation presents a novel computational efficient approach to address the uncertainties in air traffic system by using chance constrained optimization model.

First, a chance constrained model for a single airport ground holding problem is proposed with the concept of service level, which provides a event-oriented performance criterion for uncertainty. With the validated advantage on robust optimal planning under uncertainty, the chance constrained model is developed for joint planning for multiple related airports. The probabilistic capacity constraints of airspace resources provide a quantized way to balance the solution's robustness and potential cost, which is well validated against the classic stochastic scenario tree-based method.

Following the similar idea, the chance constrained model is extended to formulate a traffic flow management problem under probabilistic sector capacities, which is derived from a previous deterministic linear model. The nonlinearity from the chance constraint makes this problem difficult to solve, especially for a large scale case. To address the computational efficiency problem, a novel convex approximation based approach is proposed based on the numerical properties of the Bernstein polynomial. By effectively controlling the approximation error for both the function value and gradient, a first-order algorithm can be adopted to obtain a satisfactory solution which is expected to be optimal. The convex approximation approach is evaluated to be reliable by comparing with a brute-force method.

Finally, the specially designed architecture of the convex approximation provides massive independent internal approximation processes, which makes parallel computing to be suitable. A distributed computing framework is designed based on Spark, a big data cluster computing system, to further improve the computational efficiency. By taking the advantage of Spark, the distributed framework enables concurrent executions for the convex approximation processes. Evolved from a basic cloud computing package, Hadoop MapReduce, Spark provides advanced features on in-memory computing and dynamical task allocation. Performed on a small cluster of six workstations, these features are well demonstrated by comparing with MapReduce in solving the chance constrained model.

1. INTRODUCTION

1.1 Motivation

As a key component for national well-being, air transportation system's safety and efficiency have been prominent. Driven by the growing United States (U.S.) and world economy, the demand for aviation is growing over the long run. The latest forecast from the Federal Aviation Administration (FAA) calls for U.S. carrier passenger growth over the next 20 years to average 1.9 percent per year [1]. In recent years, the air transportation system has been facing critical safety issues all the time, while more and more passengers are experiencing ever-increasing flight delays and cancellations. According to the Bureau of Transportation Statistics (BTS), nearly one in five airline flights arrived at its destination over 15 minutes late in 2017[1], and the average annual total cost of air transportation delays was over $30 billion [2]. Moreover, the expanding traffic demand on the current air transportation network will also increase the workload of the air traffic controller, which might threaten the safety of operations. All of the above facts pose a significant challenge to the development of Next Generation Air Transportation System (NextGen). The importance of a safe, efficient, robust, and (partially) automated Air Traffic Management (ATM) system is paramount.

The goal of Air Traffic Flow Management (ATFM) is to allocate airspace resources such that the balance between capacity and demand is maintained, subject to both en-route and airport capacity constraints. Airport and airspace sector capacities are greatly influenced by weather conditions such as fog, snow, wind and reduced visibility. These severe weather conditions may reduce both airspace and airport capacity

[1]Data retrieved from the Bureau of Transportation Statistics, URL: https://transtats.bts.gov/HomeDrillChart.asp

such that the demand and supply situation of ATFM is made worse and eventually may result in delays and cancellations. According to the BTS, severe weather has been identified as the most important causal factor for traffic delays in the United States [3]. As it is shown in Figure 1.1 and 1.2, weather accounts for more than half flight delays in most of the months in 2017. Moreover, the incomplete knowledge of the weather forecasting brings uncertainty into capacities, which also poses a significant challenge to ATFM [4]. Strategic traffic flow management decisions made under uncertainty can cause nationwide severe congestion in the National Airspace System (NAS). This fact motivates the need for stochastic optimization algorithms for ATFM that account for capacity uncertainty.

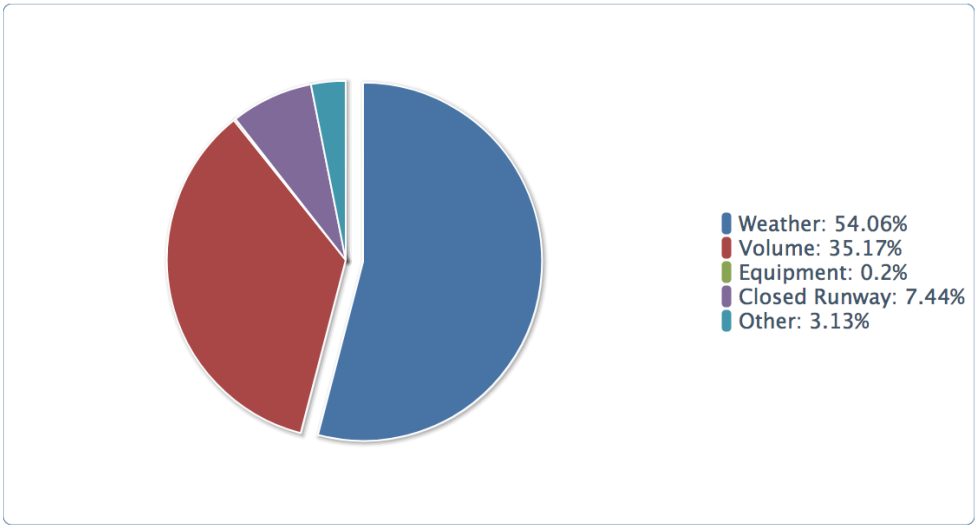


Figure 1.1. Causes of national aviation system delays in 2017

The NAS is a highly connected network, which includes a large number of shared resources, such as aircraft, crew, passengers and gate space. The connective resources further complicates the ATFM problem. For example, airlines usually fly one aircraft on daily scheduled itineraries that require visits to a sequence of airports. In this case, the late arriving aircraft delay early in the day has a significant impact on the downstream delay performance [5,6]. As a result of the high connectivity in NAS, it is desired to have a scalable approach to solve large-scale problems with long planning

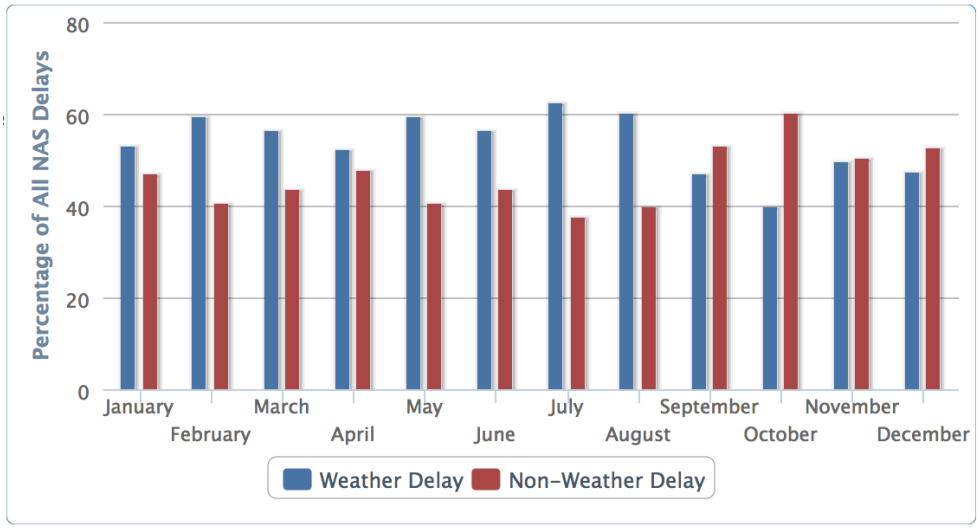


Figure 1.2. Weather's share of delayed flights in 2017

horizon rather than only considering local regional problem within a short period. However, managing large-scale flight operations is a challenging task that needs the help of computer-based decision support tools (DSTs). Due to dynamic natures of air traffic, efficient solutions are critical to the applicability of DSTs. The interval of radar-based position update in the enhanced traffic management system (ETMS) for en route traffic is roughly one minute. Ideally, a DST should deliver a solution within this timeframe. As a result, computational efficiency becomes a concern in ATFM.

In the current state of fundamental research, we are facing two major challenges in mitigating the disruptions in the air transportation system caused by uncertainties in stochastic airspace capacity, among others.

1. Most existing methods neglect uncertainties and formulate the ATFM as a deterministic optimization problem; the uncertainties are often handled subjectively using past human experience through verbal communications among different traffic controllers in the system. With digital communications and a more accurate weather information system in NextGen, an urgent need exists to develop innovative stochastic ATFM optimization framework.

2. Most stochastic ATFM studies are based on a probabilistic scenario-tree approach, which is subject to the curse dimensionality and extremely difficult, if not impossible, to be applicable to real-world ATFM problems.

This work is motivated by these challenges, and aims to achieve the following objective: Develop and evaluate an efficient decision support algorithm for large-scale air traffic management in the presence of uncertainty.

1.2 Air Traffic Systems

This section will provide an introduction of the air traffic systems in U.S., which includes the structure of the NAS and some commonly used concepts in ATFM.

The NAS is a large-scale connected network, which consists of airports and airspace. The airspace has a hierarchical structure, which contains a single Air Traffic Control System Command Center (ATCSCC, or simply, Command Center) and 22 Air Route Traffic Control Centers (ARTCCs, or simply, Center) [7]. Each center is further divided into multiple sectors, the smallest control unit in NAS. Each sector is monitored by one or more air traffic controllers to ensure flight safety. Due to safety issues, there is a capacity associated with each sector, which is the maximum number of aircraft that the controllers can handle at the same time. The structure of high level sector in NAS is shown in Figure 1.3.

The control and coordination of aircraft in the NAS is provided by Air Traffic Control (ATC) and AFTM. To be specific, the primary duty of ATC is to ensure safe separations between aircraft in the system, and ATFM is responsible for balancing air traffic demand with the system capacity. Usually, ATFM is performed at the ARTCC level with 10-20 sectors. Both the demand and capacity are time-varying, since the demand is driven by the time-varying traffic need and the capacity is affected by many time-varying factors (e.g, weather, controller and runway configuration). Once the predicted demand exceeds capacity during some time periods (typically, 15 minutes), the control tools of ATFM, called Traffic Management Initiatives (TMIs)

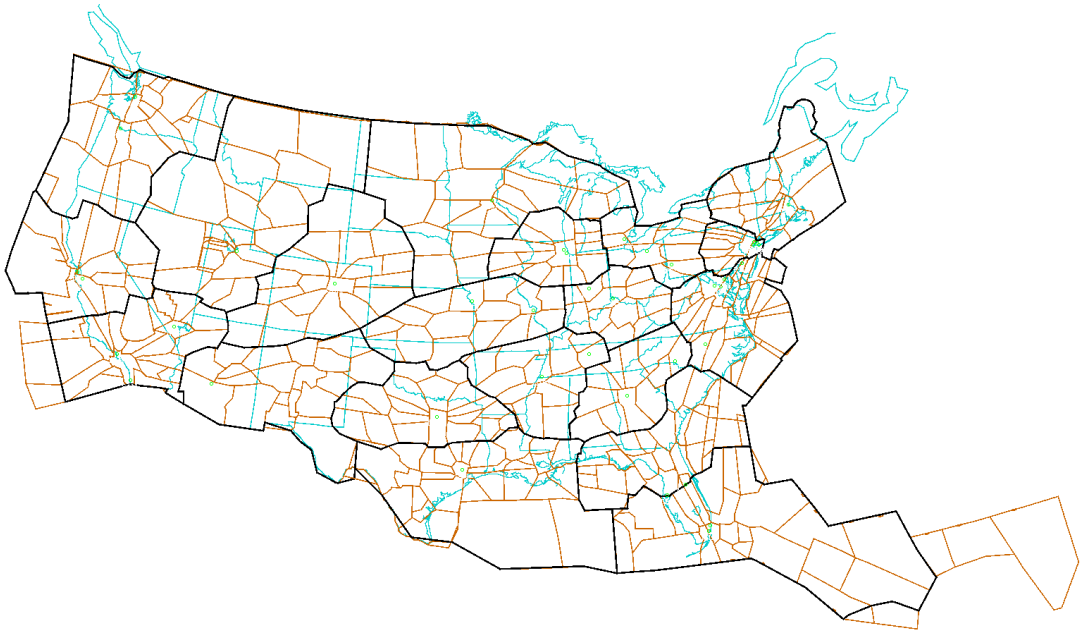


Figure 1.3. An illustration of high level sector structure in NAS

will be performed to mitigate potential congestions. The commonly used TMIs and associated concepts are introduced as follows:

- Ground Delay Program (GDP): A GDP is a procedure that flights are assigned a later time slot of arrival via ground delay at departure airport to avoid airborne delay, because it is cheaper and safer to delay flights on the ground than to hold them when they are airborne. A GDP is often issued to control air traffic volume to airports where the projected traffic demand is expected to exceed the airport's airport acceptance rate (AAR); the AAR describes the number of arrivals an airport is capable of accepting for a length/period of time (usually 15 min or more). It is normally a result of the AAR being reduced for some reason: most often, adverse weather.

- Miles-in-Trail (MIT) Restriction: MIT describes the minimum allowable miles between successive aircraft departing/arriving at an airport, over a fix, through

a sector, or on a specific route. MIT is used to apportion traffic into a manageable flow, as well as to provide space for additional traffic (merging or departing) to enter the flow of traffic. For example, standard separation between aircraft in the en route environment is 5 n miles [4]. During a weather event, this separation may increase significantly.

- Collaborative Decision Making (CDM): CDM is a joint initiative of the FAA and airlines, which established a paradigm that allows the airline to swap flights within the arrival slot alloted to it. CDM provides airlines with flexibility to improve their internal costs by intra-airline slot substitution based on their own business interest.

- Collaborative Routing: it is a similar program with CDM, which tries to apply the concept of CDM to en route traffic. The primary purpose is to mitigate en route congestions in weather affected regions by providing alternative routes.

Currently, most of the decisions for TMIs are made by controllers based human experience; computer aided DSTs will release the controller's workload in the future with NextGen.

1.3 Literature Review

In the past three decades, the ATFM problem in air transportation has been studied by many researchers in order to address air traffic congestion. The AFTM optimization research has two major categories: Ground Holding Problem (GHP) and Traffic Flow Management (TFM). The GHP only considers airport arrival/departure rate to mitigate airport congestions, but the TFM accounts for both the en route sector and airport constraints. This section presents an investigation of prior work related to the modeling of GHP and TFM problems, both deterministic and stochastic, and associated efforts in improving the computational efficiency.

The primary purposes of studying GHP is to support the GDP action at airports, which is one of the most effective TMIs to alleviate airport congestion. The objective of this problem is to minimize the sum of airborne and ground delay costs. Most GHPs focus on modeling a response to a reduced AAR [5]. The first effort dates back to 1987, when Odoni was among the first to propose the mathematical formulation of GHP [8]. Terrab and Odoni presented a deterministic model for the single-airport ground-holding problem (SAGHP) problem, which minimizes the total ground holding cost [9]. Later, Hoffman and Ball proposed a deterministic model of the SAGHP with banking constraints, which added the constraint that flights must arrive within pre-specified time windows. Such a condition is useful to model hubspoke operations at major airports [10]. To mitigate the exemption bias in GDP, a fairness allocation concept is introduced to model equity [11].

The first multiple airport ground holding problem (MAGHP) was introduced by Vranas et al. [12]. This MAGHP model was formulated as a deterministic integer program to assign optimal ground delays in a network of airports. However, the computational burden is too expensive that prevents the practice of this model in reality. Later, Bertsimas and Stock Patterson proposed a binary integer programming formulation that considered both airspace and airport capacities, known as the BSP model [13]. The description of the state of aircraft is based on the trajectory of individual aircraft; therefore, BSP is a Lagrangian model. A limitation of Lagrangian models is that the dimension of the model is related to the number of aircraft involved in the planning time horizon. The BSP model is proved to be non-deterministic polynomial-time (NP) hard by deriving the equivalent job-shop scheduling problem [13]. The key contribution of BSP was the development of strong formulation of ATFM, where many constraints were demonstrated to be facet-defining, which results in good computational efficiency. Subsequently, Bertsimas presented several extensions of the BSP model to account for other features, such as rerouting [14–16].

To overcome the computational limitation of the Lagrangian models, the Eulerian model of ATFM was proposed [17], which is inspired by the Daganzo Cell

Transmission Model [18, 19]. Since the Eulerian approach spatially aggregates the air traffic, its computational complexity does not depend on the number of aircraft, but only on the size of the network problem [20]. Afterwards, an aggregate Eulerian-Lagrangian model was proposed to eliminate the splitting and diffusion problems of some Eulerian models by taking into account the origin-destination information of flights [21]. Following this, a link transmission model was developed based on the Eulerian-Lagrangian model to further improve the computational efficiency [22]. Moreover, the distributed algorithms for these aggregate models have also been proposed using the dual decomposition method [23, 24].

In parallel, computational difficulty can also be overcome by hardware [25]. With a more powerful computer, the BSP benchmark increased to involve 3,000 flights while the running time was reduced to around 16 minutes [14]. Following this, a multi-threaded programming approach was employed to achieve further speedup [26]. The implementation enforces the CPU to run at full scale thereby increasing efficiency. But the parallelism was limited to a standalone computer. More recently, an Eulerian-Lagrangian model was solved by massive parallel computing [21,22]. The running time decreased from 2 hours to 6 minutes by splitting the computations on a cluster of 10 Dell workstations [27]. The design made full use of distributed computation resources to increase efficiency. However, it requires extensive programming skills to implement multi-threaded programming on a cluster, such as dealing with communication and synchronization issues. To overcome this limitation, a Cloud computing framework with Apache Hadoop MapReduce was implemented to reduce the development workload from multi-threaded programming [28], where Hadoop MapReduce is a software framework to process large-scale data in parallel on large cluster. With its built-in fault-tolerance capability, the MapReduce framework could not only be efficient but also robust. However, MapReduce is not well suited for iterative optimization since in each iteration the data has to be read from HDFS (Hadoop Distributed File System) and there is a significant cost of starting and finishing a MapReduce job. To further improve the efficiency, this dissertation will extend the Hadoop-based air traffic flow

management from MapReduce framework to Spark, a cluster computing platform suitable for large-scale data processing [29]. Further speedup could be achieved by Spark's ability to run computation in memory. Moreover, the unbalanced workload limitation on MapReduce could be solved by Spark's dynamic schedule allocation feature and the Spark framework abstracts away MapReduce implementation details to help reduce the difficulty of programming.

Since weather conditions are difficult to predict and have a significant impact on capacities, considerable efforts have been made to address the capacity uncertainty. Due to the computational complexity of solving large-scale ATFM problems, most of the stochastic ATFM models are limited to optimizing flows into a single airport (SAGHP). As one of the first attempts, Richetta and Odoni formulated a stochastic integer programming model for the SAGHP [30].

Later, FAA implemented a new GDP paradigm, known as collaborative decision making, in which the airlines have more autonomy about their schedules. Under CDM paradigm, the arrival slots are first allocated to individual flights based on the planned airport acceptance rates (PAARs) [31]. Then, the airlines are allowed to exchange the arrival slots among themselves, which is the key feature of CDM [32]. Many models were proposed to assist the implementation of GDP under CDM. Ball et al. formulated an aggregative static stochastic model with dual network structure, which solves for an optimal PAARs during different time intervals [33]. However, once the ground-holding strategies were decided "once and for all" at the beginning of planning time horizon, they could not be revised even for flights that have yet to depart. Mukherjee and Hansen improved this dynamic model by allowing for ground-holding revisions contingent on updated scenario realizations [34]. Recently, Mukherjee and Hansen proposed a model that incorporated dynamic rerouting into SAGHP [35].

In all of the aforementioned models, the uncertainty in capacities was represented through a finite number of scenarios arranged in a probabilistic decision tree. As time progressed, the branches of the tree were realized, resulting in better information

about future capacities [36]. Moreover, the techniques were developed to determine probabilistic capacity profiles and scenario tree forecasts from historical data [36]. Unfortunately, the probabilistic scenario-tree approach suffers significantly from the practical difficulty of not knowing the exact distribution of the data to generate relevant scenarios. Furthermore, it generally becomes intractable quickly as the number of scenarios increases, thereby posing substantial computational challenges.

Besides the scenario tree method, robust optimization can also address decision-making under uncertainty. The robust optimization formulations of the ATFM problem was studied in [37] to address capacity uncertainties. However, the robust optimization may suffer from highly conservative solutions, since it is a consequence of the optimization over the worst-case realization of the uncertainty parameters. Consequently, there is an alternative method to incorporate probabilistic information called *Chance Constraints*. The idea is to constrain the chance of a constraint violation, given probabilistic information about future state disturbances. This is less conservative than the robust approach of constraining against the constraint violation for all possible disturbances. Currently, only one article has discussed the ATFM problem with Chance constraints [38], which is formed as a Mixed-Integer Linear Programming (MILP) model based on the BSP model. However, this MILP model uses the brute-force method to enumerate all possible capacity combinations. Thus the exponentially increased computational complexity prevents it from being applicable to large-scale problems in reality.

1.4 Contributions

This contributions of this work are summarized in the follows:

- **A stochastic optimization approach to address uncertainty.** This dissertation presents a Chance Constrained Model (CCM) to handle uncertainty in air traffic systems. The chance constrained model introduced the probabilistic capacity constraints to the previous deterministic models. This is a fundamen-

tally different stochastic approach than the traditional scenario tree method. The benefit of chance constrained model can help overcome the computational limitation for large-scale problem under uncertainty.

- **A convex approximation to solve the CCM.** To make the CCM solvable for large-scale problem, this dissertation develops a convex approximation-based approach. The approximation is based on the numerical properties of the Bernstein polynomial, which is capable of effectively controlling the approximation error for both the function value and the gradient. Moreover, the approximation approach is specially designed to have the ability to be solved in parallel in a distributed manner, which is the fundamental for distributed computing.

- **Distributed computing framework for stochastic ATFM** To track the large-scale stochastic problem, a distributed computing framework is introduced to overcome the computation burden. This dissertation designs and employs a Spark-based distributed computing framework to carry out the computation for solving the large-scale CCM for ATFM. The prototype of Spark-based distributed computing framework can be easily adapted to solve other large-scale dynamical systems.

- **Service level evaluation** A new metric associated with the CCM is proposed, called service level. The service level represents the reliable/risk level of the system. Low service level will produce result with high risks that could lead to failure in high chance. The service level is evaluated for the previous stochastic approach based on scenario tree and the CCM. The CCM is demonstrated to guarantee the required service level.

- **Model validation** The proposed CCM model is validated through a test with real traffic data. In addition to the convex approximation approach, a brute force method is also presented for en route air traffic flow. The brute force method is a modified version of the MILP model [38], which can generate real

optimal solution for small sized problem. These two methods are compared using the same flight plan to validate the performance of the convex approximation approach.

- **Implementation of the CCM with GDP** The CCM is implemented for the multi-airport GDP in a metroplex, which is compared with the result from the traditional scenario tree method. Two features are demonstrated with CCM: (i) the service level is guaranteed with CCM, (ii) the CCM model provides a flexibility to generate robust solution under adjustable service level.

- **Implementation of the CCM with ATFM** The CCM is also implemented with the Spark-based distributed computing framework to solve sector level Traffic Flow Management problems, which demonstrates the efficiency and tractability for solving large-scale stochastic problem.

This research focuses on air traffic management. However, this chance-constrained optimization method and its computation platform are potentially helpful in their application to several other domains in air transportation, such as airport surface operations and airline management under uncertainties.

1.5 Organization of This Dissertation

The rest of this dissertation is organized as follows. In chapter 2, the chance constrained model is applied to the multiple airports ground delay program. The service level is introduced and evaluated first; then a comparison test between chance constrained model and scenario tree method is performed to demonstrate the robustness of results and the adjustable service level. Chapter 3 introduces the chance-constrained model with a convex polynomial approximation-based approach to solve it. Then the main algorithm based on the polynomial approximation-based approach is presented with computational complexity. In Chapter 4, the chance constrained model is proposed to account for uncertainty in future capacities of en route airspace.

The performance of the approximation-based approach is validated by comparing with the brute force method. Chapter 5 demonstrates the parallel computing framework for the approximation-based approach. Conclusion remarks and future directions are summarized in Chapter 6.

2. GROUND DELAY PROGRAM WITH CHANCE CONSTRAINT

2.1 Introduction

The ground delay program is one of the most effective strategic TMIs used to alleviate congestion costs and it ensures safe and efficient air traffic [39]. In a GDP, flights are held on the ground at their origin airports when there is an expected reduction of landing capacity at the destination airport. The landing capacity is also referred as Airport Acceptance Rate, which describes the number of arrivals an airport is capable of accepting per hour. The assigned ground delay helps absorb airborne delay such that the traffic supply-demand balance is maintained with cheaper and safer delay cost.

With rapid growth of air traffic, the airports in a metropolitan area can not be considered as separated entities, but rather as interdependent system, known as a *metroplex* [40]. A metroplex phenomenon is an interaction between two or more airports in close geographically proximity [41]. Adverse weather usually affects multiple airports in a metroplex simultaneously, such that the joint AARs of a metroplex is reduced, since adverse weather such as fog, snow, wind, and reduced visibility may require greater spacing between flights [4, 5]. The imperfect weather forecast brings uncertainty into the GDP planning. Decisions made under uncertainty can cause airborne delays for multiple airports simultaneously, which greatly lower the efficiency in those busy metroplex airspace. This highlights the importance of addressing weather uncertainty in the GDP planing in a metroplex to mitigate congestions.

This chapter proposes an alternative method to incorporate probabilistic information for GDP planing with chance constraints. The idea is to constrain the chance of a constraint violation, given probabilistic information about future state distur-

bances. The major advantage of chance constrained model is the ability to provide robust solutions with user-defined service level, where the service level represents the chance of the constraints not being violated. The service level can be defined by the air traffic authority or airlines. First, the concept of service level is introduced for the air traffic management. Then a chance constrained model is developed based on the Ball et al. model to provide a robust optimal PAARs with required service level for SAGHP [33]. The service level is evaluated with the same flight plan for three different methods, which are deterministic model, scenario based stochastic model (Ball et al.) and chance constrained model. In the end, to further demonstrate the advantages of the chance constrained model for multi-airport systems GDP planning, both the Ball et al. model and the chance constrained model are applied to a metroplex ground delay problem (MAGDP). The evaluation used real flight schedules from the NYC metroplex airports: John F. Kennedy International (JKF), Newark Liberty International (EWR) and LaGuardia (LGA) Airports.

2.2 Service Level

2.2.1 Definition

Service level is a concept that is often used in supply chain management and in inventory management to measure the performance of a system. With the certain goals are defined, the service level gives the percentage to which those goals should be achieved. Several definition of service levels are mentioned in literature, the most commonly used service level that is highly related to ATM, is the α service level, which is also known as type 1 service level [42].

The α service level is an event-oriented performance criterion. It measures the probability that all the customer demand within a given time interval will be satisfied without delay. The mathematical definition of α service level is shown as follows:

$$\alpha = Prob\{Period\ Demand \leq Inventory\ on\ hand\} \tag{2.1}$$

where α denotes the probability that an arbitrarily arriving demand will be completely served from stock on hand. In order to determine the safety stock that guarantees a target α service level, the stationary probability distribution of the inventory on hand must be known.

In ATM, the demand is traffic demand and the inventory at hand is actually the airspace capacity of airport or sector. Therefore, by introducing the α service level into ATM, we can define a similar performance criterion as follows:

$$\alpha = Prob\{Period\ Traffic\ Demand \leq Period\ Capacity\} \tag{2.2}$$

where α represents the probability that the traffic demand does not exceed the airspace capacity at each time period.

2.2.2 Evaluation Method

The service level can be easily evaluated by using Monte Carlo simulation method. The Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results [43]. Once the capacity distribution is known, the Monte Carlo method can be easily designed to evaluate the service level conveniently.

Given the capacity distribution and scheduled flight plan, the Monte Carlo based evaluation method for the service level is shown as follows:

1. Calculate the traffic demand for the target sector/airport j, as D_j.
2. Randomly sample N independent points based on the given capacity distribution, where N is a big integer number (e.g. 10000).
3. For each sample point C_i, if $C_i \geq D_j$, then $I_i = 1$; otherwise, $I_j = 0$.
4. The service level for sector/airport j: $\alpha_j = \frac{\sum_{i=1}^{N} I_j}{N}$

The similar evaluation process can be done for the service level of multiple sectors or airports system. The only difference is that the random sample is drawn from the joint capacity distribution.

2.3 Single Airport Ground Holding Problem with Chance Constraints

The current operations of ATM heavily focus on deterministic algorithms. The most commonly used stochastic model is developed based on the scenario tree method, such as the Ball et al. model [33]. The scenario tree method only considers the expected optimal objective with the probabilistic scenario tree. However, the expected optimal oriented metric cannot guarantee the robustness of the solution. The extreme cases with half very good results and half very bad results will also provide a good expected objective. Therefore, there is no limit on the percentage of good results, which is the advanced feature that chance constraints have.

To demonstrate the evaluation of the current system's service level, the simple and classic SAGHP is applied in this section. First, the deterministic model and static scenario tree model for the SAGHP are reviewed based on the Ball et al. model [33]. Then the proposed chance constrained model is derived based on the previous deterministic model. The same problem is solved by all the three models to get the corresponding optimal solutions. The service level is evaluated individually for a deterministic case, a scenario tree case and a chance constrained case for comparison.

2.3.1 Deterministic Model

The deterministic model for SAGHP is a simplified version of the Ball et al. model by setting the capacity to be deterministic. The objective of SAGHP is to minimize the total cost of ground delay and airborne delay, where airborne delay is often more expensive due to the safety issues. The deterministic formulation is

$$\textbf{min} \sum_{t=1}^{T} (c_g Y_t + c_a Z_t)$$

s.t.

$$\begin{aligned}
& X_t + Y_t - Y_{t-1} = S_t \quad t = 1, ..., T+1 \\
& Z_{t-1} + X_t - Z_t \leq D_t \quad t = 1, ..., T+1 \\
& Z_0 = Z_{T+1} = 0 \\
& Y_0 = Y_{T+1} = 0 \\
& X_t, Y_t, Z_t \geq 0
\end{aligned} \tag{2.3}$$

The useful output of this model is actually the optimal PAARs (X_t) for this single airport. The decision variables Y_t and Z_t present ground and air delays for each time step respectively. c_g and c_a are the weighted cost for the ground delays and airborne delays. Since airborne delay is more expensive, the ratio of the two costs are set as $c_a/c_g = 2$.

The parameter S_t is the number of scheduled arrival flight for interval t. All flights are enforced to arrive within the time horizon by the first constraints in Eq. 2.3, since all the scheduled flight are absorbed by either PAARs or ground holdings and the ground holdings are emptied by the fourth constraint in the end.

The parameter D_t is the deterministic landing capacity for interval t. The second constraints ensure that the actual number of arrivals should not exceed landing capacity, since the extra flights will be held in the air. Solving model described by Eq. (2.3) will provide the optimal PAARs (X_t) for the airport.

2.3.2 Static Model

The static model was introduced by Ball et al. in 2003 [33], which is a static approach to choose PAARs under CDM procedures. It is static because decisions

are made based only on the current state and do not take into account updated information [44]. The formulation is

$$\textbf{min} \sum_{t=1}^{T}(c_g Y_t + \sum_{q=1}^{Q} p_q c_a Z_{t,q})$$

s.t.

$$X_t + Y_t - Y_{t-1} = S_t \quad t = 1, ..., T+1$$

$$Z_{t-1,q} + X_t - Z_{t,q} \leq D_{t,q} \quad t = 1, ..., T+1, q = 1, ...Q$$

$$Z_{0,q} = Z_{T+1,q} = 0$$

$$Y_0 = Y_{T+1} = 0$$

$$X_t, Y_t, Z_{t,q} \geq 0 \tag{2.4}$$

The difference from the deterministic model described by Eq. (2.3) is that the landing capacity is no longer a deterministic parameter but a random parameter, which follows a landing capacity distribution. To present the landing capacity distribution, the Ball et al. model choose to sample a finite set of landing capacity scenarios with associated probabilities, p_q, where the landing capacity $D_{t,q}$ under each scenario q represents one possible evolution of landing capacity over time.

The static model only considers the airborne delays to be different for each possible scenario q. Therefore, only the $Z_{t,q}$ is modified with the deterministic version, which represents the airborne delays in scenario q. All the other variables and parameters are the same with deterministic case. Similarly, the second constraints ensure that the capacity constraints are still hold under each possible scenario q. Solving the stochastic programming model described by Eq. (2.4) will provide the optimal PAARs X_t associated with the expected optimal solution.

2.3.3 Chance Constrained Model

The chance constrained model aims to incorporate the constantly changing landing capacities, which are caused by adverse weather conditions, into the SAGDP. The

current models are rather deterministic or based on predefined scenarios (like the Ball et al. model). This section proposes to impose a probabilistic constraint on landing capacities, as follows:

$$\mathbb{P}\left(Z_{t-1} + X_t - Z_t \leq \xi_t\right) \geq \alpha \quad t = 1, ..., T+1 \tag{2.5}$$

where $\mathbb{P}(\cdot)$ is the probability measure for the stochastic landing capacities, meaning that the landing capacity will only raise a feasibility issue with the probability of $\alpha \in (0, 1)$, where α is the service level defined in section 2.2. The random components ξ_t are random parameters that represent the stochastic landing capacities. Thus, the SAGHP under the stochastic landing capacities can be written as:

$$\begin{aligned}
&\textbf{min} \sum_{t=1}^{T} (c_g Y_t + c_a Z_t) \\
&\textbf{s.t.} \\
&\quad X_t + Y_t - Y_{t-1} = S_t \quad t = 1, ..., T+1 \\
&\quad \mathbb{P}\left(Z_{t-1} + X_t - Z_t \leq \xi_t\right) \geq \alpha \quad t = 1, ..., T+1 \\
&\quad Z_0 = Z_{T+1} = 0 \\
&\quad Y_0 = Y_{T+1} = 0 \\
&\quad X_t, Y_t, Z_t \geq 0
\end{aligned} \tag{2.6}$$

The difference from the deterministic model is that the capacity constraints are replaced with the probabilistic capacity constraints (2.5). This problem is referred to *chance constrained SAGHP*. The chance constrained model directly uses the landing capacity distribution rather than generating predefined scenario set from the distribution (like the Ball et al. model).

2.3.4 Service Level Evaluation

The current system's service level can be evaluated by running multiple Monte Carlo simulations. For a specific scheduling plan, we run it for 10000 times. In each

time, the capacities are one realization of the uncertain parameters. Then the service level is the percentage of the successful tasks, in which the capacity constraints are not violated at all.

Setup

To evaluate the service level, the first thing needed is a distribution of landing capacity for a single airport. The San Francisco International (SFO) airport is chosen to be the target airport in this section. The observed landing capacities of SFO for 368 days from May 2015 to October 2015 and from May 2016 to October 2016 is analyzed. The data was extracted from the Aviation System Performance Metrics (ASPM) database [45]. The distribution of landing capacities for SFO is shown in Figure 2.1. The distributions are estimated by the Kernel Density Estimation (KDE), which is a non-parametric way to estimate the probability density function (PDF) without assuming any distributional priori property [46]. In Figure 2.1, the empirical data is shown in green bar chart and the KDE-based PDF is shown as the red dashed line. The individual distributions are demonstrated to be very close to the normal distribution. Figure 2.1 shows the cumulative distribution function (CDF) of a normal distribution (blue line) and the CDF for the estimated distribution SFO airport (red line).

The chance constrained model for SAGHP problem is a very special case that the landing capacity has a individual distribution with only one random parameter. From the CDF in Figure 2.1, it can be easy to get the associated capacity limit to satisfy the chance constraint. For example, if the service level $\alpha = 0.8$, it means

$$\mathbb{P}\left(Z_{t-1} + X_t - Z_t \leq \xi_t\right) \geq 0.8 \tag{2.7}$$

Then, from the CDF in Figure 2.1, it is easy to know $\mathbb{P}(31 \leq \xi_t) = 0.8$. Therefore, the chance constraint is equivalent to

$$Z_{t-1} + X_t - Z_t \leq 31 \tag{2.8}$$

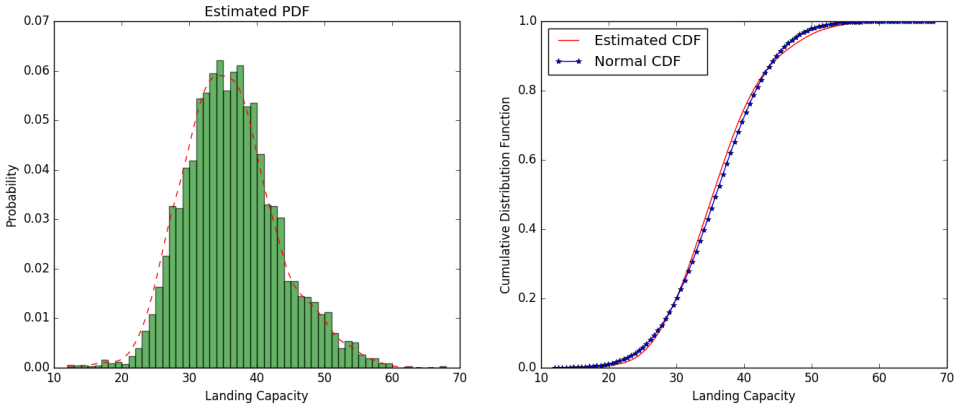


Figure 2.1. Landing distribution at SFO

which means the chance constraint can be equivalently transformed to a deterministic capacity limit with a static capacity reduction. After the equivalent transformation, the chance constraint model for SAGHP can be solved as a deterministic case. The equivalent arrival capacity associated with 80% service level, 31, is used in the evaluation.

Scenarios for the Ball et al. model were generated by successively sampling landing capacities from the distributions. Different scenario samples will result in different minimum expected costs. We chose to sample 500 scenarios because the expected cost almost keeps the same with more than 500 scenarios. Each of the 500 scenarios is assigned with the same 1/500 probability to calculate the expected cost of the static model.

The deterministic capacity is chosen to be the average estimated arrival capacity. Actually, the capacity profile shows the hourly throughput that an airport is able to sustain during periods of high demand, represented as the range of estimated arrival and departure capacity [47]. Each weather condition has a unique capacity range, and an arrival or departure priority operation also affects the capacity profile. That is because they need to share the same runway resource. For example, the SFO airport

capacity under visual weather condition is shown in Figure 2.2. The estimated rate is 55 (arrivals) and 45 (departures) in visual conditions, and the estimated maximum arrival priority rate is 73 in visual conditions. Therefore, the estimated normal arrival capacity, 55, is used as the capacity limit in the service evaluation.

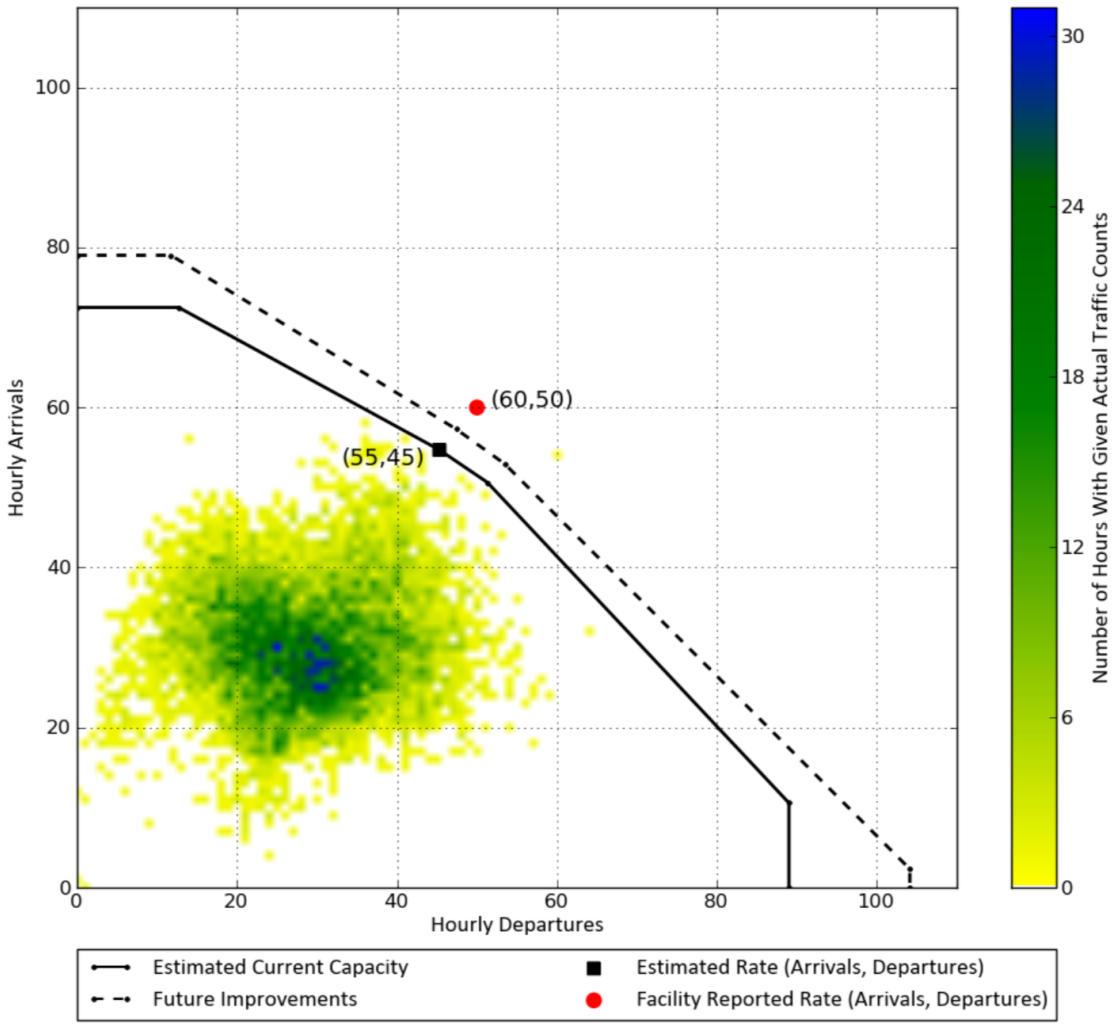


Figure 2.2. The SFO airport capacity under visual weather condition, captured from Ref. [47]

The three models of SAGHP were evaluated using the same flight schedules for SFO on 20 May 2016. The number of arrivals per hour for this day for SFO, taken

from the ASPM database, is shown in Figure 2.3. We discretized the flight schedule and modified the last time interval with infinite capacity to ensure all flights could land within the time horizon. Solutions to both the deterministic and scenario-based model were found using the Gurobi mathematical programming solver [48].

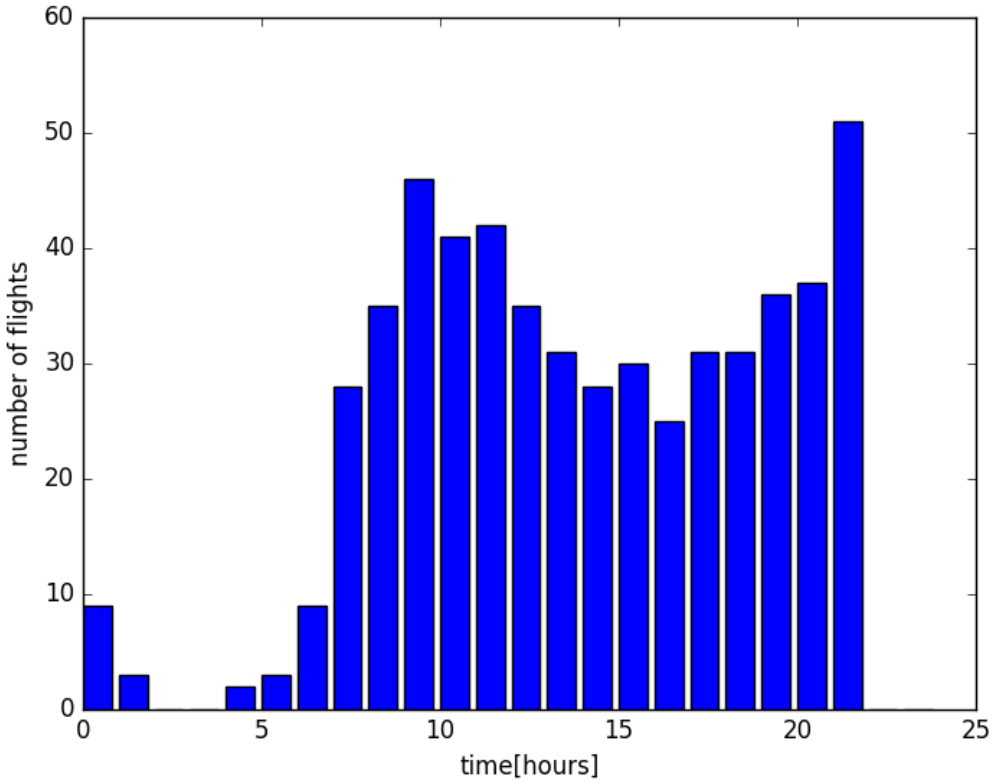


Figure 2.3. Arrivals schedule at SFO

Result

The optimal PAARs results from three methods are shown in Figure 2.4. The PAARs in deterministic case are exactly the same with the schedule because the demand for every step is below the chosen deterministic capacity. The PAARs from

the scenario tree based static method is a little more conservative. Some flights are delayed to a later step in case of any possible bad scenario associated with severe weather. The solution of scenario tree based model depends on the probability estimation of the bad scenario. The higher the probability of bad scenario is estimated to be, the more delays will be assigned. The chance constrained model appears to be the most conservative solution, where the capacities for rush hours are limited to 31 by the 80% service level requirement. Indeed, the 80% is a high service level requirement and it is reasonable to get such a conservative solution.

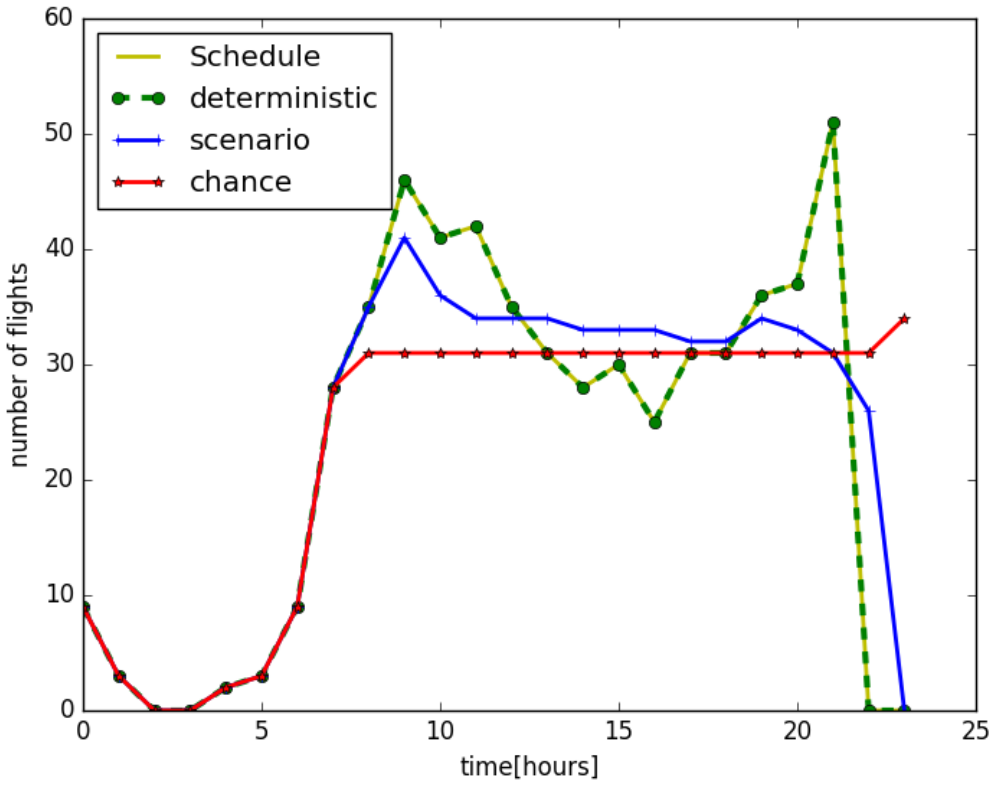


Figure 2.4. The optimal PAARs of SFO under 0.8 service level

Based on the above PAARs, the service level evaluation is performed for each result with the method in Section 2.2.2. For each PAAR, the Monte Carlo simulation

is run for 10000 times. The service level for each method is shown in Figure 2.5. It is obvious that the chance constraint model successfully keeps the percentage of successful tasks to be above the required service level, 0.8, at each time step. However, the service level for the deterministic and scenario tree method are both very low, especially during the rush hours. Moreover, the scenario tree method is still better than the deterministic model in service level. That is because the scenario tree method still used the distribution information.

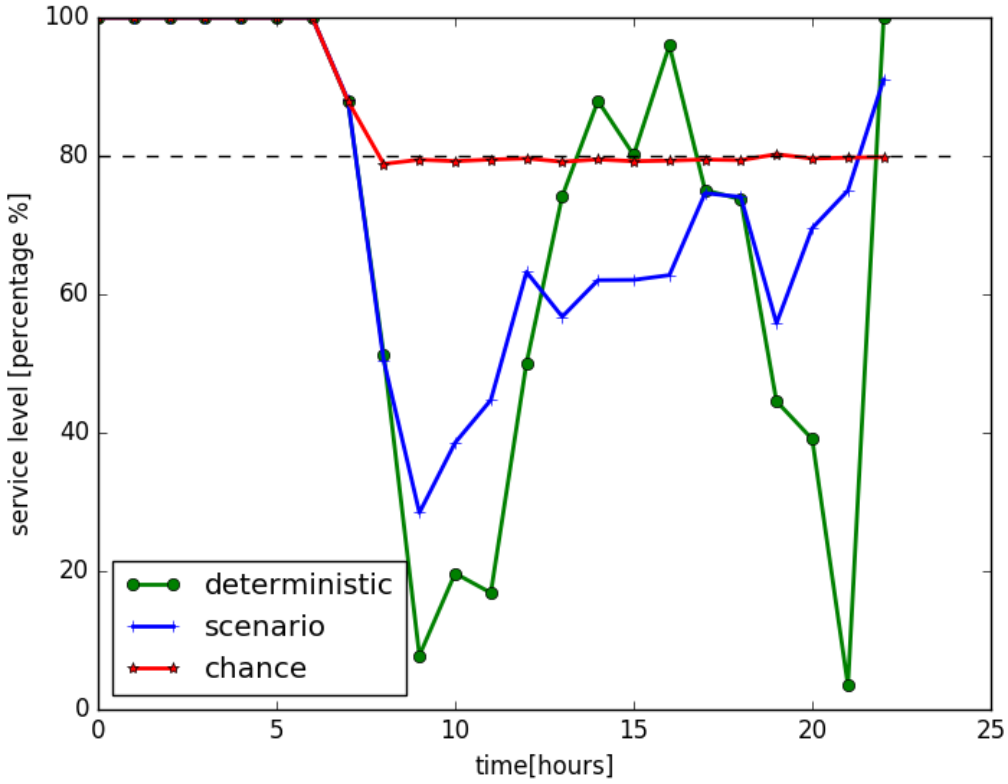


Figure 2.5. The service level for each time step at SFO

2.4 Chance Constrained Model for MAGHP in a Metroplex

The SAGHP is a very simple case for chance constrained model, where the chance constraints can be easily transformed to an equivalent deterministic case by finding the corresponding static capacity reduction under the required service level. In reality, weather often affects multiple airports simultaneously, especially in a metroplex. Only correlated stochastic joint capacity distribution is meaningful for GDP planning in a metroplex. However, it is often difficult to get the corresponding static capacity from a joint capacity distribution, and there may be multiple capacity combinations that associated with the same service level requirement. Therefore, a new method with solving approach is needed, which is introduced in detail in Chapter 3. This section will demonstrate the chance constrained modeling for MAGHP.

The proposed chance constrained model for MAGHP is derived based on the previous static model for the SAGHP [33]. First, the static model is modified to schedule PAARs for all airports in a metroplex simultaneously. Then the same problem is solved by the newly proposed chance constrained model for comparison. Then the service level is evaluated for both of the static and chance model. In the end, the impact for adjustable service level is studied and demonstrated.

2.4.1 Static Model

The same static model in Section 2.3 is adopted here for choosing PAARs under CDM procedures. We choose to modify the Ball et al. model to consider multi-airports in a metropolitan area (e.g. New York metroplex). The formulation is

$$\textbf{min} \sum_{t=1}^{T} \sum_{i=1}^{M} (c_g Y_t^i + \sum_{q=1}^{Q} p_q c_a Z_{t,q}^i)$$

s.t.

$$\begin{aligned}
& X_t^i + Y_t^i - Y_{t-1}^i = S_t^i \quad t = 1, ..., T+1, i = 1, ...M \\
& Z_{t-1,q}^i + X_t^i - Z_{t,q}^i \leq D_{t,q}^i \quad t = 1, ..., T+1, i = 1, ..., M, q = 1, ...Q \\
& Z_{0,q}^i = Z_{T+1,q}^i = 0 \\
& Y_0^i = Y_{T+1}^i = 0 \\
& X_t^i, Y_t^i, Z_{t,q}^i \geq 0
\end{aligned} \tag{2.9}$$

The difference is that multiple SAGHPs are combined together by sampling the landing capacity of each airport $D_{t,q}^i$ simultaneously from a joint distribution. Similarly, to present the joint landing capacity distribution, we choose to sample a finite set of landing capacity scenarios with associated probabilities, p_q, where each scenario q represents one possible evolution of landing capacity over time. All the parameters are similar with the previous definitions in Section 2.3. The only difference is the superscript "i", which represents the parameter is associated with airport i. For example, the parameter S_t^i is the number of scheduled arrival flight for interval t, airport i. The similar meanings are represented by Y_t^i and $Z_{t,q}^i$.

All flights are enforced to arrive within the time horizon by the first constraints in Eq.2.9, since all the scheduled flight are absorbed by either PAARs or ground holdings and the ground holdings are emptied by the fourth constraint. The landing capacity limit is ensured by the second constraints and the extra flights will be held in the air. Solving model described by Eq. (2.9) will provide the optimal PAARs X_t^i for each airport, respectively.

2.4.2 Chance Constrained Model

A similar chance constrained model is introduced to the MAGDP, which corresponding to the constantly changing joint landing capacities that are caused by adverse weather conditions. The proposed probabilistic constraint on joint landing capacities is as follows:

$$\mathbb{P}\left(Z_{t-1}^i + X_t^i - Z_t^i \leq \xi_t^i, \quad i = 1, ..., M\right) \geq \alpha \quad t = 1, ..., T+1 \tag{2.10}$$

where $\mathbb{P}(\cdot)$ is the probability measure for the stochastic landing capacities, meaning that the landing capacity will only raise a feasibility issue with the probability of $\alpha \in (0, 1)$, where α is still service level. The random components ξ_t^i are random parameters that represent the correlated, stochastic landing capacities, and only correlated random capacities are meaningful for the MAGDP planning because adverse weather conditions will usually affect multiple airports of the metroplex simultaneously. Thus, the MAGDP planning under the stochastic landing capacities can be written as:

$$\textbf{min} \sum_{t=1}^{T} \sum_{i=1}^{M} (c_g Y_t^i + c_a Z_t^i)$$

$$\textbf{s.t.}$$

$$X_t^i + Y_t^i - Y_{t-1}^i = S_t^i \quad t = 1, ..., T+1, i = 1, ...M$$

$$\mathbb{P}\left(Z_{t-1}^i + X_t^i - Z_t^i \leq \xi_t^i, \quad i = 1, ..., M\right) \geq \alpha \quad t = 1, ..., T+1$$

$$Z_0^i = Z_{T+1}^i = 0$$

$$Y_0^i = Y_{T+1}^i = 0$$

$$X_t^i, Y_t^i, Z_t^i \geq 0 \tag{2.11}$$

The difference from the deterministic model is that the capacity constraints are replaced with the probabilistic capacity constraint (2.10). This problem is referred to *chance constrained MAGDP optimization.*

2.4.3 The Solving Framework

The solving approach for chance constrained model is summarized in Chapter 3. The general idea is to use convex approximation to approximate the original chance constraints based on one assumption that the landing capacity distribution follows a log-concave distribution. Only a summarized solving framework is shown in this section. The details of the solving approach is introduced in Chapter 3.

Based on the approximation approach in the Chapter 3, the function values and gradients for the chance constraint $g_t(x)$ can be estimated for any given point $\bar{x}$. Therefore a first-order algorithm can be adopted to solve the whole problem. In this section, the feasible direction method [49] is adopted as the primary algorithm

The flowchart of the solving algorithm for the chance constrained problem is shown in Figure 2.6. Based on that the construction of the polynomial approximation for each individual marginal function is independent. Therefore, at each step, the chance constraint can be approximated in parallel at the given point $\bar{x}$. Then the results are gathered to provide the first-order information, which is used to search for the feasible direction and optimal search step. Note that the algorithm needs to call the approximation process during every iteration until the final converge. Therefore, the parallel computing framework can greatly improve the computational efficiency by the fact that the approximation process has the most expensive computing cost of the whole process.

2.4.4 Experimental Setup

The assumption that the landing capacity distribution follows a log-concave distribution, but lacks closed form distributional information, would be justified with two phases. First, in reality of air traffic management, the historical data from the landing capacity distribution is in the form of empirical distribution [50]. By using proper distribution estimation methods (such as the Kernel Density Estimation), the empirical distribution will be presented as a continuous distribution without the dis-

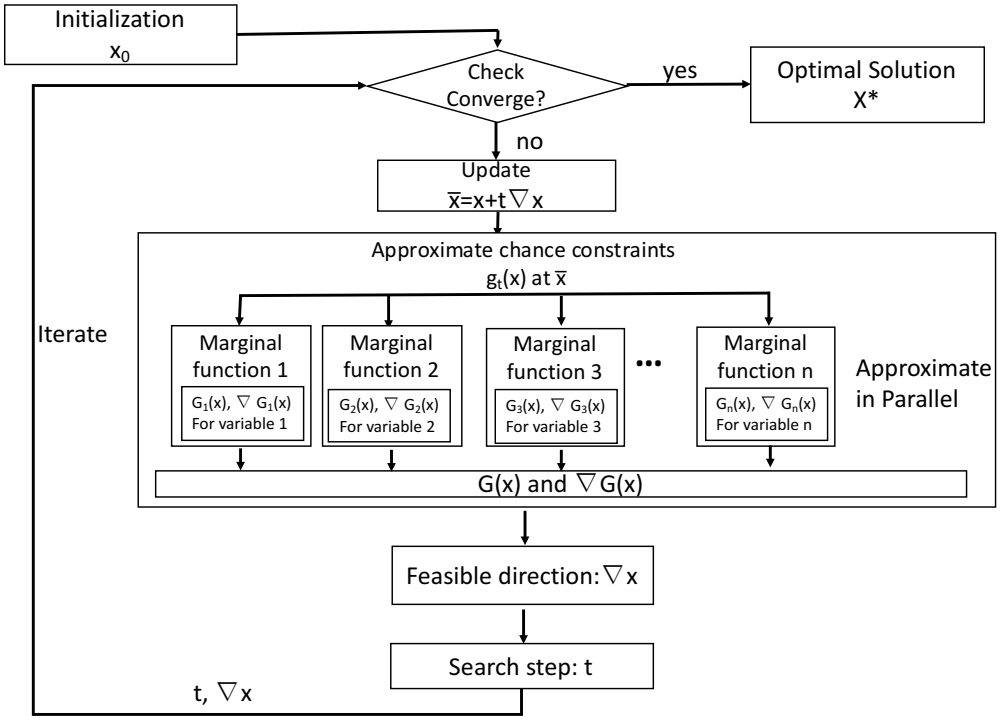


Figure 2.6. The algorithm flowchart for the chance constrained problem

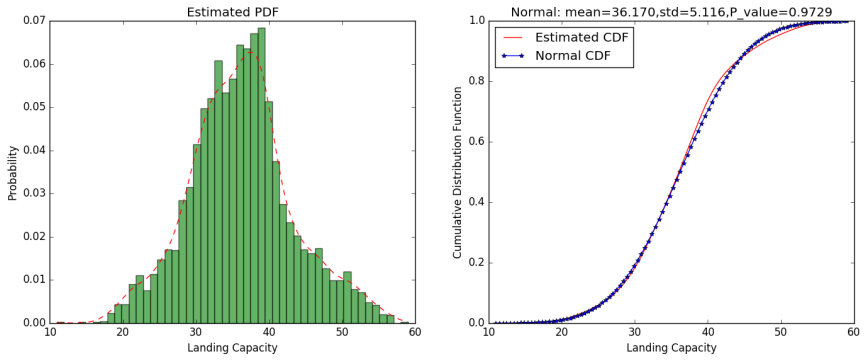
tributional information. By the Glivenko-Cantelli Theorem (see [51]), the empirical distribution function estimates the cumulative distribution function and converges with a probability of 1. That is, the empirical distribution can be presented as an underlying continuous distribution. Second, once the empirical distribution is in the format of a continuous distribution (but still lacks distribution information), this paper would further assume logconcavity because so many commonly used distributions are, indeed, log-concave. For example, the normal distribution, uniform distribution, gamma distribution (with a shape parameter greater than 1), beta distribution (with all parameters greater than 1), Weibull distribution, Laplace distribution, logistic distribution, exponential distribution and extreme value distribution are log-concave. There, are only a few commonly used distributions that are not log-concave, such as the lognormal distribution, t-distribution, and Chi-square distribution, which are of-

ten used to describe the distributions of various statics rather than random variables raised from real problems.

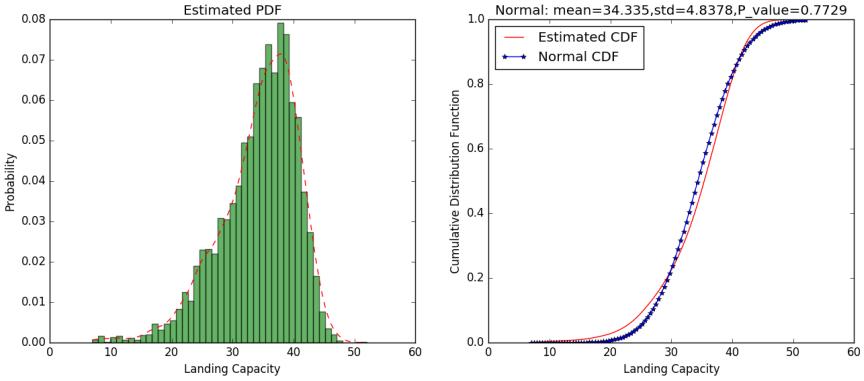
To evaluate the chance constrained model for MAGDP, a joint distribution of landing capacities is required. Moreover, to further support the log-concave distribution assumption, we would like to justify it with real empirical data. We analyzed the observed landing capacities of NYC metroplex (JFK, EWR and LGA) for 368 days from May 2015 through October 2015 and from May 2016 through October 2016. Landing capacities were computed based on the algorithm in references [44, 52] and the "arrivals for metric computations" data from the ASPM database [45]. The individual distribution of landing capacities for each airport is shown in Figure 2.7.

The distributions are estimated by the Kernel Density Estimation, which is a non-parametric way to estimate the PDF without assuming any distributional priori property [46]. In Figure 2.7, the empirical data is shown in green bar chart and the KDE-based PDF is shown as the red dashed line. The individual distributions are demonstrated to be very close to the normal distribution. Figure 2.7 shows the cumulative distribution functions of a normal distribution (blue line) and the CDF for the estimated distribution for each airport (red line). A Kolmogorov-Smirnov test of normality was also applied and the result, referred as a p-value, is indicated in the title of each sub-figure. The p-value generally indicate that the estimated distribution can be reasonably approximated as normal at the significant level 0.05. Therefore, the empirical landing capacities of all airports in NYC metroplex are fitted by KDE to be a joint multi-normal distribution, which is log-concave. Please note that the convex approximation method can work with any log-concave distribution, the normality is not required for general cases. The covariance matrix is shown in Table 2.1, which confirms the correlation between the airports in the same metroplex.

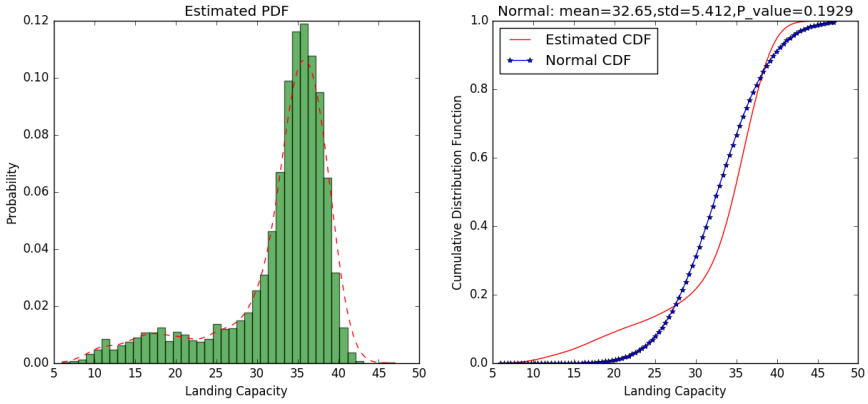
Scenarios for the Ball et al. model were generated by successively sampling landing capacities from the joint distributions. Different scenario samples will result in different minimum expected costs. The Figure 2.8 shows the minimum expected cost along various sample sizes. The blue area is the 95% confidence interval for each



(a) Landing Distribution at JFK



(b) Landing Distribution at EWR



(c) Landing Distribution at LGA

Figure 2.7. Landing distributions for NYC metroplex

Table 2.1. Covariance matrix for the joint distribution of NYC metroplex landing capacities

	JFK	EWR	LGA
JFK	4.604	0.805	0.2592
EWR	0.805	3.652	1.633
LGA	0.2592	1.633	4.407

sample size. We chose to sample 500 scenarios because the 95% confidence interval of the expected cost almost keeps the same with more than 500 scenarios. Each of the 500 scenarios is assigned with the same 1/500 probability to calculate the expected cost of the static model in Section 2.4.1.

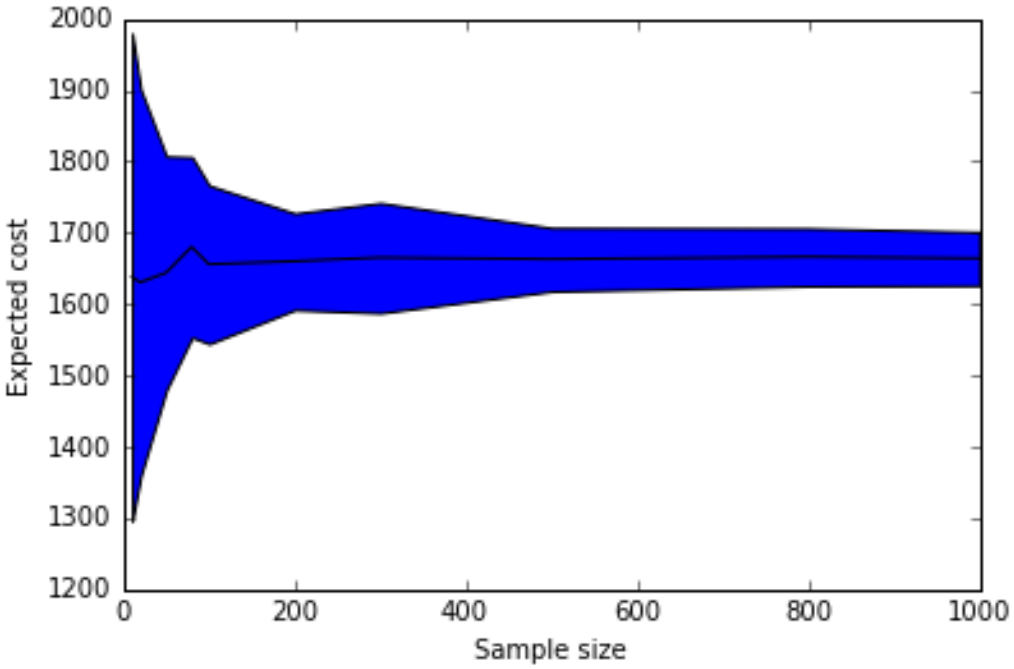
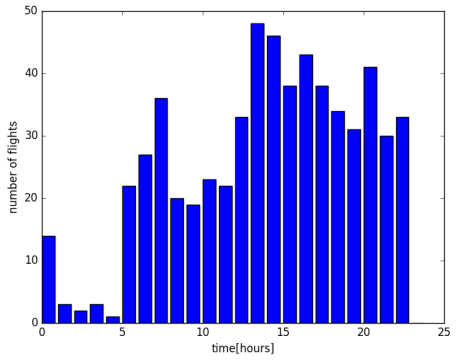
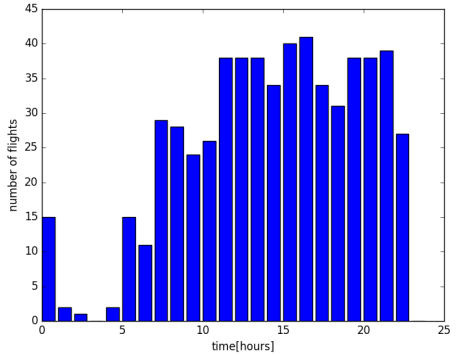


Figure 2.8. Objective converge along the number of sample scenarios

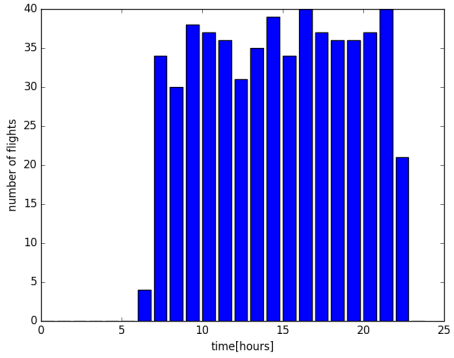
The two approaches of MAGDP were evaluated using the same flight schedules for NYC metroplex on 20 May 2016. The number of arrivals per hour for this day for



(a) Arrivals schedule at JFK



(b) Arrivals schedule at EWR



(c) Arrivals schedule at LGA

Figure 2.9. Arrival schedules for NYC metroplex airports on May 20 2016

all airports, taken from the ASPM database, is shown in Figure 2.9. We discretized the flight schedule and modified the last time interval with infinite capacity to ensure all flights could land within the time horizon. Solutions to the Ball et al. model were found using the Gurobi mathematical programming solver [48]. The chance programming was implemented in the Python programming language.

2.4.5 Service Level Evaluation

The detailed results of optimal PAARs from static and chance model for three airports in NYC metroplex are shown in Figures 2.10 to 2.12. In all of them, the red line shows the optimal PAARs for each hour with Ball et al. model, referred as scenario, and the yellow line represents the optimal PAARs with the chance constrained model under service level 0.8. For each of the individual airport, by comparing both the scenario based and chance based result with the flight schedule (the blue line), we can find that both of the optimal PAARs are almost identical with the schedule before 12:00, which represents the slack time. In the peak period, the scenario-based method slightly reduced the number of planned arrivals and compensate the schedule in the last time step. The assigned delays account for any possibility of bad scenarios under severe weather.

By comparing the results across all the three airports, the solutions of the scenario-based method have the similar trend, and all of them appears to be independent with each other. However, the chance constrained model provides totally different solutions with the scenario ones. The results of chance constrained model illustrate the influence of joint planning associated with the joint landing capacity distribution. With the goal to satisfy the joint 80 % service level, the three airports collaborate with each other. For example, the PAAR is low at 16:00 in JFK, but it is relatively high in both EWR and LGA at the same time. The similar pattern can be observed at 18:00, where the PAAR is high in JFK and is relatively low in EWR and LGA.

Therefore, they keep the joint 80 % service level by sharing the joint information and planning together.

The above joint planning pattern is further confirmed by the service level evaluation. Based on the solved optimal PAARs, the same service level evaluation method in Section 2.2.2 is performed. The Monte Carlo simulation is still run for 10000 times for each time step. The difference is that one sample from the joint landing capacity distribution is a 3-by-1 vector, corresponding one possible combination of traffic demand, rather than a single scale number in the previous SAGHP. The result of service level evaluation is shown in Figure 2.13. The red line represents service level based on the chance constraint model, which is almost kept above the 80% line at each time

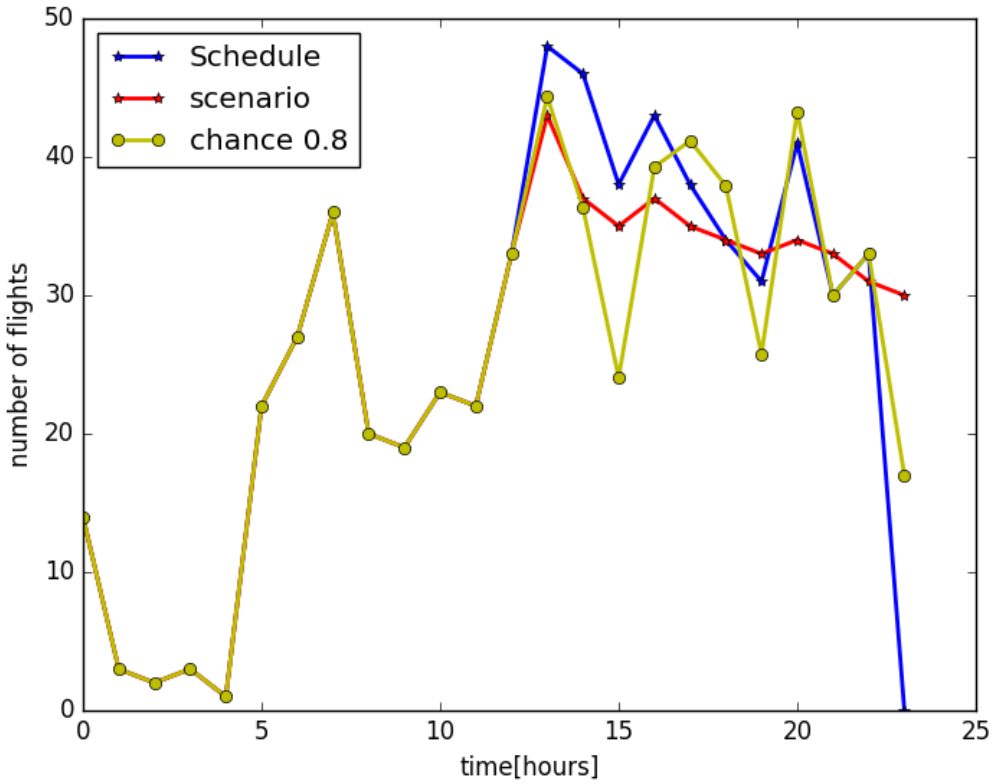


Figure 2.10. Optimal PAARs for JFK on May 20 2016 under service level 0.8

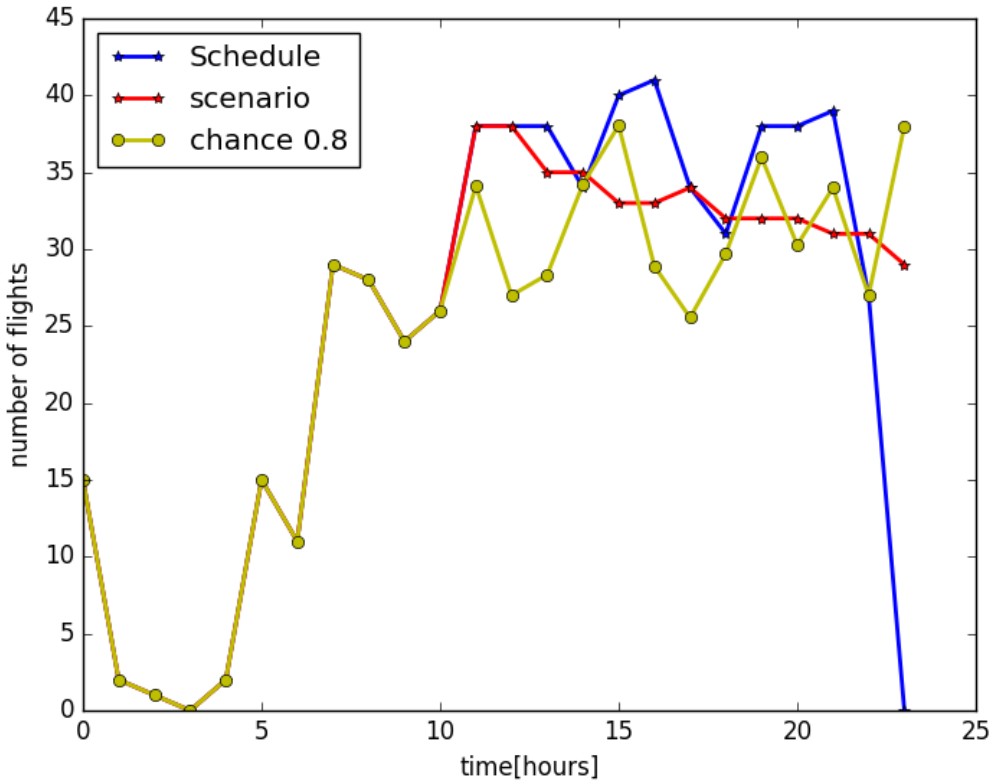


Figure 2.11. Optimal PAARs for EWR on May 20 2016 under service level 0.8

step. However, the scenario model (blue line) and deterministic model (green line) have weak performance on service level evaluation, especially during rush hours. Here the deterministic case uses the same plan with the scheduled flight, which means no control is performed. The service level is as low as 10% percent, which demonstrates the necessary of stochastic planning. By considering the information of landing capacity distribution, the scenario model indeed improved the average service level, but the service level is not controllable. Only the chance constrained model provides the optimal solution with guaranteed service level, which corresponding to the balance between risk and robustness.

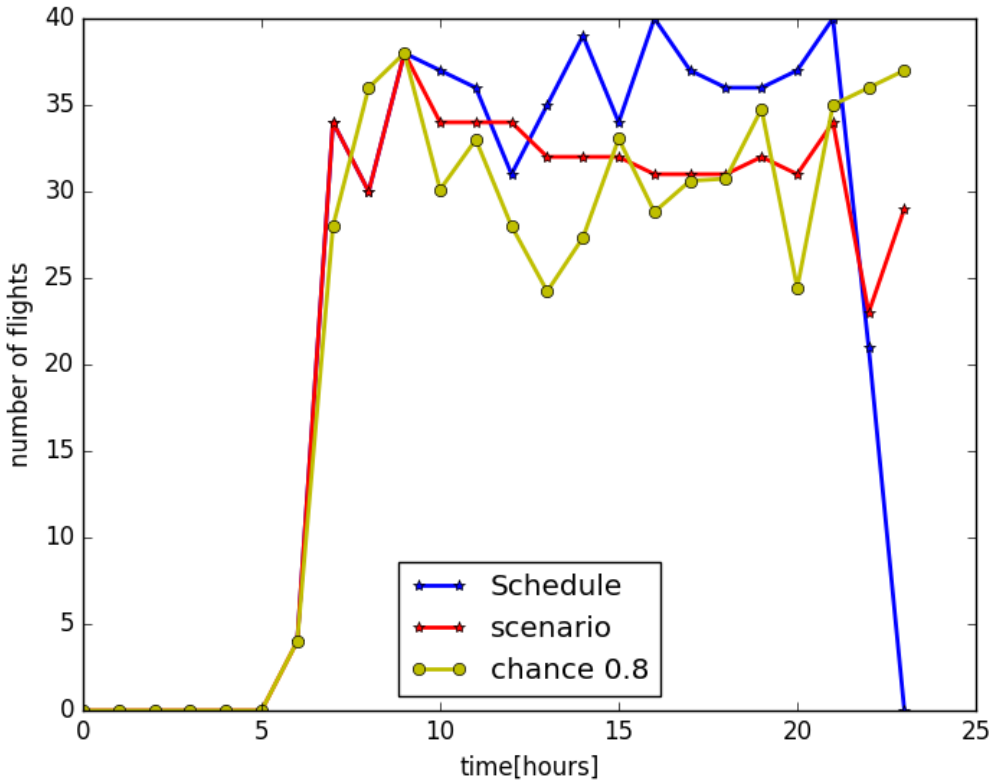


Figure 2.12. Optimal PAARs for LGA on May 20 2016 under service level 0.8

2.4.6 Impact of Adjustable Service Level

One of the key advantages of the chance constrained model is the ability to provide robust solutions with user-defined service level. Figure 2.14 shows the total delays of NYC metroplex airports under various service level from 0.5 to 0.9 and the delays with 500 scenarios static model. The relative cost ratio of air to ground delay is chosen to be 2, $c_a/c_g = 2$. In Figure 2.14, the number of delays will increase with the service level, which is consistent with the intuition that the high service level will result in conservative solutions. This is also confirmed by the ratio of ground to air delays. The numbers above each bar in Figure 2.14 represent the ratio of ground to air

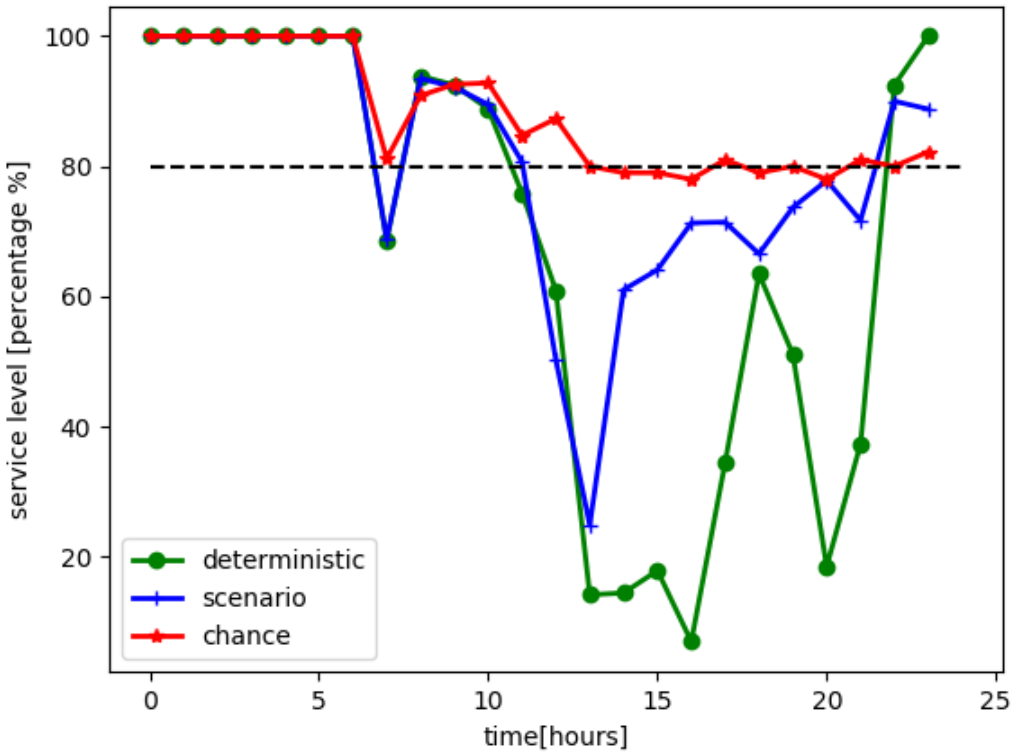


Figure 2.13. The service level for MAGHP at each time step

delays. The low service level, which represents aggressive planning, produces more air delays than ground delays. On the other hand, the high service level, which represents conservative planning, results in more ground delays than air delays. However, the scenario-based method (Ball et al. model) will only produce one average result with respect to the minimum expected cost, which lacks the ability to adjust the planning strategy under different service levels.

Figure 2.15 shows the details of the optimal PAARs at JFK under various service level. The red line shows the optimal PAARs for each hour with Ball et al. model, referred as scenario, and the yellow line represents the optimal PAARs with the chance constrained model under certain service level. By comparing both the results with the

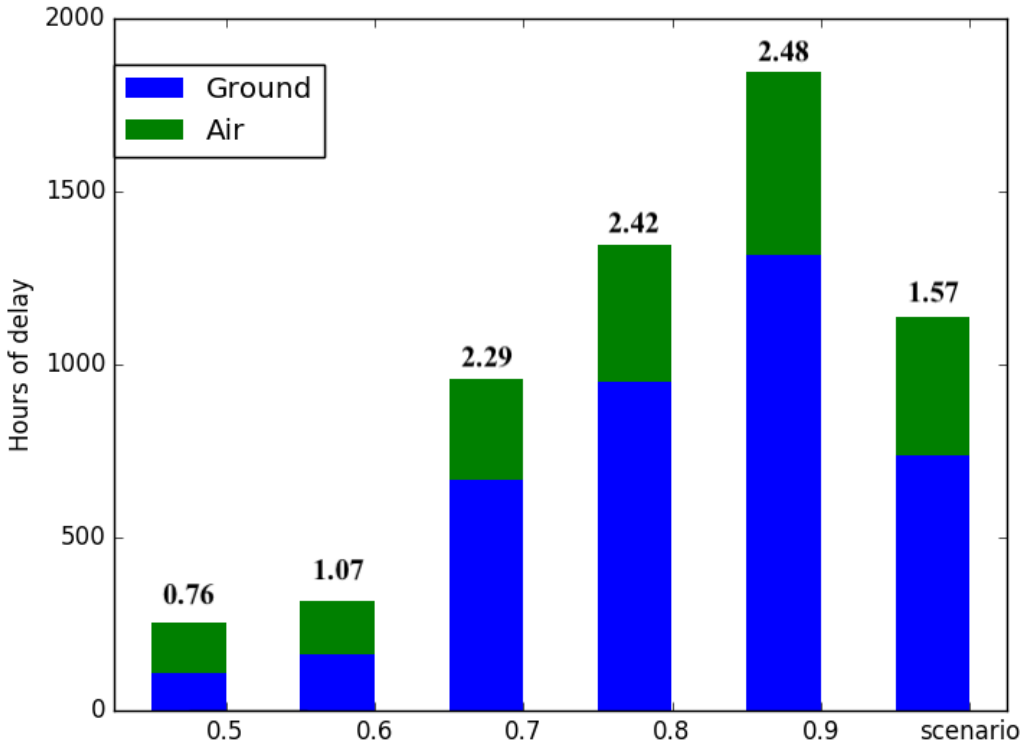


Figure 2.14. The ground and air delay under various service level

flight schedule (the blue line), we can find that all of the optimal PAARs are almost identical in the slack time (early morning). In the peak period, the scenario-based method slightly assigned more delays for the arrivals and compensate the schedule in the last time step. However, the chance constrained model generates totally different solutions. In general, the low service level will provide aggressive planning, in which the PAAR almost follows the schedule. On the other hand, the conservative planning under the high service level will assign more ground delays in the beginning to avoid possible air delays and then compensate the schedule in the latter time step. Meanwhile, the similar results are observed for other two airports (EWR and LGA) in 2.16 and 2.17. Therefore, the results demonstrate that the the conservativeness level is

positive correlated with the robustness level. The chance constrained model, introduced in this chapter, provides a quantized way to balance the solution's robustness and potential cost by choosing a proper service level.

2.5 Discussion

Different service levels represent different reliable/risk levels. Low service level will produce result with high risks. To be specific, although the total delay with service level 0.5 is very small, there are 50% chance that the capacity constraints are violated, which means the tasks will fail with 50% chance. To help clarify the service level, a similar example is given as follows: if there are two machines, one costs little but has 50% chance to be broken, another one costs much but has only 10% chance to be broken. Which one will the user choose? The choice is actually based on the users preference of risk-cost balance, which is the service level.

The service level can be adjusted based on the weather prediction. If the service provider is very confident about the good weather condition in the planning horizon, then a low service level could be chosen which has low cost/delay. However, if there is a high chance to have convective weather condition, a high service level solution should be chosen to ensure the success of the schedule. Since air transportation has a high requirement for safety, we would recommend for high service level. The common choice is 80% or 85%, since it is a similar idea with the confidence level in statistics.

The main purpose of this chapter is to provide an optimal solution once the service level is defined. However, the selection of the service level is also a very important prior process if the idea of "service level" will be integrated into the future ATM system. It is also highly related to the safety requirement and operational cost. It is suggested to research on the selection of the service level and the evaluation of the service level in the future work.

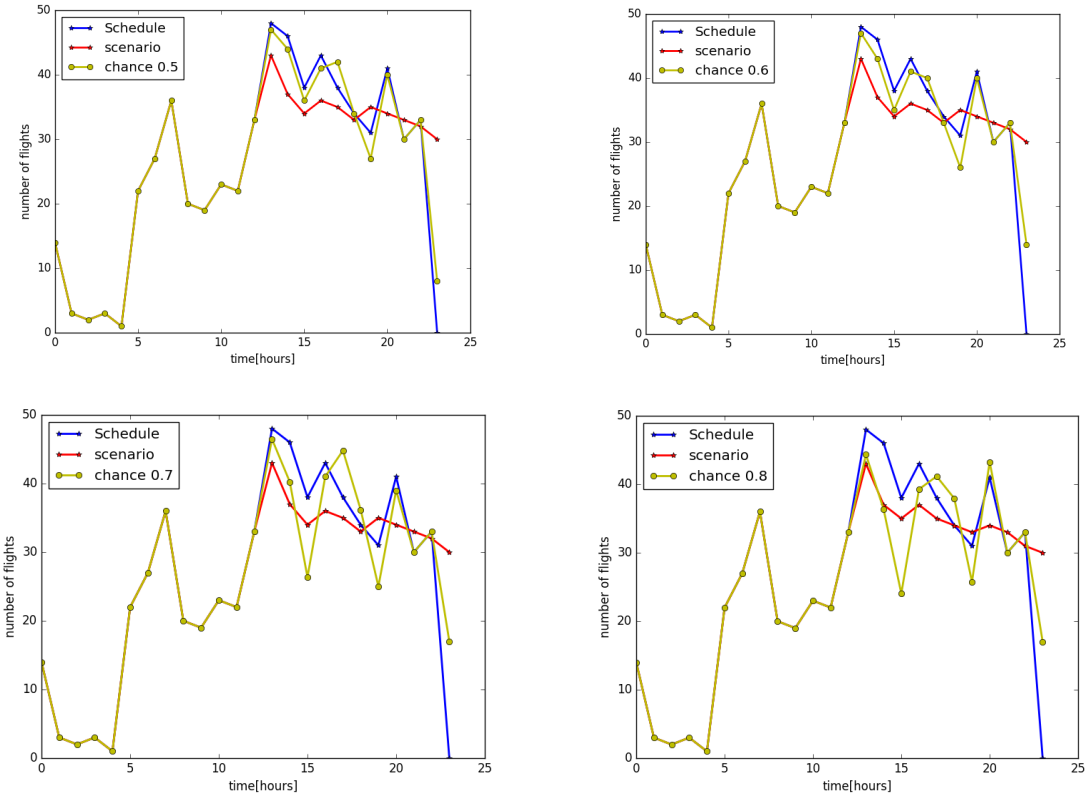


Figure 2.15. The optimal PAARs of JFK under various service level

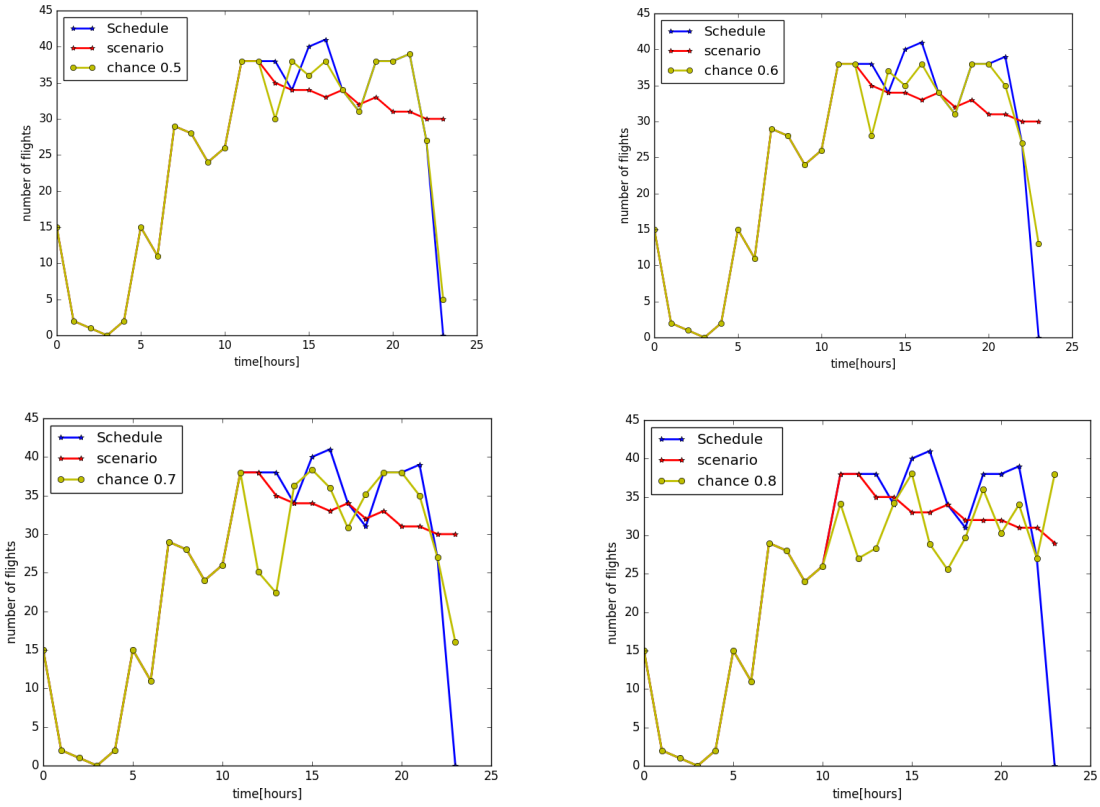


Figure 2.16. The optimal PAARs of EWR under various service level

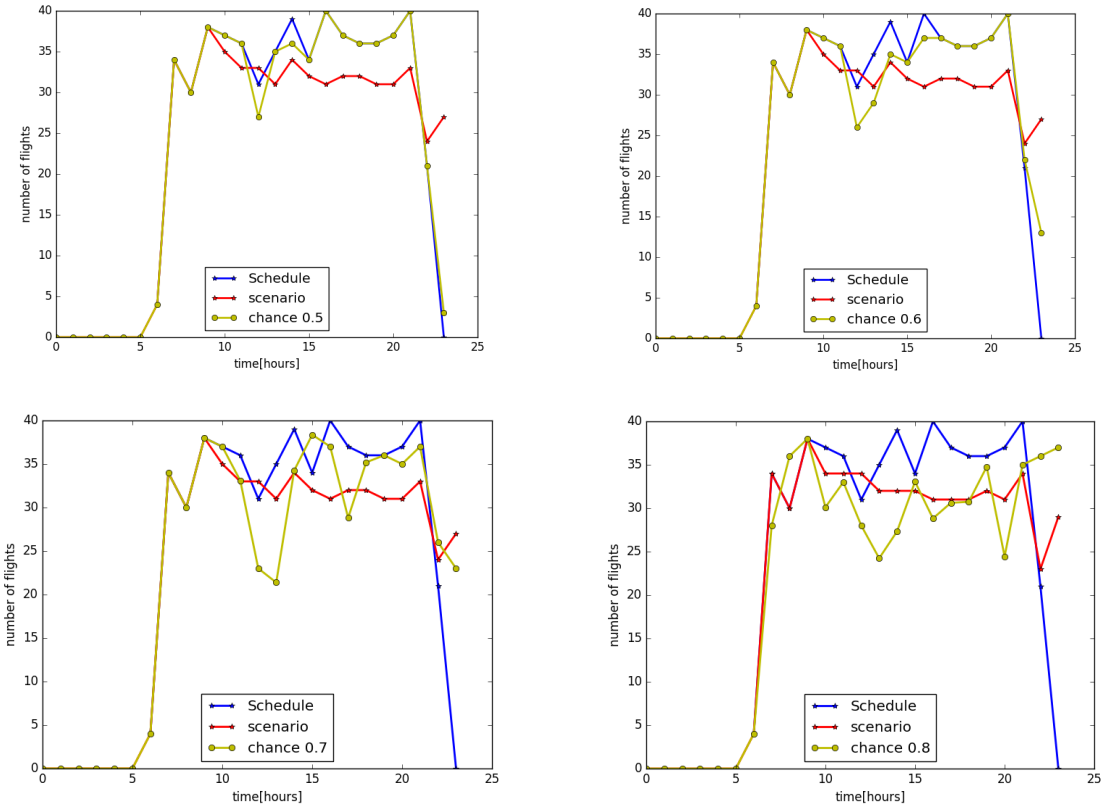


Figure 2.17. The optimal PAARs of LGA under various service level

2.6 Conclusion

This chapter introduces a novel chance constrained approach for ground delay program planning under uncertainty. The major advantage of the chance constrained model is the ability to provide robust solutions with user-defined service level. First, the concept of service level is introduced for the air traffic management, which represent the chance of the constraints not being violated. Then the approach is compared with the Ball et al. model and deterministic model for selecting planned airport acceptance rates for the single airport SFO and the airports (JFK, EWR and LGA) in the NYC metroplex. Although the Ball et al. model was found to be more efficient, the chance constrained model shows the ability to provides a quantized way to balance the solution's robustness and potential cost by choosing a proper service level. Moreover, the service level evaluations demonstrate that the chance constrained model is able to provide solution with guaranteed service level, while the other two methods cannot. The application of this chance constrained method is not only limited to the ground delay program but also helpful in many other domains of air transportation.

3. CONVEX APPROXIMATION APPROACH FOR CHANCE CONSTRAINT

3.1 Problem Definition

The scheduling optimization problem in ATFM subject to capacity constraints is commonly modeled as an integer programming (IP) problem, such as runway scheduling [53–55], arrival sequencing [5,56], and rerouting [16,57,58]. In general, the deterministic IP model can be summarized as a standard optimization form as following:

$$\begin{aligned} \textbf{min} \quad & c'x \\ \textbf{s.t.} \quad & \\ & g_t(x) = R_t x \leq d_t \quad t = 1, ..., T \\ & Ax \leq b \\ & x \in \mathbb{Z}_+ \end{aligned} \tag{3.1}$$

where c represents the vector of the weight coefficients; A, b and R_t are the coefficients vectors corresponding to the original linear constraints, in which t is included as one dimension; $g_t(x)$ represents the capacity constraints for each time step.

By treating the capacities as random parameters, the scheduling problem under stochastic capacities can be modeled as a chance constrained optimization problem with a form:

$$\begin{aligned} \textbf{min} \quad & c'x \\ \textbf{s.t.} \quad & \\ & g_t(x) = \mathbb{P}(R_t x \leq \xi_t) \geq \alpha \quad t = 1, ..., T \\ & Ax \leq b \\ & x \in \mathbb{Z}_+ \end{aligned} \tag{3.2}$$

where $g_t(x)$ becomes the probabilistic capacity constraints for each time step; the random components ξ_t are random parameters that represent the correlated, stochastic capacities; α is the required service level.

The difference from the deterministic model is that the capacity constraints are replaced with the probabilistic capacity constraint, also known as chance programming. The chance programming indicates that some of the constraints may be violated at a well-controlled, very low chance. In general, the chance programming problem is not easy to solve [59]. The traditional solution approach to chance programming is the sample average approximation (SAA). However, the SAA approach becomes intractable quickly due to the exponential growth of state space with the number of sampled scenarios. Moreover, the SAA approach will only yield a feasible solution rather than a optimal solution. Therefore, this chapter will propose a convex approximation method to efficiently solve the chance constrained model, which could overcome the computational limitation of the SAA method when solving large-scale problems.

3.2 The Brute-force Method

In this section, the brute-force method is introduced based on a previous chance-constrained model [38], which is formed as a MILP optimization model. Although

the MILP model can provide accurate solutions to the chance-constrained model, the disadvantage with the exponentially increased computational demands prevents the MILP model from being applied to real operations with large-scale problems. Therefore, this dissertation introduces a convex polynomial-based approximation method to efficiently solve the chance-constrained model, which could overcome the computational limitation of the MILP model when solving large-scale problems.

In order to solve the chance-constrained ATFM problem, the brute-force MILP optimization model is proposed in [38], which is developed based on the Bertsimas-Stock Patterson model [13]. The MILP model enumerates all admissible sector capacity combinations, whose joint probability values satisfy the chance constraints, to form a feasible set for sector capacity combinations. A simple example is illustrated as following: considering just two sectors, ξ_1, ξ_2 are defined to represent the sector capacity as random variables, and the joint probability is shown as Table 3.1. The number of aircraft assigned to each sector is denoted as sn_1 and sn_2, respectively. The goal is to satisfy the chance constraint $P(\xi_1 \geq sn_1, \xi_2 \geq sn_2) \geq \alpha$. Table 3.2 shows the $P(\xi_1 \geq sn_1, \xi_2 \geq sn_2)$ for all combinations of sn_1 and sn_2, based on Table 3.1. If α is defined as being 0.8, only ($sn_1 = 1$, $sn_2 = 1$) and ($sn_1 = 2$, $sn_2 = 1$) can be chosen to form a feasible set. Therefore, the MILP model defines a set of safe capacity limits for the two sectors, which allow them to fulfill the chance constraint.

Table 3.1. Example joint probability

ξ_1, ξ_2	1	2
1	0.06	0.14
2	0.24	0.56

Based on the same idea of the feasible capacity set, the probabilistic constraint (2.8) in Section 2.2 can be replaced by several linear constraints. Suppose that there are m sectors(treating an airport as a sector) in total, the cumulative joint probability

Table 3.2. Cumulative example joint probability

sn_1, sn_2	1	2
1	1	0.7
2	0.8	0.56

matrix for all combinations that can be built, which is an m-dimensional matrix. This m-dimensional matrix is translated into a vector $P(p_i, t)$, in which each element represents the probability for a combination in the set of all admissible sector capacity combinations (denoted as M), for every time step t. A matrix $I(p_i, j)$ is defined to link the elements of $P(p_i, t)$ to their corresponding capacity limits (sn_j) in each dimension of the m-dimension matrix. Moreover, a binary variable $\delta(p_i, t)$ is defined as an indicator for which element in $P(p_i, t)$ is activated. The probabilistic constraint (2.3) can be replaced by the following linear constraints:

$$\sum_{(i,k)\in Q_{s_i}} x_i^k(t) \leq \sum_{p_i \in M} \delta(p_i, t) I(p_i, j) \tag{3.3}$$

$$\sum_{p_i \in M} P(p_i, t)\delta(p_i, t) \geq \alpha \tag{3.4}$$

$$\sum_{p_i \in M} \delta(p_i, t) = 1 \quad \delta(p_i, t) \in \{0, 1\} \tag{3.5}$$

Constraint (3.3) enforces the number of aircraft assigned to each sector to be under the safe limits. Constraint (3.4) ensures that the chance constraint will be satisfied. Constraint (3.5) indicates that only one feasible combination can be activated for each time step.

Therefore, the chance-constrained problem can be transformed into a MILP problem. However, it is important to note that the MILP formulation requires the introduction of an additional binary variable for every possible combination of sector capacities, for every time step. For example, considering a one-hour problem with

10 sectors, each sector has 10 possible capacity values. Then 60×10^{10} new binary variables will be introduced. Therefore, the computational complexity of the original problem is increased exponentially, which prevents the MILP formulation from being applicable to the large-scale problem in the real ATM operation.

3.3 Convex Approximation

3.3.1 Log-concave Assumption

The chance constraint would greatly complicate the computational perspective of the problem because of the loss of convexity, in both its feasible set and the constraint itself. Even though it is extremely difficult to solve a chance-constrained optimization for a global optimal solution, there are exceptions. In [60], the author showed that under less restrictive assumptions, the chance-constrained model in Section 3.1 would have a convex feasible set. The constraint would be equivalently transformed into a convex program, which would be efficiently solved, as long as the function and its gradient (or subgradient) are available.

The required assumptions are imposed, based on the log-concavity of the distribution presented in [61]. The associated definitions and theorems are shown as follows:

Definition 3.3.1 *A function $f(z) \geq 0$, $z \in \mathbb{R}^m$ is said to be logarithmically concave (in short form, log-concave), if for any z_1, z_2 and $0 < \lambda < 1$, we have the inequality*

$$f(\lambda z_1 + (1-\lambda)z_2) \geq [f(z_1)]^{\lambda}[f(z_2)]^{(1-\lambda)}$$

If $f(z) \geq 0$ for $z \in \mathbb{R}^m$, then this means that $logf(z)$ is a concave function in $\mathbb{R}^m$.

Definition 3.3.2 *A probability measure defined on the Borel sets of $\mathbb{R}^m$ is said to be logarithically concave (log-concave) if for any convex subsets of $\mathbb{R}^m$: X, Y and $0 < \lambda < 1$ we have the inequality*

$$\mathbb{P}(\lambda X + (1-\lambda)Y) \geq [\mathbb{P}(X)]^{\lambda}[\mathbb{P}(Y)]^{(1-\lambda)}$$

where $\lambda X + (1-\lambda)Y = \{z = \lambda x + (1-\lambda)y \mid x \in X, y \in Y\}$.

Based on these two definitions, we have:

Theorem 3.3.1 *If* $\xi \in \mathbb{R}^m$ *is a random variable, the probability distribution of which is log-concave, then the probability distribution function* $F(x) = \mathbb{P}(\xi \leq x)$ *is a log-concave function in* $\mathbb{R}^m$.

The proof of Theorem 3.3.1 and the rationale of Definition 3.3.1 and 3.3.2 are presented in [62], and this dissertation omits them.

Suppose the capacity distribution is log-concave with a probability distribution function $F_\xi(x)$ (the log-concave assumption will be justified later). Then we have $1 - F_\xi(R_t x) \geq \alpha$, and by taking log of the both sides, Theorem 3.3.1 is applied to the probabilistic constraint of model 3.2 to get a convex function.

$$\mathbb{P}(R_t x \leq \xi_t) \geq \alpha \tag{3.6}$$

$$\iff \quad 1 - F_\xi(R_t x) \geq \alpha \tag{3.7}$$

$$\iff \quad log(1-\alpha) \geq log(F_\xi(R_t x)) \tag{3.8}$$

$$\iff \quad log(F_\xi(R_t x)) - log(1-\alpha) \leq 0 \tag{3.9}$$

Thus, the new model can be written as

$$\begin{aligned} \min \quad & c'x \\ s.t. \quad & \\ & g_t(x) = log(F_\xi(R_t x)) - log(1-\alpha) \leq 0 \quad t = 1, ..., T \\ & Ax \leq b \\ & x \in \mathbb{Z}_+ \end{aligned} \tag{3.10}$$

With the exception of chance constraint, $g_t(x)$, model described by Eq. (3.10) is a linear model. Although constraint $g_t(x)$ is nonlinear, it is convex by Theorem 3.3.1, which makes model described by Eq. (3.10) a convex program with respect to x. For

any feasible point x_0 of model described by Eq. (3.10), as long as we have the function value at x_0, i.e., $g(x_0)$ and its gradient $\bigtriangledown g(x_0)$ (subgradient if g(x) is nondifferentiable) at x_0, then a first-order gradient algorithm can be adopted to obtain the optimal solution [49].

The assumption that ξ follows a log-concave distribution, but lacks closed form distributional information, would be justified with two phases. First, in air traffic management, the historical data of ξ from the underlying distribution are mostly in the form of empirical distributions. When handled properly, the empirical distribution will be presented as a continuous distribution without the distributional information. By the Glivenko-Cantelli Theorem (see [51]), the empirical distribution function estimates the cumulative distribution function and converges with a probability of 1. That is, the empirical distribution can be presented as an underlying continuous distribution. Second, once the empirical distribution is in the format of a continuous distribution (but still lacks distribution information), this dissertation would further assume logconcavity because so many commonly used distributions are, indeed, log-concave. For example, the normal distribution, uniform distribution, gamma distribution (with a shape parameter greater than 1), beta distribution (with all parameters greater than 1), Weibull distribution, Laplace distribution, logistic distribution, exponential distribution and extreme value distribution are log-concave. There are only a few commonly used distributions that are not log-concave, such as the lognormal distribution, t-distribution, and Chi-square distribution, which are often used to describe the distributions of various statics rather than random variables raised from real problems.

In fact, most of the previous work on airspace capacity prediction has focused on the Airport Acceptance Rate (AAR), the number of arrivals an airport is capable of accepting each hour. As mentioned in the introduction, most studies focus on generating scenario tree of AAR forecasts from historical data [36,63]. For generating AAR distributions, the most common approach is the Weather Translation Model for Ground Delay Program Planning [50, 64]. Recently, a AAR Distribution Prediction

Model is proposed based on the Bayesian network model, which could predict the distribution based on the weather forecasts [65]. There is few literature on en route sector capacity distribution analysis, which is still a open area for future study.

With the log-concave distribution assumption, the formulation (3.10) is relaxed into a continuous problem. Then the formulation becomes a standard constrained optimization problem as follows:

$$\begin{aligned} \min \quad & f_0(x) = c'x \\ s.t. \quad & \\ & g_t(x) \leq 0, \quad t = 1, ..., T \\ & Ax \leq b \\ & x \geq 0 \end{aligned} \tag{3.11}$$

where x, A and b represent the vectors, in which t is included as one dimension.

I want to emphasize that all of the approximation method described in next section will be based on this standard model (3.11), and to be concise, all of the indices in this chapter are independent of the former ones in Chapter 2.

3.3.2 Details of the Approximation

The key to solving model described by Eq. (3.11) is to effectively evaluate the gradient (or subgradient) of $g_t(x)$, since the gradient could lead to the deepest feasible search direction. This subsection will build a polynomial-based approximation of $g(x)$ and use the gradient of the polynomial to approximate its original. Such an approximation has two advantages: first, thanks to the shape-preserving property of the Bernstein polynomial (see the definition in the Definition 3.3.3), we would effectively control the approximation errors for both the function values and their gradient at the same time. Second, we show that, under a large enough sample size, the obtained optimal solution will converge to the true optimal solution.

Problem setup

Suppose a feasible $x \in \mathbb{R}$ such that $Ax \leq b, x \geq 0$, $x = [x^1; \ldots; x^n]$ where $x^1, \ldots, x^n \in \mathbb{R}^1$. We would impose an upper bound and a lower bound on each component of x, as follows:

$$\ell^i \leq x^i \leq u^i, i = 1, \ldots, n. \tag{3.12}$$

We are interested in

$$\nabla g_t(x) := \left[\frac{\partial g_t(x)}{\partial x^1}; \ldots; \frac{\partial g_t(x)}{\partial x^n}\right] \tag{3.13}$$

and each component $\frac{\partial g_t(x)}{\partial x^i} : \mathbb{R} \to \mathbb{R}$ is a univariate function with respect to $x^i \in [\ell^i, u^i]$.

Let's define the i^{th} marginal function of $g_t(x)$ as $g_t^i(x^i)$, which is the unvariate function with respect to $x^i \in [\ell^i, u^i]$. $g_t^i(x^i)$ is essentially the function $g_t(x)$ with $x^1, ...x^{i-1}, x^{i+1}, ..., x^n$ as a constant value. In other words, the univariate function $g_t^i(x^i)$ is the orthogonal projection of $g_t(x)$ onto x^i. Since $g_t(x)$ is convex, all of its marginal functions $g_t^i(x^i), i = 1, ...n$ are convex with respect to x^i.

Our approach is to approximate all of the marginal functions of $g_t^i(x^i)$ with a convex, differentiable polynomial of degree k, $p_k(x^i)$ at a fixed x. Then, we estimate $\frac{\partial g_t(x)}{\partial x^i}$ by $p_k'(x^i)$, such that the problem of approximating $g_t(x)$ is decomposed into n independent univariate approximation problems.

Bernstein polynomial

In this dissertation, the Bernstein polynomial is adopted to construct the approximation $p_k(x^i)$. For the sake of simplifying the notation, we use $\phi(y)$ to represent one univariate function $g_t^i(x^i)$. Without a loss of generality, we assume $y \in [0, 1]$ because we can make a linear change of variable, if necessary, to transform any finite interval $[\ell^i, u^i]$ onto $[0, 1]$.

Definition 3.3.3 *The Bernstein polynomial of a function* $\phi(y), y \in [0,1]$ *is*

$$B_k(\phi; y) := \sum_{j=0}^{k} \binom{k}{j} y^j (1-y)^{k-j} \phi(j/k) \tag{3.14}$$

and

$$B_k(\phi; 0) = \phi(0), \ B_k(\phi; 1) = \phi(1). \tag{3.15}$$

Theorem 3.3.2 (Bernstein Theorem) *Let* $\phi(y)$ *be continuous on* $[0,1]$*. Then*

$$\lim_{k\to\infty} B_k(\phi; y) = \phi(y) \tag{3.16}$$

any point $y \in [0,1]$ *and the limit* (3.16) *hold uniformly in* $[0,1]$*. That is, given an* $\epsilon > 0$*, for all large enough* k*, we have*

$$|\phi(y) - B_k(\phi; y)| \leq \epsilon, y \in [0,1]. \tag{3.17}$$

The proof is in [66], and we omit it.

Theorem 3.3.3 *There exists a sequence of component functions:*

$$\psi_0(y), \psi_1(y), \psi_2(y), \ldots, \tag{3.18}$$

each is convex on $[0,1]$*, such that any function* $\phi(y)$ *that is convex on* $[0,1]$ *may be approximated with arbitary accuracy on* $[0,1]$ *by a sum of non-negative multiples of the component functions.*

The proof is in [67]. Since this result plays the central role of this dissertation, we will present the proof as a courtesy. We adopt our notations (not the original) as being consistent with our problem.

Proof First, we assume that $\phi(y)$ is twice differentiable on $[0,1]$ because if otherwise, we can apply Theorem 3.3.2 to construct a (convex) Bernstein polynomial, which approximates $\phi(y)$ to within $\frac{\epsilon}{2}$ on $[0,1]$ using a degree of $k > 2$. We then use the

obtained Bernstein polynomial to replace $\phi(y)$. We use $\phi'(y)$ and $\phi''(y)$ to denote the first- and second-order derivatives of $\phi(y)$, respectively. Let

$$B_k(\phi''; y) = \sum_{j=0}^{k} \binom{k}{j} y^j (1-y)^{k-j} \phi''(j/k) \tag{3.19}$$

represent the Bernstein polynomial of degree k for $\phi''(y)$. Let us observe that $y^j(1-y)^{k-j} \geq 0$ on $[0,1]$ and that in (3.19) are being approximated by the sum of non-negative multiples of the polynomials $y^j(1-y)^{k-j}$. For $k \geq 2$, define $p_k(y)$ by

$$p_k''(y) = B_{k-2}(\phi''; y),\ p_k'(0) = \phi'(0),\ p_k(0) = \phi(0). \tag{3.20}$$

We see that $p_k(y)$ is a polynomial of degree at most k. We also define $\beta_{j,k}(y)$ for $2 \leq j \leq k$, by

$$\beta_{j,k}''(y) = y^{j-2}(1-y)^{k-j},\ \beta_{j,k}'(0) = \beta_{j,k}(0) = 0. \tag{3.21}$$

To complete the definition of polynomials $\beta_{j,k}(y)$, we define

$$\beta_{0,k}(y) = \text{sign}[\phi(0)],\ \beta_{1,k}(y) = y\text{sign}[\phi'(0)]. \tag{3.22}$$

The relevance of the choice of functions (3.22) will be seen later. We then have

$$p_k(y) = \sum_{j=0}^{k} c_j^* \beta_{j,k}(y), \tag{3.23}$$

where $c_j^* \geq 0$ and $\beta_{j,k}''(y) \geq 0$ on $[0,1]$. Now, given any $\epsilon > 0$, applying Theorem 3.3.2, we have

$$|B_{k-2}(\phi''; y) - \phi''(y)| \leq \epsilon \tag{3.24}$$

on $[0,1]$. That is

$$|p_k''(y) - \phi''(y)| \leq \epsilon \tag{3.25}$$

on $[0,1]$ and therefore, for $y \in [0,1]$,

$$\int_0^y (p_k''(t) - \phi''(t))dt \ \leq \int_0^y |p_k''(t) - \phi''(t)|dt \leq \epsilon y \leq \epsilon. \tag{3.26}$$

Using (3.20), the inequality (3.26) gives

$$|p_k'(y) - \phi'(y)| \leq \epsilon \tag{3.27}$$

for $y \in [0,1]$. Similarly, another integration shows that

$$|p_k(y) - \phi(y)| \leq \epsilon \tag{3.28}$$

for $y \in [0,1]$. Note that the polynomial $\beta_{j,k}(y)$ may be $\psi_0(y), \psi_1(y), \psi_2(y), \ldots$. We set

$$\psi_j(y) = \beta_{j,k}(y),\ 2 \leq j \leq k,\ \psi_0(y) = \text{sign}[\phi(0)],\ \psi_1(y) = y\text{sign}[\phi'(0)] \tag{3.29}$$

where for $j \geq 2$,

$$\beta''_{j,k}(y) = y^{j-2}(1-y)^{k-j} = y^{j-2}\sum_{i=1}^{k-j}(-1)^i\binom{k-j}{i}y^i \tag{3.30}$$

and we have

$$\beta_{j,k}(y) = y^j\sum_{i=0}^{k-j}(-1)^i\frac{\binom{k-j}{i}y^i}{[(i+j)(i+j-1)]}. \tag{3.31}$$

■

Theorem 3.3.2 shows that the Bernstein polynomial can approximate any continuous univariate function on a closed interval. However, for a convex function $\phi(y)$, its Bernstein polynomial approximation may not be convex because the sampled data may not actually be convex due to experimental numerical error. Besides, for the simple Bernstein polynomial, the degree of the polynomial need be doubled to halve the error [67]. Thus, we discard the idea of directly approximating $g_t^i(x^i)$ by the Bernstein polynomial. Instead, Theorem 3.3.3 shows that for any convex function $\phi(y), y \in [0,1]$, we can always approximate both $\phi(y)$ and its derivative $\phi'(y)$ within ϵ uniformly with the polynomial

$$p_k(y) = \sum_{j=0}^{k} c_j^*\psi_j(y) \tag{3.32}$$

of degree k, regardless of the differentiability of $\phi(y)$. As long as all of the coefficients in (3.32) are non-negative, $c_j^* \geq 0, j = 0, 1, \ldots, k$, $p_k(y)$ will be convex. We also need $k+1$ non-negative coefficients $c_0^*, \ldots, c_k^*$ to construct $p_k(y)$. To obtain these

coefficients, we need a set of points with coordinates $(y_i, \phi(y_i)), i = 1, 2, \ldots, k+1$. We solve the following problem

$$\min \left\{ \max_{i=1,\ldots,k+1} |\phi(y_i) - \sum_{j=0}^{k} c_j^* \psi_j(y_i)|, c_j^* \geq 0 \right\} \tag{3.33}$$

the $p_k(y)$ with coefficients $c_0^*, \ldots, c_k^*$ is called the best approximation of degree k.

Choose proper degree of Bernstein polynomial

We now need to determine the proper choice of the degree k. The following theorem addresses this issue.

Theorem 3.3.4 (Jackson's Theorem V) *If $\phi(y)$ is r-differentiable on $y \in [0,1]$, and $\phi(y)$ is approximated by $p_k(y)$, then the approximation error of $\phi(y)$ on $[0,1]$ satisfies:*

$$\max_{y\in[0,1]} |\phi(y) - p_k(y)| \leq \left(\frac{\pi}{2}\right)^r \frac{|\phi^{(r)}(\omega)|}{[(k-r+2)\ldots k(k+1)]}, k \geq r \tag{3.34}$$

where $\phi^{(r)}(\omega)$ represents the r-order derivative of $\phi(y)$ at some $\omega \in [0,1]$.

The proof of this theorem is in [68]. Theorem 3.3.4 shows that if we approximate an $r-$differentiable function by $p_k(y)$, the error will be quickly reduced by increasing the order of the polynomial. For example, when the degree increases from k to $k+1$, the rate of the error-bound reduction will be

$$\frac{\left(\frac{\pi}{2}\right)^r \frac{|\phi^{(r)}(\omega)|}{[(k-r+3)\ldots(k+1)(k+2)]}}{\left(\frac{\pi}{2}\right)^r \frac{|\phi^{(r)}(\omega)|}{[(k-r+2)\ldots k(k+1)]}} = \frac{k-r+2}{k+2} < 1. \tag{3.35}$$

In the previous discussion, we assume that $\phi(y)$ is twice-differentiable, i.e., $r = 2$. If we evaluate the error bound when $k = 4$, and $\frac{\pi^2}{2}\frac{|\phi^{(2)}(\omega)|}{4\times 5}$ is the scale of the error base valued at 1, we present the results in the following Table 3.3.

From Table 3.3, increasing k from 4 to 5 will reduce the error bound to 0.67 of its original value, while increasing k from 4 to 20 will reduce the error bound to 0.048

Table 3.3. Error bounds as the degree k of $p_k(y)$ increases

degree (k)	Error Bound
4	1
5	0.67
6	0.48
7	0.36
8	0.28
9	0.22
10	0.18
12	0.13
20	0.048
40	0.012
50	0.008

of its original value. When we increase k from 4 to 50, the new error bound will be reduced to 0.008 of its original value. Given the result of Theorem 3.3.3 and the good performance of the "best approximation," i.e., $p_k(y)$, the error bound when $k = 4$ would already be a well-bounded value. Thus, when we use $k = 50$, the new error bound will be reduced to a fraction of 0.008, which should serve us adequately well.

Determine the approximation points

At last, we determine the $k + 1$ coordinates, i.e., $(y_i, \phi(y_i)), i = 1, \ldots, k + 1$ to construct $p_k(y)$.

Proposition 3.3.1 *Let $p_k(y)$ be the polynomial constructed from $k + 1$ coordinates $(y_i, \phi(y_i)), i = 1, \ldots, k + 1$, Then,*

$$\phi(y) - p_k(y) \ = \frac{\phi^{(k+1)}(\omega)}{(k+1)!} \Pi_{i=1}^{k+1}(y - y_i) \tag{3.36}$$

where ω lies in the smallest interval containing $y_1, \ldots, y_{k+1}$ and y.

This proposition is in [69, Lecture 20]. Since we can apply Theorem 3.3.2 to approximate any continuous function by a Bernstein polynomial, which is differentiable, we can assume that the $(k+1)$-order derivative $\phi^{(k+1)}(y)$ exists, and it is a bounded value for $y \in [0,1]$. Thus, in order to reduce the error of approximation, we need to minimize $\Pi_{i=1}^{(k+1)}(y - y_i)$ by choosing the *Chebyshev nodes* on $[0,1]$, as follows:

$$y_i := \frac{1}{2} - \frac{1}{2}\cos\left(\frac{2i-1}{2k+2}\pi\right),\ i = 1, \ldots, k+1. \tag{3.37}$$

The technical detail regarding the minimization of the approximation error by adopting Chebyshev nodes is in [69].

Steps for convex approximation

The procedure to estimate $\phi'(y)$ by the polynomial $p_k(y)$ is summarized in the following steps:

Step 0. The polynomial $p_k(y)$ is defined by the formulation in Theorem 3.3.3.

Step 1. Determine the overall error bound $\epsilon > 0$.

Step 2. Choose the degree k based on Theorem 3.3.4.

Step 3. Calculate k+1 Chebyshev nodes y_i and coordinates $(y_i, \phi(y_i))$,i=1,.....k+1 (Proposition 3.3.1).

Step 4. Solve the model (3.33) for the coefficients c_0^*,...,c_k^* and construct $p_k(y)$ in Eq. (3.32).

Step 5. Use $p_k'(y)$ as an approximation of $\phi'(y)$.

3.4 Algorithms

Based on the approximation approach in Section 3.2, the function values and gradients for the chance constraint $g_t(x)$ can be estimated for any given point $\bar{x}$. Therefore a first-order algorithm can be adopted to solve the whole problem. In this paper, the gradient mapping method in [49] is adopted as the primary algorithm for

two reasons: first, this method terminates within a polynomial number of iterations; second, only the first-order information (i.e., function value and gradient) is required.

To perform the gradient mapping method, the formulation (3.6) can be transformed into the parametric max-type function:

$$f(d;x) = max\{f_0(x) - d; g_t(x), t \in \mathbb{T}\}, d \in \mathbb{R}^1, x \in Q. \tag{3.38}$$

where the functions g_t are convex and smooth, and Q is a closed convex set defined by $Ax \leq b$ and $x \geq 0$. Moreover, a linearization of a parametric max-type function f(d;x) is shown as:

$$f(d;\bar{x};x) = \max_{t\in\mathbb{T}}\{f_0(\bar{x}) + \langle f_0'(\bar{x}), x - \bar{x}\rangle - d; g_t(\bar{x}) + \langle g_t'(\bar{x}), x - \bar{x}\rangle\}. \tag{3.39}$$

To introduce a gradient mapping in a standard way, let us fix some $\gamma > 0$, denoted by:

$$f_\gamma(d;\bar{x};x) = f(d;\bar{x};x) + \frac{\gamma}{2}\|x - \bar{x}\|^2 \tag{3.40}$$

$$f^*(d;\bar{x};\gamma) = \min_{x\in Q} f_\gamma(d;\bar{x};x) \tag{3.41}$$

$$x_f(d;\bar{x};\gamma) = arg\min_{x\in Q} f_\gamma(d;\bar{x};x) \tag{3.42}$$

$$g_f(d;\bar{x};\gamma) = \gamma(\bar{x} - x_f(d;\bar{x};\gamma)). \tag{3.43}$$

where $g_f(d;\bar{x};\gamma)$ is the constrained gradient mapping of the problem (3.6).

The main algorithm for the chance-constrained problem is shown in Algorithm 3.1.

3.5 Computational Complexity

The computational complexity of the polynomial approximation approach for the chance-constrained problem is analyzed as follows. First, the computational complexity of evaluating $g_t(x)$ and $\bigtriangledown g_t(x)$ at a given $\bar{x}$ is demonstrated. Second, we show the overall complexity with the gradient mapping as the main algorithm. Note that the arithmetic operations count is a measure of the computational complexity, which ignores the fact that adding or multiplying large integers or a high-precision floating

input : Standard chance constrained model
output: Optimal solution

1: Initialization: Choose $x_0 \in Q$, $\kappa = 0.25$, $L = 10$, $d_0 = 1$ and accuracy $\epsilon > 0$.

2: rth iteration ($r \geq 0$).

a) Set $x_{r,0} = x_r$, $y_{r,0} = x_r$ and $\alpha_0^* = 0.5$

for the jth internal iteration:

Approximate $g_t(y_{r,j})$ and $\nabla g_t(y_{r,j})$ by the method in Section 3.3

Compute $f(d_r; y_{r,j})$ and $f'(d_r; y_{r,j})$.

Set $x_{r,j+1} = x_f(d_r; y_{r,j}; L)$

Solve $\alpha_{j+1}^* \in (0,1)$ from equation: ${\alpha_{j+1}^*}^2 = (1 - \alpha_{j+1}^*)\alpha_j^{*2}$

Set $\beta_j^* = \frac{\alpha_j^*(1-\alpha_j^*)}{\alpha_j^{*2}+\alpha_{j+1}^*}$ and $y_{r,j+1} = x_{r,j} + \beta_j^*(x_{r,j+1})$

If $f^*(d_r; x_{r,j}; 0) \geq (1-\kappa) f^*(d_r; x_{r,j}; L)$

then stop the internal process and set j(r)=j.

Set $j^*(r) = arg \min_{0 \leq j \geq j(r)} (f^*(d_r; x_{r,j}; L)$ and $x_{r+1} = x_f(d_r; y_{r,j^*(r)}; L)$.

Global stop: If at some iteration of the internal scheme we have $f^*(d_r; x_{r,j}; L) \leq \epsilon$

b) update d_r: $d_{r+1} = d^*(x_{r,j(r)}, d_r)$, where $d^*(\bar{x}, d)$ is the root in d of function $f^*(d; \bar{x}; 0)$

r=r+1

Algorithm 3.1. Chance-constrained optimization based on convex approximation

point number is more demanding than adding or multiplying single-digit integers. In other words, this paper charges the uniform cost for each computational operation.

First of all, we need to calculate the Chebyshev nodes and evaluate the $\phi(y_i)$. The cost of calculating each coordinate $(y_i, \phi(y_i))$ is a constant value, denoted as P. To construct each $g_t^i(x^i)$, we need k+1 coordinates, which takes $(k+1)P$. The construction of model (3.33) needs to calculate $\psi_0(x^i), \ldots, \psi_k(x^i)$. Since these terms are simple polynomials and each one of their calculations only takes up to $O(k)$ arithmetic operations, the total cost of calculating $\psi_0(x^i), \ldots, \psi_k(x^i)$ will take $k+1$

times of $O(k)$ for each item (there are k+1 items for a simple polynomial of degree k). Thus, it takes up to $(k+1)^2O(k)+(k+1)P$ to construct model(3.33).

Model (3.33) is a convex optimization problem with k+1 variables, and its computational complexity is $O((k+1)^3)$ in the worst-case scenario, according to [70]. With the obtained $c_0,\ldots,c_k$, it will take $O(k)$ to calculate the value of $g_t(x)$ and $\nabla g_t(x)$. Therefore, it takes

$$n\{O((k+1)^3)+O(k)+(k+1)^2O(k)+(k+1)P\}$$

arithmetic operations to obtain the approximate values of $g_t(x)$ and $\nabla g_t(x)$.

By adopting the gradient mapping method in [49], we can get the following result:

Proposition 3.5.1 *The gradient mapping method takes at most*

$$\frac{1}{\ln(2(1-\kappa))}\ln\frac{d_0-d*}{(1-\kappa)\epsilon} \tag{3.44}$$

iterations to obtain an ϵ-optimal solution, where κ is a constant (e.g., $\kappa=0.25$) and d_0, d^* are the progressively updated penalty coefficients.

The proof is in [49]. In the proof, both κ and d_0-d^* are well-bounded values. Therefore, the iteration value of (3.44) will be bounded, as well.

Therefore, the overall number of arithmetic operations toward an ϵ-optimal solution will be

$$n\{O((k+1)^3)+O(k)+(k+1)^2O(k)+(k+1)P\}\frac{1}{\ln(2(1-\kappa))}\ln\frac{d_0-d*}{(1-\kappa)\epsilon}$$

when the gradient mapping algorithm is used.

4. TRAFFIC FLOW MANAGEMENT WITH CHANCE CONSTRAINT

4.1 Introduction

In order to efficiently balance traffic demand and capacity, optimization of ATFM relies on accurate predictions of future capacity states. However, these predictions are inherently uncertain due to factors, such as weather. The traditional methods often formulate this stochastic ATFM problem as a MILP model based on the scenario tree method. However, this MILP model often needs to enumerate all possible capacity combinations under different predefined scenarios. Thus the exponentially increased computational complexity prevents it from being applicable to large-scale problems in reality.

This chapter presents a computationally efficient method to address uncertainty in ATFM by using the chance-constrained optimization method, which is introduced in Chapter 3. First, a chance-constrained model is developed based on a previous deterministic Integer Programming optimization model of ATFM to include probabilistic sector capacity constraints. Then, to efficiently solve such a large-scale chance-constrained optimization problem, the convex approximation-based approach in Chapter 3 is applied. The approximation is based on the numerical properties of the Bernstein polynomial, which is capable of effectively controlling the approximation error for both the function value and gradient. Thus, a first-order algorithm can be adopted to obtain a satisfactory solution, which is expected to be optimal. To validate the proposed convex approximation approach, numerical results are reported by comparing it with the brute-force method.

4.2 Deterministic Aggregate Traffic Flow Management Modeling

The proposed stochastic Traffic Flow Management model is derived based on a previous deterministic Link Transmission Model (LTM) for ATFM [22]. The LTM is a data-driven model. It establishes a route network based on radar tracks extracted from Aircraft Situation Distributed to Industry (ASDI) data compiled by the ETMS [71, 72]. As shown in Figure 4.1, a sector is a basic airspace session that is monitored by one or more air traffic controllers. Each flight path is a sequence of *links* that connects a departure airport and an arrival airport, with each link being an abstraction of a passage through a sector. The travel time of a link is extracted from historical flight data. There are thousands of aircraft traveling on their paths throughout the day, forming a multi-commodity network across the NAS [21].

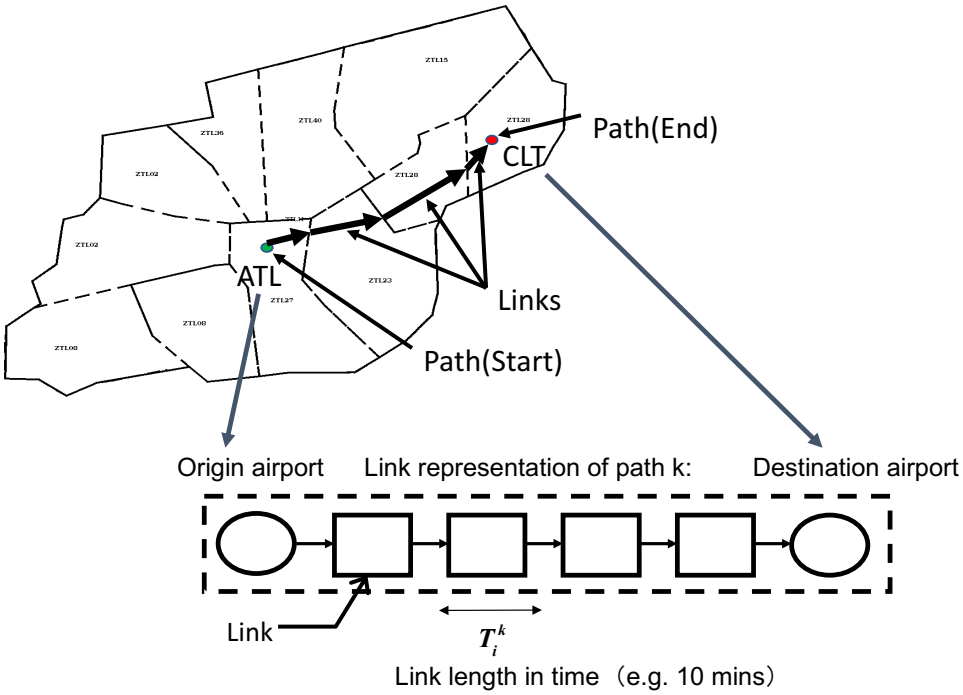


Figure 4.1. Link transmission model.

The LTM is modeled as a discrete-time linear system, where the state variable $x_i^k(t)$ is defined as the aircraft count in link i on route k at time t, and $q_i^k(t)$ is

the outflow of this link. The dynamics of the traffic flow are governed by the flow conservation (for the first link, its upstream outflow is departure $f^k(t)$.):

$$x_i^k(t+1) \;=\; x_i^k(t) - q_i^k(t) + q_{i-1}^k(t) \qquad \forall \quad i \in \{0, \cdots, n^k\}, k \in \mathbb{K}, t \in \mathbb{T}$$

A typical deterministic TFM problem is formulated as an IP problem. By controlling the flow rate $q_i^k(t)$, delays are minimized, while sector counts are kept below the sector capacity $C_s(t)$, and departure and arrival volume is also constrained under departure capacity $C_{dep}(t)$ and arrival capacity $C_{arr}(t)$.

$$\min \qquad d = \sum_{t\in\mathbb{T}} \sum_{k\in\mathbb{K}} \sum_{1\le i\le n^k} c_i^k x_i^k(t) \tag{4.1}$$

$s.t.$

$$x_i^k(t+1) = x_i^k(t) - q_i^k(t) + q_{i-1}^k(t) \tag{4.2}$$

$$\sum_{(i,k)\in Q_{s_i}} x_i^k(t) \le C_s(t), \sum_{(0,k)\in A_{arr}} q_0^k(t) \le C_{arr}(t), \sum_{(n^k,k)\in A_{dep}} q_{n^k}^k(t) \le C_{dep}(t) \tag{4.3}$$

$$\sum_{t\in\mathbb{T}} q_0^k(t) = \sum_{t\in\mathbb{T}} q_{n^k}^k(t) = \sum_{t\in\mathbb{T}} f^k(t) \tag{4.4}$$

$$\sum_{t=T_0^k+T_1^k\cdots+T_i^k}^{T_*^k} q_i^k(t) \le \sum_{t=T_0^k+T_1^k\cdots+T_{i-1}^k}^{T_*^k-T_i^k} q_{i-1}^k(t) \tag{4.5}$$

$$\sum_{t=0}^{T_0^k+T_1^k\cdots+T_i^k-1} q_i^k(t) = 0, \qquad x_i^k(0) = 0 \tag{4.6}$$

$$x_i^k(t) \in \mathbb{Z}_+, \qquad q_i^k(t) \in \mathbb{Z}_+ \tag{4.7}$$

$$\forall \quad T_*^k \ge T_0^k + T_1^k \cdots + T_i^k, \quad i \in \{0, \cdots, n^k\}, \quad k \in \mathbb{K}, \quad t \in \mathbb{T}, \quad s \in \mathbb{S}$$

The objective d is to minimize the weighted total flight time of all flights in the planning time horizon, which reflects the realistic goal to minimize delays. Constraints (4.1)-(4.7) regulate traffic flow behaviors. Constraints (4.3) enforce en route and airport capacity constraints, where the link n^k (0) is defined as a special link represents the destination (origin) airport. Constraints (4.4) ensure that the accumulated departures equal to the accumulated arrivals. Constraints (4.5) show that every

flight must dwell in a link for at least T_i^k minutes. Constraints (4.6) and (4.7) represent the initial states and integer constraints, respectively. The detailed discussions about these constraints is shown in Reference [22].

The solution to the above TFM problem is the optimal traffic flow, as well as the flow control for each route. Specifically, vector $[x_1^k(t), x_2^k(t), \cdots, x_{n^k}^k(t)]$ represents the state of route k at time t. As t evolves, the states of the vector represent the movement of the traffic flow. Given that traffic control is generally applied to individual aircraft rather than a flow, the flow control obtained from this model seems impracticable. However, a disaggregation process can convert the flow control into flight-specific actions. The idea is that these optimal states, i.e., vector $[x_1^k(t), x_2^k(t), \cdots, x_{n^k}^k(t)]$, can be used as constraints for scheduling the flights on route k, where variables are defined as ground delays and airborne delays associated with individual flights. The disaggregation process is discussed in detail in Reference [73]. After the disaggregation process, the flow controls are translated into delays imposed on individual flights in each sector.

4.3 Chance Constraints

The probabilistic TFM model aims to incorporate the constantly changing airspace capacities, which are caused by adverse weather conditions, into the TFM optimization. The current TFM model is rather deterministic, i.e., considering the stochastic airspace capacities $C_s(t)$, $C_{arr}(t)$ and $C_{dep}(t)$ as deterministic values. This chapter proposes to impose a probabilistic constraint on traffic flow capacities, as follows:

$$\mathbb{P}\left(\begin{array}{c} \sum_{(i,k)\in Q_{s_i}} x_i^k(t) \le C_s(t), \quad \forall s_i \in Q_{s_i} \\ \sum_{(0,k)\in A_{arr}} q_0^k(t) \le C_{arr}(t), \quad \forall s_i \in A_{arr} \\ \sum_{(n^k,k)\in A_{dep}} q_{n^k}^k(t) \le C_{dep}(t), \quad \forall s_i \in A_{dep} \end{array}\right) \ge \alpha, t \in \mathbb{T} \tag{4.8}$$

where $\mathbb{P}(\cdot)$ is the probability measure for the stochastic airspace capacities, meaning that the sector capacity will only raise a feasibility issue with the probability of

$\alpha \in (0, 1)$. Similarly, α is the service level in this problem, which is defined in Chapter 2.

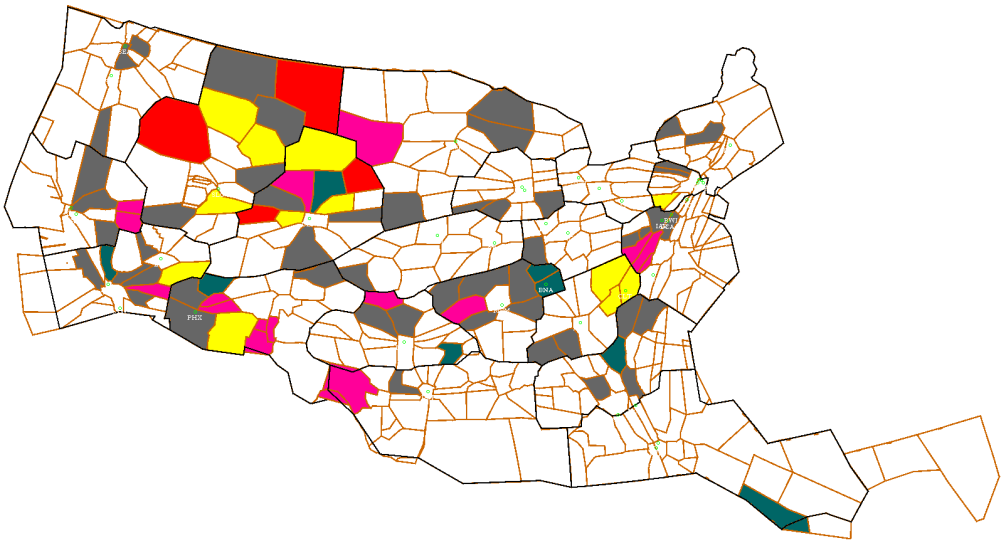


Figure 4.2. Sector capacity affected by adverse weather on 2013/04/10, where the color represent the reduction of capacity. The red one has high reduction.

All random components $C_s(t)$, $C_{arr}(t)$ and $C_{dep}(t)$ are random vectors that represent the correlated, stochastic airspace capacities, and only correlated random capacities are meaningful for the TFM optimization because adverse weather conditions will usually affect multiple sectors at the same time, as it shown in Figure 4.2. Since the constraints (4.3) are all linear, the probabilistic constraint (4.8) can be simply written as:

$$\mathbb{P}(T(t)x(t) \leq \xi(t)) \geq \alpha, t \in \mathbb{T} \tag{4.9}$$

where $x(t)$ denotes the vector of the decision variables with the associated coefficient matrix $T(t)$ at time t, and $\xi(t)$ is representing the random vectors at time t, which follows a joint distribution.

Thus, the TFM optimization under the stochastic airspace capacities can be written as:

$$\min \quad d = \sum_{t\in\mathbb{T}} \sum_{k\in\mathbb{K}} \sum_{1\le i\le n^k} c_i^k x_i^k(t) \tag{4.10}$$

$s.t.$

$$x_i^k(t+1) = x_i^k(t) - q_i^k(t) + q_{i-1}^k(t) \tag{4.11}$$

$$\mathbb{P}(T(t)x(t) \le \xi(t)) \ge \alpha \tag{4.12}$$

$$\sum_{t\in\mathbb{T}} q_0^k(t) = \sum_{t\in\mathbb{T}} q_{n^k}^k(t) = \sum_{t\in\mathbb{T}} f^k(t) \tag{4.13}$$

$$\sum_{t=T_0^k+T_1^k\cdots+T_i^k}^{T_*^k} q_i^k(t) \le \sum_{t=T_0^k+T_1^k\cdots+T_{i-1}^k}^{T_*^k-T_i^k} q_{i-1}^k(t) \tag{4.14}$$

$$\sum_{t=0}^{T_0^k+T_1^k\cdots+T_i^k-1} q_i^k(t) = 0, \qquad x_i^k(0) = 0 \tag{4.15}$$

$$x_i^k(t) \in \mathbb{Z}_+, \qquad q_i^k(t) \in \mathbb{Z}_+ \tag{4.16}$$

$$\forall \quad T_*^k \ge T_0^k + T_1^k \cdots + T_i^k, \quad i \in \{0, \cdots, n^k\}, \quad k \in \mathbb{K}, \quad t \in \mathbb{T}, \quad s \in \mathbb{S}$$

The only difference from the deterministic model is that the capacity constraints (4.3) are replaced with the probabilistic capacity constraint (4.8). This problem is referred as *chance-constrained TFM optimization*.

Moreover, notice that the above TFM model is a linear model, except for the probabilistic constraint (4.8). Thus, the stochastic TFM problem can be written in the standard chance constrained form as following:

$$\begin{aligned} \min \quad & c'x \\ s.t. \quad & \\ & g_t(x) = \mathbb{P}(T(t)x(t) \le \xi(t)) \ge \alpha, t \in \mathbb{T} \qquad (4.17) \\ & Ax \le b \\ & x \in \mathbb{Z}_+ \end{aligned}$$

where c represents the vector of the weight coefficients; A and b are the coefficients vectors corresponding to the original linear constraints, in which t is included as one dimension.

Therefore, the solving methods that are introduced in Chapter 3, such as the brute force and convex approximation method, can be easily adopted based on this standard chance constrained formulation.

4.4 Model Validation with a Small-Sized Example

4.4.1 Example Setup

As stated in Chapter 3, the brute-force MILP method has a limitation in handling large-scale real problems, but it could provide accurate results for small-sized problems. Therefore, the MILP method could be an ideal contrast to the approximation-based method if a small-sized TFM problem could be provided.

In order to evaluate the accuracy of the novel approximation-based method, a small-sized TFM problem is developed to perform the comparison. As shown in Figure 4.3, the designed small TFM problem consists of five sectors (ZOB29, ZOB47, ZOB49, ZOB79, ZOB26 at the Cleveland Air Route Traffic Control Center) and four airports (denoted as $k_1 : DTW, k_2 : TOL, k_3 : CLE, k_4 : ERI$). The flight plan is shown in Table 4.1, which contains six flight routes with the corresponding departure schedule for each time step ($f_k(t)$). Each flight is assumed to be able to traverse each sector in one time period, and there are 11 planning time periods in total (note that these are abstract periods and could be defined by real time periods, such as 15 min, in a full-scale problem). For simplification, the capacity of each sector is assumed to be the same and set at a maximum of 4. All sectors are assumed to be independent and subject to the same probability distribution, given in Table 4.2. The corresponding cumulative probability ($P(C \geq sn_j)$) is shown in Table 4.3, which represents the probability of satisfying the sector's capacity limit when assigning sn_j flights to that sector.

Note that if the sectors are not independent, then only the calculations of the joint probability distribution need to be changed, and the method to form the feasible combination set of the MILP model is the same.

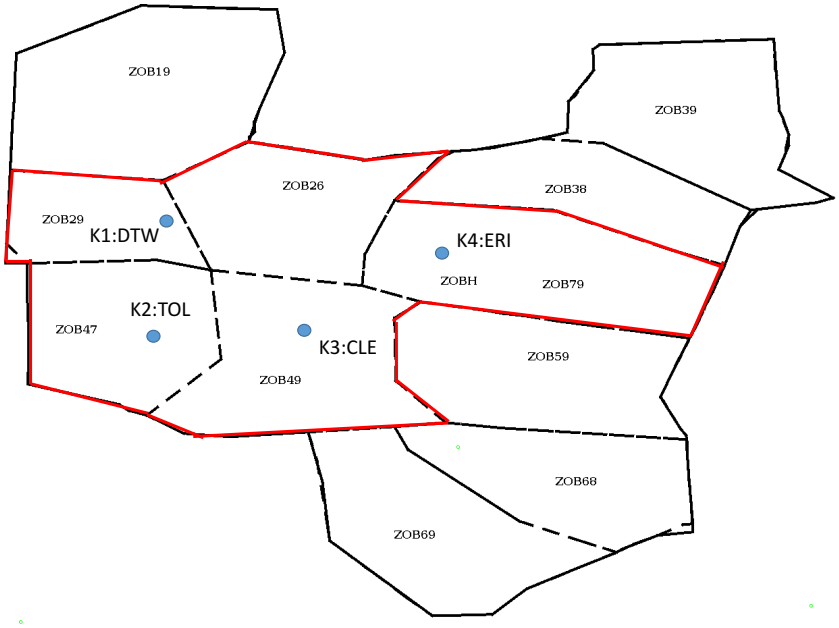


Figure 4.3. Small-sized example.

4.4.2 Result of MILP

Since there are five sectors and each sector has four possible capacity values (0 and 1 have the same probability). Therefore, for each time step, we have $4^5 = 1024$ possible capacity combinations and each of them needs one associated binary variable, which represents the status of activation for each combination. For such a small case with ten discrete time intervals, at least $1024 * 10 = 10240$ binary variables are needed, which indicates the limitation of the MILP method for any realistic large scale problems.

The MILP model is solved with the chance constraint under service level of 0.8 ($\alpha = 0.8$). The MILP is implemented in the Python programming language and

Table 4.1. Flight schedule

Flights	$f_k(t)$	Origin	Destination	Route
f_1	[2,1,1,0,0,0,0,0,0,0,0]	DTW	CLE	$KDTW$-$ZOB29$-$ZOB47$-$ZOB49$-$KCLE$
f_2	[2,1,1,0,0,0,0,0,0,0,0]	CLE	TOL	$KCLE$-$ZOB49$-$ZOB47$-$KTOL$
f_3	[2,1,1,0,0,0,0,0,0,0,0]	TOL	ERI	$KTOL$-$ZOB47$-$ZOB26$-$ZOB79$-$KERI$
f_4	[2,1,1,0,0,0,0,0,0,0,0]	ERI	DTW	$KERI$-$ZOB79$-$ZOB26$-$ZOB29$-$KDTW$
f_5	[2,1,1,0,0,0,0,0,0,0,0]	DTW	TOL	$KDTW$-$ZOB29$-$ZOB47$-$KTOL$
f_6	[2,1,1,0,0,0,0,0,0,0,0]	CLE	ERI	$KCLE$-$ZOB49$-$ZOB79$-$KERI$

Table 4.2. Sector capacity distribution

Capacity(C)	1	2	3	4
$P(C)$	0.0321	0.0871	0.2369	0.6439

Table 4.3. Cumulative probability : $P(C \geq sn_j)$

number of flights(sn_j)	0	1	2	3	4	≥ 5
$P(C \geq sn_j)$	1.0	1.0	0.9679	0.8808	0.6439	0

the solutions are found with the Gurobi mathematical programming solver [48]. The result is then collected to check the feasibility of the chance constraint, which is shown in Table 4.4. It is clear to see that the chance constraints are all satisfied at each time step t. The original objective is 118 based on the MILP method, which is the accurate optimal integer solution. Later, we will use this accurate optimal solution to evaluate the approximation-based method.

4.4.3 Result of Approximation-based Approach

To perform the convex approximation-based approach, a log-concave continuous probability distribution is provided, as shown in Figure 4.4, which has the exact cumulative probability ($P(C \geq sn_j)$) with the discrete one in Table 4.2. Therefore, the comparison is meaningful with the same probability information. With the same service level of 0.8 ($\alpha = 0.8$), the result, based on the convex approximation-based method in Chapter 3, is shown in Table 4.5, which are continuous real numbers before the rounding process. It is clear that the convex approximation-based method could provide feasible optimal solutions, which also satisfy all of the chance constraints for each time step t. The original objective is reduced to 115.8 because the feasible set

Table 4.4. Optimal flight flow based on MILP

Time	sn_1	sn_2	sn_3	sn_4	sn_5	$P \geq 0.8$
0	0	0	0	0	0	1
1	0	0	0	0	0	1
2	3	2	2	0	0	0.825
3	1	3	1	2	2	0.825
4	2	2	1	0	3	0.825
5	1	3	2	1	2	0.825
6	2	2	2	2	2	0.849
7	1	3	2	1	2	0.825
8	1	1	1	2	0	0.9679
9	1	0	1	0	1	1
10	0	0	0	0	0	1
obj=118						

of the integer problem is only a subset of the real-valued problem. Thus, this smaller optimal objective is reasonable and could be evidence to demonstrate that the convex approximation-based approach could achieve a valid real-valued optimal solution.

To achieve the integer-valued solution for the original problem, the Branch-and-Bound (B&B) algorithm is performed with the Integer Programming solver Gurobi 6.02 [48], where the real-valued optimal solution is provided as an initial point. As shown in Table 4.6, the rounding process provides a feasible integer-valued solution, which is sub-optimal, but very close to the accurate optimal integer solution. There are two reasons for the sub-optimal solution: first, there are errors in the approximation process, since we only choose the polynomial with a finite degree to approximate the original chance constraints; second, only a simple B&B cutting process is performed to achieve the integer result. Although it is sub-optimal, the error between the

Table 4.5. Optimal flight flow based on the approximation method in real value

Time	sn_1	sn_2	sn_3	sn_4	sn_5	$P \geq 0.8$
0	0	0	0	0	0	1
1	0	0	0	0	0	1
2	2.156	1.165	2.229	1.648	1.166	0.894
3	2.283	2.562	1.000	2.331	2.194	0.803
4	2.061	2.574	1.551	1.370	2.236	0.836
5	1.874	2.228	2.226	0.966	2.015	0.859
6	1.667	2.033	1.799	0.851	1.836	0.905
7	1.959	1.773	1.781	0.833	1.719	0.910
8	0.000	2.108	1.552	0.000	0.883	0.949
9	0.000	1.552	0.000	0.000	0.000	0.986
10	0.000	0.000	0.000	0.000	0.000	1
obj=115.8						

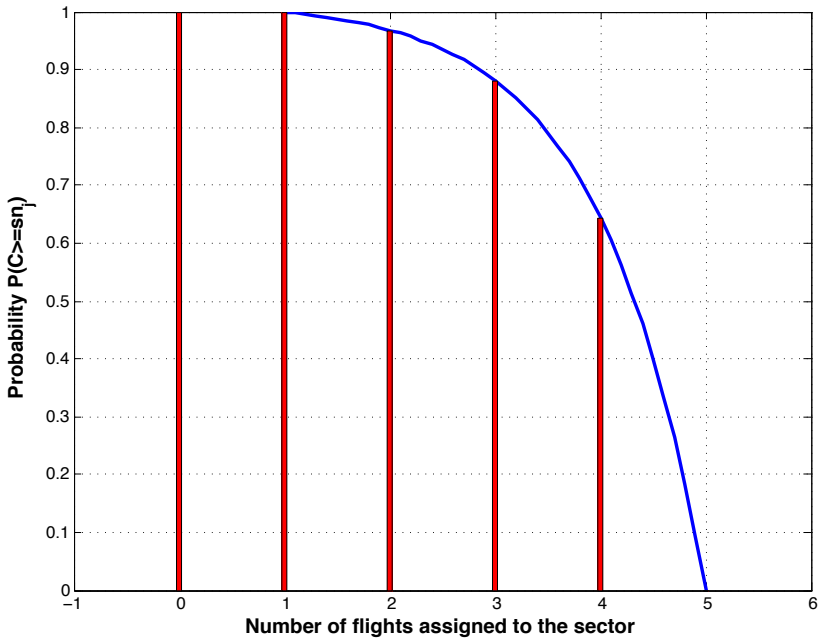


Figure 4.4. Cumulative probability function

two objectives is only 3%. Thus, it is reasonable to believe that the approximation-based approach could provide a reliable integer solution to the chance-constrained TFM problem.

Two comments should be added for this small sized experiment. First, the error gap between the approximation based approach and the optimal integer solution can be adjusted by the degree of the Bernstein polynomial. As it is demonstrated in Chapter 3, higher degree of the Bernstein polynomial could help decrease the approximation error gap. For example, the degree of 10 is chosen for this experiment. Increasing the degree to 20 or 30 can improve the accurate. However, the improvement is very limited since the current error gap is already around 3% and higher degree may face computational issues for the approximation step, which will be discussed in details later in Chapter 5. Second, the service level in this experiment can be easily evaluated by multiplying all the independent probability for each sector. If the sectors follow a joint capacity distribution, then the Monte Carlo evaluation

Table 4.6. Optimal flight flow based on the approximation method in integer value

Time	sn_1	sn_2	sn_3	sn_4	sn_5	$P \geq 0.8$
0	0	0	0	0	0	1
1	0	0	0	0	0	1
2	2	1	3	0	1	0.853
3	2	3	0	2	2	0.800
4	3	2	2	0	2	0.800
5	2	3	2	1	2	0.800
6	0	2	2	2	2	0.878
7	2	2	2	0	0	0.907
8	0	2	1	1	0	0.968
9	0	1	0	2	1	0.968
10	1	0	0	0	2	0.968
obj=121						

method that is introduced in Chapter 2 should be used because it is often much faster to get the probability from the Monte Carlo simulation rather than the numerical integration over multi-variable joint probability density function. Moreover, the close form probability density function may be not available at all.

4.5 Large Scale Experiment

This section presents a large-scale ATFM optimization, employing the proposed chance constrained model. The traffic data are extracted from the ASDI, which provides historical traffic data, as well as flight plans. A 2-hour NAS-wide instance is used, which represents the high-traffic period of a day, with 2,326 paths and 3,054 flights involved. The chance constraints are only performed in the focused area, the Chicago Air Route Traffic Control Center (ZAU) and the Indianapolis Air Route

Traffic Control Center (ZID). The area of these two centers are shown in Figure 4.5 and the details of associated high altitude sectors are listed in Table 4.8 and 4.9, which are extracted from the Future ATM concepts evaluation tool (FACET) [74]. The joint Gaussian distribution is adopted for the correlated capacities, where the mean is set as the normal capacity and the covariance matrix is randomly set up with numbers from [0,2].

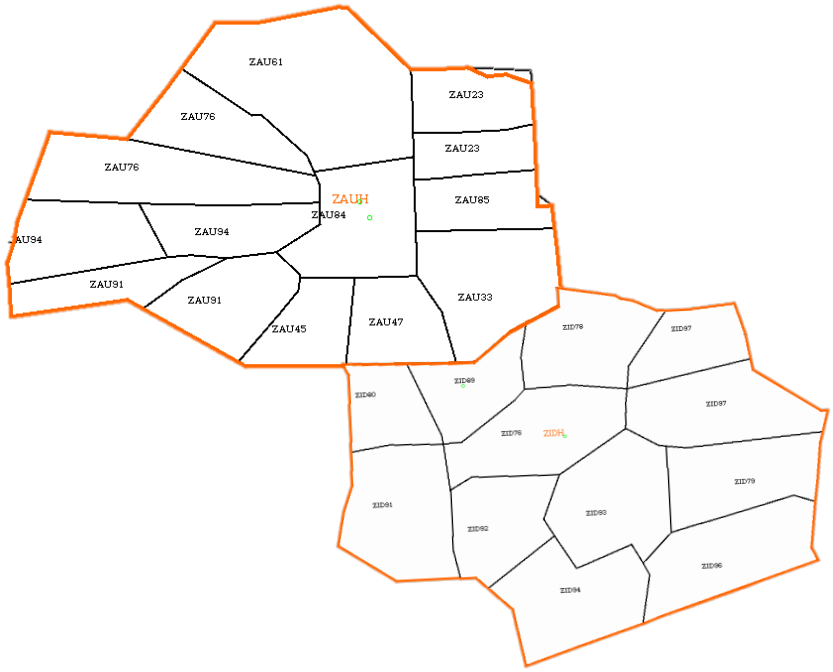


Figure 4.5. ZAU center and ZID center.

To demonstrate that the approximation-based approach could provide a reliable solution, several large-scale ATFM cases with different problem sizes are tested. Since the size of the ATFM problem is highly related to the number of sectors, the simulation plans with various number of sectors (where chance constraints are performed) are extracted. The focused sectors for each case are listed in Table 4.7, where different service levels are assigned for each case.

Table 4.7. Simulation cases with focused sectors

Cases	Sectors	Service level
5 sectors	[ZAU84, ZAU47, ZAU33, ZAU85, ZAU23]	85%
7 sectors	[ZAU84, ZAU47, ZAU33, ZAU85, ZAU23 ZID80, ZID89]	80%
10 sectors	[ZAU84, ZAU47, ZAU33, ZAU85, ZAU23 ZID80, ZID89, ZID78, ZID76, ZID97]	75%
15 sectors	[ZAU84, ZAU47, ZAU33, ZAU85, ZAU23 ZID80, ZID89, ZID78, ZID76, ZID97 ZID96, ZID93, ZID94, ZID92, ZID91]	70%
20 sectors	[ZAU84, ZAU47, ZAU33, ZAU85, ZAU23 ZID80, ZID89, ZID78, ZID76, ZID97 ZID96, ZID93, ZID94, ZID92, ZID91 ZAU61, ZAU76, ZAU94, ZAU91, ZAU45]	65%

The 2 hour horizon is divided into 24 intervals, each interval is 5 minutes. For each time interval, the service level is evaluated with the Monte Carlo method in Section 2.2.2. Based on the results from the chance constrained model, 10000 Monte Carlo simulations are run for each time interval and the service level is the percentage of successful tasks, i.e. the number of flights in all focused sectors are below the simulated capacities. The results of service level evaluation are shown in Figure 4.6. It is clear that the required service levels are kept for all case, though there are some small violations at rush time. There are two possible reasons for the violations: first, there are small approximation error in the convex approximation approach; second, the 10000 Monte Carlo simulation may not converge to the true probability with the joint distribution. Different Monte Carlo simulations may result in slightly different service levels, but the average trend is still kept around the required service level.

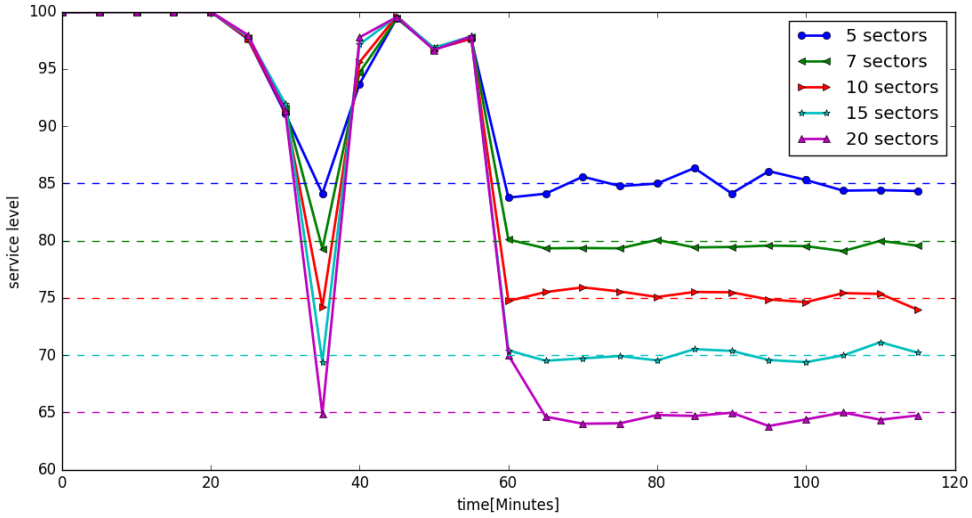


Figure 4.6. Service level for large scale experiment at each time step.

Lastly, Figure 4.7 shows the relative objective error between the approximation-based approach and the accurate optimal solution. The average error gap is kept around 5% with various number of sectors from 5 to 20, which confirmed that the quality of approximation-based solution is also reliable for large-scale ATFM problems. Due to the exponentially growing complexity of the MILP method, we only test up to 20 sectors. Moreover, if the number of possible capacity combinations is over 100000, a sample process is applied to filter the first 100000 feasible combinations which satisfy the required service level. Actually, the approximation-based approach can handle large-scale problem with more than 20 sectors, but it is difficult to get the accurate optimal solution for comparison.

4.6 Conclusion

This chapter introduces a novel polynomial approximation-based chance-constrained optimization method to address uncertainty in ATFM, which could provide a compu-

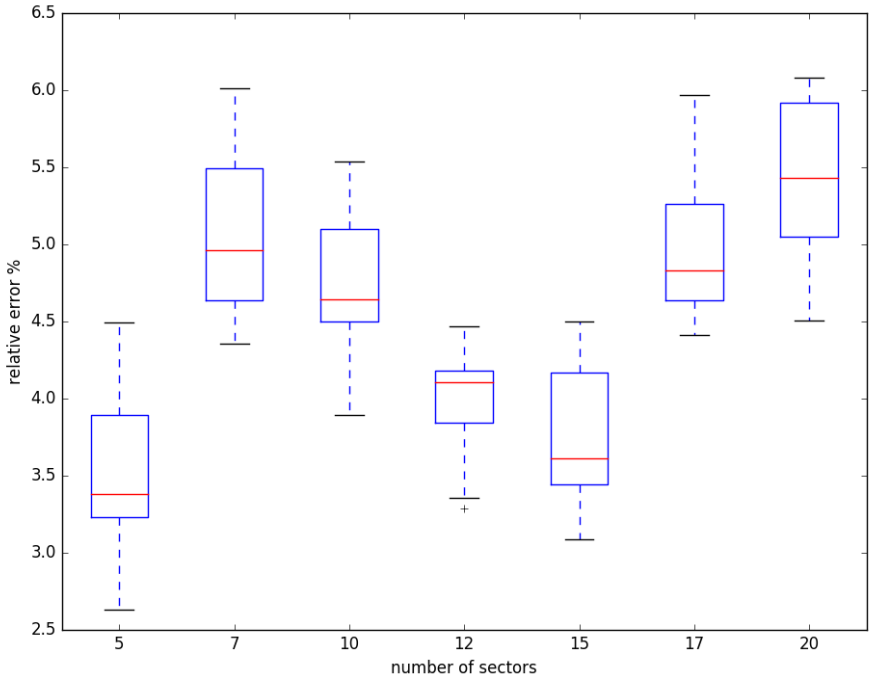


Figure 4.7. Relative error with different ATFM problem sizes.

tationally efficient approach. Based on a previous deterministic Integer Programming optimization model of ATFM, a chance-constrained model is developed to include probabilistic sector capacity constraints. Then, a polynomial approximation-based approach can be applied to efficiently solve such a large-scale chance-constrained optimization problem. The approximation is based on the numerical properties of the Bernstein polynomial, which is capable of effectively controlling the approximation error for both the function value and gradient. Thus, the gradient mapping (a first-order algorithm) is adopted to obtain a satisfactory solution which is expected to be optimal. Numerical results are reported to evaluate the polynomial approximation-based approach by comparing it with the brute-force method, which demonstrates that the approximation-based approach could provide reliable solutions. This chance-constrained optimization method and its computation platform are potentially helpful in their application to many domains in air transportation. This method and platform

are not only limited to ATFM, but can also be extended to airport surface operations and airline management under uncertainties.

Table 4.8. High altitude sector information for ZAU

Center ID	Center name	Altitude	Sector name	Sector ID	Capacity
3	Chicago	High	ZAU95	954	16
3	Chicago	High	ZAU94	955	16
3	Chicago	High	ZAU92	956	16
3	Chicago	High	ZAU91	957	18
3	Chicago	High	ZAU90	958	16
3	Chicago	High	ZAU85	959	14
3	Chicago	High	ZAU84	960	15
3	Chicago	High	ZAU83	961	16
3	Chicago	High	ZAU82	962	13
3	Chicago	High	ZAU76	963	17
3	Chicago	High	ZAU75	964	17
3	Chicago	High	ZAU71	965	18
3	Chicago	High	ZAU61	966	18
3	Chicago	High	ZAU60	967	18
3	Chicago	High	ZAU52	968	16
3	Chicago	High	ZAU47	969	14
3	Chicago	High	ZAU46	970	10
3	Chicago	High	ZAU45	971	13
3	Chicago	High	ZAU36	972	12
3	Chicago	High	ZAU34	973	11
3	Chicago	High	ZAU33	974	15
3	Chicago	High	ZAU25	975	13
3	Chicago	High	ZAU24	976	15
3	Chicago	High	ZAU23	977	15

Table 4.9. High altitude sector information for ZID

Center ID	Center name	Altitude	Sector name	Sector ID	Capacity
8	Indianapolis	High	ZID99	998	18
8	Indianapolis	High	ZID98	999	18
8	Indianapolis	High	ZID97	1000	18
8	Indianapolis	High	ZID96	1001	18
8	Indianapolis	High	ZID95	1002	21
8	Indianapolis	High	ZID94	1003	17
8	Indianapolis	High	ZID93	1004	19
8	Indianapolis	High	ZID92	1005	17
8	Indianapolis	High	ZID91	1006	19
8	Indianapolis	High	ZID75	1007	13
8	Indianapolis	High	ZID88	1008	14
8	Indianapolis	High	ZID87	1009	15
8	Indianapolis	High	ZID86	1010	18
8	Indianapolis	High	ZID85	1011	17
8	Indianapolis	High	ZID84	1012	16
8	Indianapolis	High	ZID83	1013	16
8	Indianapolis	High	ZID82	1014	16
8	Indianapolis	High	ZID81	1015	17
8	Indianapolis	High	ZID79	1016	18
8	Indianapolis	High	ZID78	1017	16
8	Indianapolis	High	ZID77	1018	15
8	Indianapolis	High	ZID76	1019	17
8	Indianapolis	High	ZID89	1020	14
8	Indianapolis	High	ZID80	1021	13
8	Indianapolis	High	ZID66	1022	14
8	Indianapolis	High	ZIDPKZ	1023	20

5. DISTRIBUTED COMPUTING FRAMEWORK

Due to the dynamic nature of national air traffic system, optimizing the ATFM problem is often time-consuming, especially for large-scale problems. As shown in Chapter 4, the ATFM problem is generally modeled as an integer programming problem, which requires computationally expensive optimization algorithms. The nonlinear constraints introduced by the uncertainty makes the stochastic version of ATFM more difficult to solve.

To overcome the computational burden, this chapter presents a customized Spark-based framework that greatly speeds up the optimization process, where Spark is a big data cluster-computing platform. First, the Apache Spark framework is introduced, including its system architecture and its advantages over Hadoop MapReduce. Then the development of the distributed computing framework for chance constrained model is presented in Section 5.2. The independent approximation processes for marginal functions in Chapter 3, are encapsulated into the Spark-based data processing model such that the approximation is automatically scheduled to run in parallel. In Section 5.3, the framework is validated to be efficient by applying on the chance constraint TFM problem introduced in Chapter 4. Section 5.4 demonstrated a comprehensive comparison between the Spark-based framework and the Hadoop MapReduce based framework.

5.1 Overview of Apache Spark

The Apache Spark [1] is a fast and general-purpose cluster computing platform, which is an open-source project under the Apache Software Foundation [2] for large-

[1]URL: https://spark.apache.org/

[2]URL: https://www.apache.org/

scale data processing. Spark is designed to extends the basic and popular MapReduce model of Hadoop to efficiently support more types of computations, such as SQL, Machine learning and graph processing. The general concept of MapReduce model consists of two sequential steps: *map* and *reduce*, where *map* applies a customized function to each listed element in parallel and returns the result in the same sized list; *reduce* applies another customized function to combine all elements into a single value. Both of these two functions need to read from and write to the disks, which takes time for these internal processes. However, Spark abstracts away from these two steps to offer supports for a wide range of applications.

The resilient distributed dataset (RDD) is the basic abstraction in Spark. An RDD in Spark is an immutable, partitioned collection of elements that can be operated on in parallel. In Spark all work is expressed as either creating new RDDs, transforming existing RDDs, or calling operations on RDDs to compute a result. Spark automatically distributes the data contained in RDDs across a cluster and parallelizes the operations to perform on them. Moreover, Spark can also work with HDFS, which is helpful in managing distributed dataset. One of the main features Spark offers for speed is the ability to run computations in memory, while the system is also more efficient than MapReduce for complex applications running on disk [29]. In distributed mode, Spark uses a master/slave architecture with one central coordinator (the *driver*) and many distributed workers (*executors*), as shown in Fig 5.1. The driver is the process where the main method of the program runs. The driver converts the program into tasks and schedules tasks on executors dynamically based on each executor's computational ability, and this dynamic allocation feature is the key to balance workload between workers, which is another key improvement from MapReduce. After finishing each scheduled task, the worker will return the necessary result to the driver and proceed to the next scheduled task.

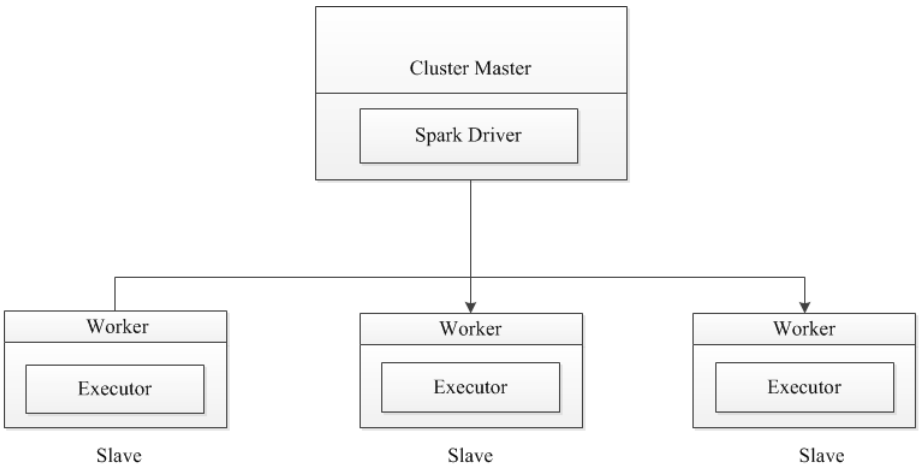


Figure 5.1. Distributed Spark system with master/slave architecture.

5.2 Distributed Framework Based on Spark

Recall that the construction of the polynomial approximation for each individual marginal function $g_t^i(x^i)$ is independent in Chapter 3. Therefore, the approximation process for the chance constraints is highly suitable for performing parallel computing. A customized distributed computing framework for the chance-constrained TFM optimization, based on the polynomial approximation method, is shown in Figure 5.2. The constraints for the chance-constrained model can be separated into two parts: the linear constraints ($Ax \leq b, x \geq 0$) and the chance constraints ($g_t(x)$). The flight plan and the probability information will provide the necessary input to construct the model's parameters. To perform the gradient mapping algorithm (a first order algorithm), the feasible convex set Q and the first-order information ($g_t(x)$ and $\bigtriangledown g_t(x)$) are two key inputs. The feasible convex set Q can be provided by the linear constraints, which is easy to perform.

The first-order information can be obtained by approximating the chance constraints, which is the key part of the distributed computing framework. First, for each time step t, there is an individual independent chance constraint, such that the whole chance constraint can be decomposed, time by time, into T individual prob-

lems. Second, since we approximate each of the marginal functions independently, the individual problem for each time step t can be further decomposed, variable by variable, into n small problems. Therefore, the original problem can be decomposed into Tn small problems in total. Each node of the Spark cluster can be assigned to solve the small approximation problems based on their computational ability. After solving the Tn independent problems in parallel, the results can be gathered by the master of the cluster, to provide first-order information ($g_t(x)$ and $\bigtriangledown g_t(x)$) to the gradient mapping algorithm.

Note that the gradient mapping algorithm needs to call the approximation process during every iteration until the final converge. Therefore, the distributed computing framework can greatly improve the computational efficiency by the fact that the approximation process has the most expensive computing cost of the whole process.

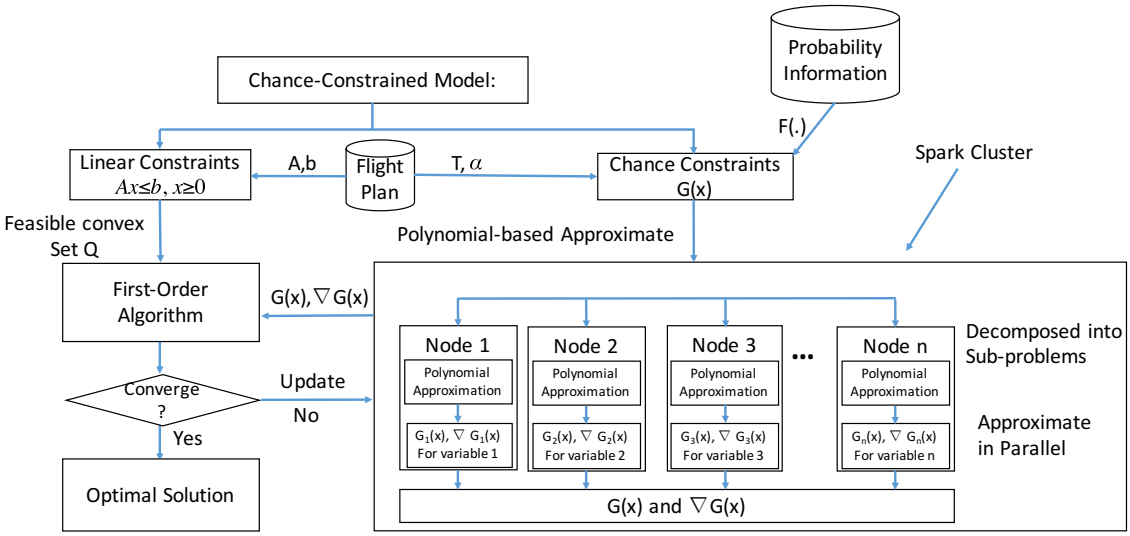


Figure 5.2. Distributed framework based on Spark

5.3 Distributed Computing Framework Validation

This section presents a real case ATFM optimization, employing the proposed distributed computing framework. The traffic data are extracted from the ASDI, which provides historical traffic data, as well as flight plans. The same joint distribution information and the flight plan in Chapter 4.5, which includes 10 sectors in the ZAU and ZID centers, is used in the section. The degree of polynomial is chosen to be 10. The optimization is performed on a small Spark cluster with six nodes. Each node of the Spark cluster is a DELL workstation configured with an 8-processor CPU. All workstations run UBUNTU 14.04 with Spark 1.3.1. The optimization subproblems were solved by calling Gurobi 6.0.2.

The running time of the optimization with different paralleling configurations is demonstrated in Figure 5.3. As a tuning parameter to control the concurrency level, the maximum number of executors allowed on a machine can be adjusted. Since the 8-processor CPU can handle 8 threads simultaneously, the maximum number of executors per machine can be up to eight. The running time decreases when more executors are used for a fixed number of machines. However, the speedup is not linear and becomes less noticeable as the number of executors approaches 8. The reason is the increasing overhead for allocating CPU time to the processors. Another speedup pattern can be observed as more machines are launched. The speedup is also not linear by the fact that it is more and more difficult to achieve further speedup as more machines are deployed. The runtime is reduced from 127 minutes with 1 machine and 1 executor to 12.3 minutes with 6 machines and 8 executors. The speedup increases about tenfold, which is less than the theoretical $6 \times 8 = 48$ times. This is due to the increasing overhead for the synchronization and communication between nodes, which is a common issue in distributed computing programming. Overall, it is clear that the distributed computing framework, indeed, improves the computational efficiency of the polynomial approximation-based approach.

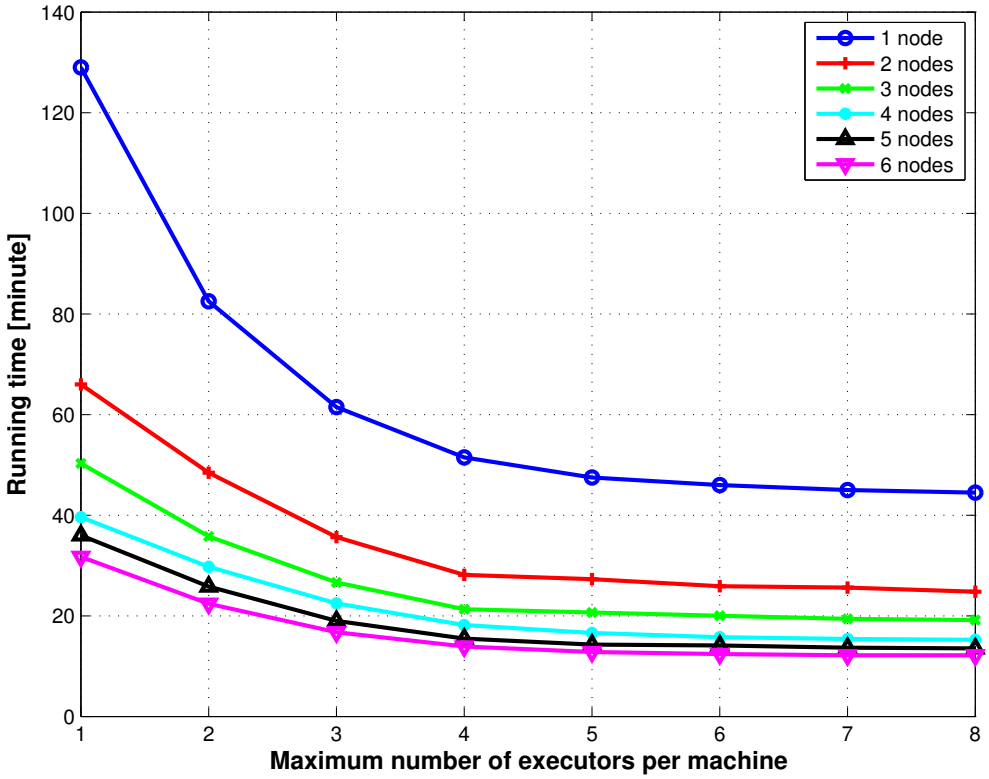


Figure 5.3. Runtime of the chance constrained AFTM optimization with the distributed computing framework.

As the other key factor of the computation time, the relationship between runtime and the polynomial degree is demonstrated in Figure 5.4, where the number of processors is fixed to be 40. Even though high polynomial degree could help reduce the approximation error, the increase of degree could also explode the computation time. Therefore the choice of the polynomial degree is a balance between the computation time and the solution accuracy. In fact, the degree can be chosen between 10 to 15 to provide a good approximation, based on the experimental experience. Further increasing the degree level will provide little help for the quality of the solution.

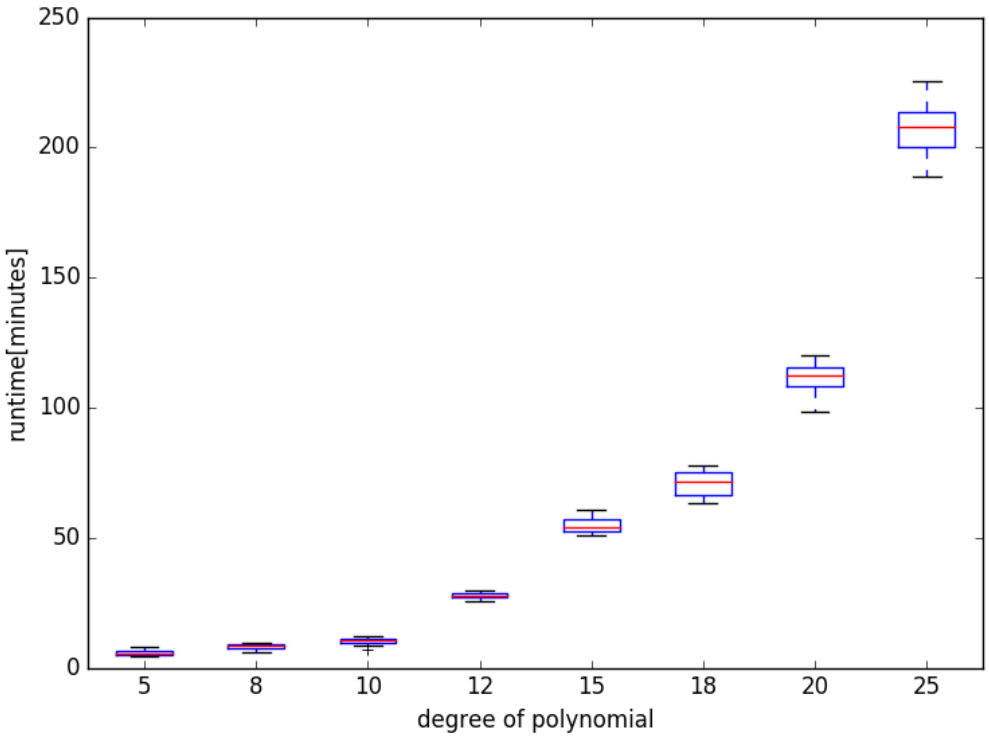


Figure 5.4. Runtime of the chance constrained AFTM optimization with different polynomial degrees

5.4 Performance Comparison between Spark and Hadoop

This section presents some details of computational performances and features for the distributed computing framework that is run on a Spark cluster. The goal is to compare the performance of the chance TFM problems on Spark with that on the existing Hadoop MapReduce framework [28]. To make the comparison meaningful, the simulation experiment was set up with the same hardware parameters. Both the Hadoop and the Spark cluster was launched with six nodes where each node was DELL workstations configured with an Intel i7 CPU and a 16 Gigabyte RAM. All workstations still run UBUNTU 14.04 with Spark 1.3.1.

The performance of the chance constrainedco TFM problem on Spark and Hadoop was compared by running the same 2-hour chance constrained instance above. The running time of the optimization on Spark and Hadoop is shown in Figure 5.5. As shown in the previous section, the running time decreases as more executors are launched for both Hodoop and Spark. Again, the speedup is not linear to the number of threads, which confirms the common issue with the parallel programming model due to inherent overheads such as communication and synchronization between workers.

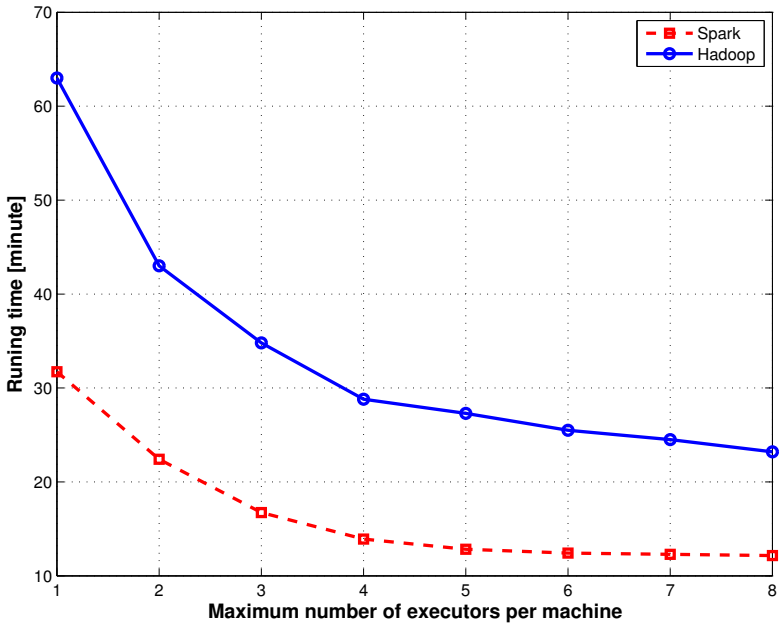
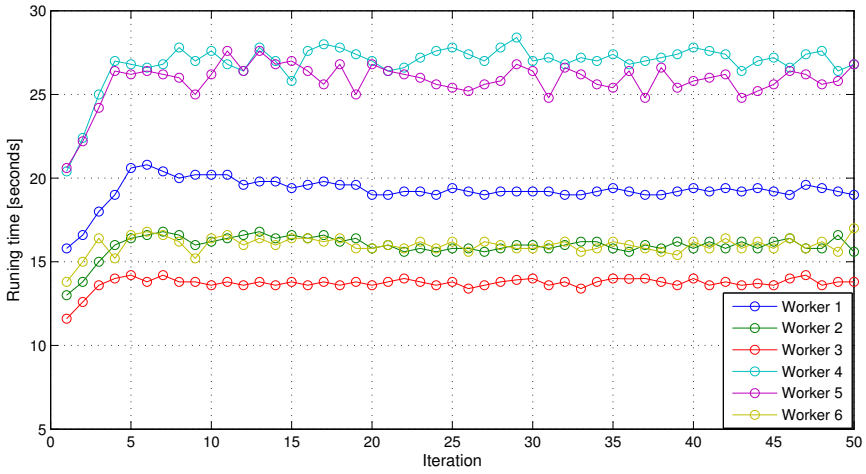
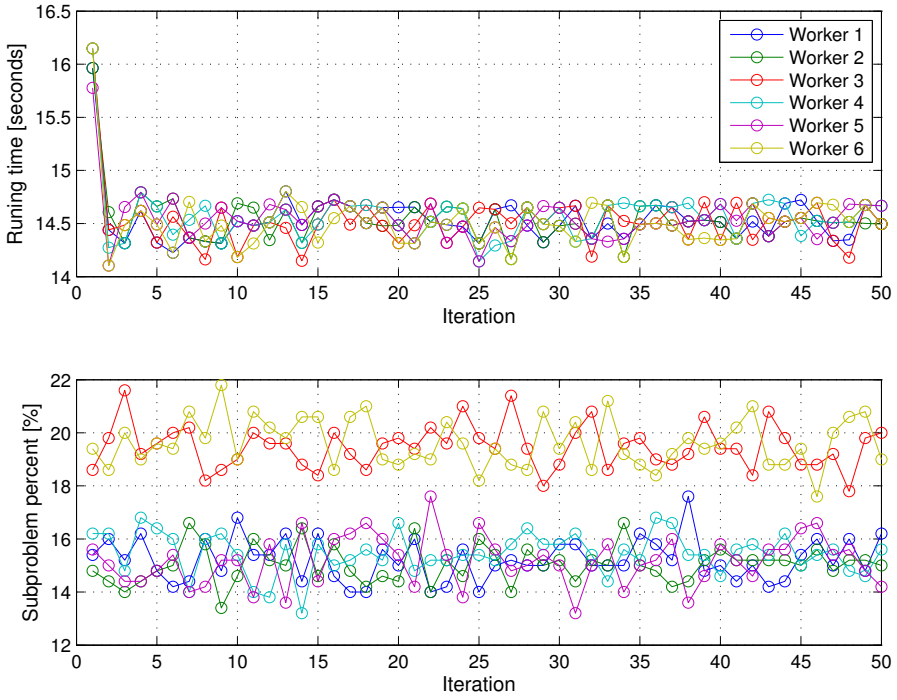


Figure 5.5. Running time decreasing as a function of the number of threads per machines.

Comparing the running time of Spark and Hadoop in Figure 5.5, Spark is about 2 times faster than Hadoop. The running time with maximum computing power (8 executors per machine) is reduced from 23.2 minutes to 12.3 minutes. One of the key reasons is that Spark's in-memory computation cuts down internal input and output processes for iterative jobs. In MapReduce, the input/output data has



(a) Unbalanced workload between worker in hadoop.



(b) Runtime and workload between workers in Spark

Figure 5.6. Comparison of Hadoop and Spark runtime.

to be read from/stored to HDFS in each iteration and there is significant cost of starting and finishing a MapReduce job. However, Spark's in-memory computation avoid such cost that parameters updates can be cached in memory between iterations in the optimization process, which contributes to the speedup. Another key reason is that unbalanced workloads cause idle time for some workers on Hadoop cluster. Figure 5.6(a) shows the running time of solving subproblems on each worker of a Hadoop cluster. Worker 3 is about 13.5 seconds ahead of worker 4 in each iteration. In the implementation of Hadoop, the subproblems were evenly distributed to each worker in the beginning. Although all the workers have the same configuration, the complexity of subproblems has a wide range due to the difference of each marginal function. However, Spark can dynamic allocate subproblems to each of the workers, which helps to reduce the gap. Figure 5.6(b) shows the running time and workload of solving subproblems on workers in Spark. Worker 3 and 6 were more powerful than others such that the Spark driver distributed 5% more subproblems to them in the same time. As a result, the runtime is almost the same on each worker in each iteration. This feature of Spark helps avoid the idle time associated with Hadoop cluster such that the average solving time in each iteration is improved.

Table 5.1. Comparison of MapReduce framework and Spark framework

	MapReduce	Spark
Code(line)	Over 1000	Under 500
Speedup	15.8	19.9
Workload balancing	No	Yes
Fault tolerance	Yes	Yes

The main differences between the MapReduce framework and Spark framework are summarized in Table 5.1. By take the advantage of Spark's RDD framework, the process doesn't need to follow the standard *map* and *reduce* procedure in MapReduce

framework such that the list-processing job is easier to program with fewer lines of code. Beyond list-processing job, Spark's RDD framework abstracts away MapReduce implementation details such that it can cover a wide range of workloads to become a capable platform for other large-scale dynamical systems which is not tractable on a traditional computational platform. In addition, the speedup result shown in Table 5.1 is compared with the results from a standalone computer.

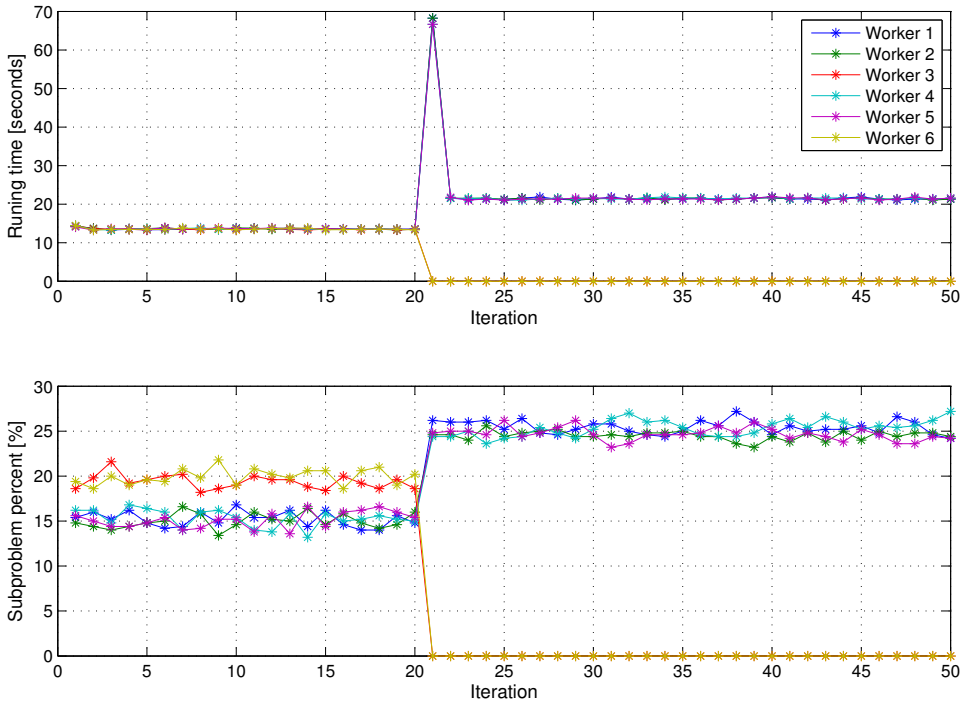


Figure 5.7. Failure tolerance test when worker 3 and 6 were shut down.

As a distributed-based computational framework, Spark also has the advantage on its built-in fault tolerance capability. A test is shown in Figure 5.7 where two workers (worker 3 and 6) were purposely shut down during the iterations. The optimization was held up by the shut down. The Master retried to schedule the failure tasks to these two workers for several times. When the Master detects that these workers fail to

respond for a preset period of time, it reschedules the tasks to be re-executed on other workers so that the job can continue. In the remaining iterations, Spark recalculated the tasks splits and assigned tasks to the alive workers. As a result, the whole optimization was completed without interruptions. Fault tolerance is an important feature that the non-cloud-based computational framework does not provide.

5.5 Discussion

Besides the demonstrated parallel approximation process in this chapter, the distributed framework is also useful in other two processes for the chance constrained ATFM problem. One is the traffic data analysis, which is a key prior processing for constructing the data-driven traffic model, such as the LTM. The distributed framework based on Spark is greatly helpful in processing large-scale data in parallel. Machine learning and data mining algorithms (e.g. clustering algorithms) can be further integrated into the whole process for route identification and probability distribution estimation. An example parameter estimation with Spark-based large-scale data analysis can be found in a previous work [75].

Another one is the Monte Carlo simulation for the joint probability evaluation. Since there are massive independent random sampling processes in the Monte Carlo simulation, the distributed framework can be easily adapted to handle it, where each worker can run their own simulations based on their individual local samples and the cluster master will gather together all the simulation results to provide the estimated joint probability. Such a design can further improve the computational efficiency in the approximation step, which is also one of the most time-consuming step in the whole algorithm.

In summary, a customized Spark-based optimization architecture for large-scale chance constrained optimization problems is firstly proposed and tested. In comparison with the MapReduce framework, chance constrained traffic flow management problems can be solved more efficiently in Spark. Spark's ability to run computa-

tion in memory saves the unnecessary step of generating output file for each job on MapReduce. In addition, the unbalanced workload limitation on MapReduce framework was overcome by Spark's dynamical schedule allocation feature. As a result, further speedup was achieved. Besides efficiency, Spark framework abstracts away MapReduce implementation details to help reduce the difficulty of programming (as measured by fewer lines of code) and Spark's distributed framework also demonstrates runtime fault tolerance. These features, as demonstrated by our experiments, make the Spark a capable platform that can potentially solve and analyze some large-scale dynamical systems which is not tractable on a traditional computational platform.

It is worth emphasizing that a fast computational framework with the proper model is the key to help deliver real-time solutions for air transportation system. In particular, a faster computational platform can solve larger problems in the same time and solve the same problem more often in face of disruptions. Moreover, real-time solutions are critical to the applicability of TFM models due to the dynamic nature of air transportation system. Therefore, a faster computational framework could have significant effect on the improvement of air transportation system.

6. SUMMARY AND FUTURE WORK

6.1 Summary

This dissertation introduces a chance constrained optimization method to address the uncertainties in air transportation system, which could provide a computationally efficient algorithm. Different with the classic stochastic scenario tree-based method, the chance-constrained model proposes to include probabilistic capacity constraints of airspace resources (e.g. airport and sector), which could guarantee the robustness of the optimal solution. Beginning with the basic SAGHP, the concept of service level is introduced to evaluate the robustness of optimal planning under uncertainty. The service level represents the chance that all constraints are not violated, which provides a event-oriented performance criterion for risks. To achieve required service level, the chance constrained model for GHP is developed based on a previous deterministic Integer Programming optimization model. Simulation has shown that the service level is kept well above the required level, while the service level for the traditional scenario tree based method and the deterministic method is as low as 20% in rush hours. Especially, the concept of joint planning for multiple related airports is also well validated under a joint service level. With the similar idea, the chance constrained model is extended to formulate a traffic flow management problem under probabilistic sector capacity, which is derived from a previous deterministic linear model, the Link Transmission Model. Simulation results show that the chance constraints are well controlled above the required service level at each discrete time interval. Moreover, the chance constrained model shows the ability to provides a quantized way to balance the solutions robustness and potential cost by choosing a proper service level.

In order to solve the exponentially growing complexity issue faced by the MILP formulation (a brute force method), a novel convex approximation based approach is

proposed to efficiently solve such a large-scale chance-constrained optimization problem. The approximation is based on the numerical properties of the Bernstein polynomial, which is capable of effectively controlling the approximation error for both the function value and gradient. Thus, a first-order algorithm can be adopted to obtain a satisfactory solution which is expected to be optimal. Numerical results are reported to evaluate the convex polynomial approximation-based approach by comparing it with the brute-force method, which demonstrates that the approximation-based approach could provide reliable solutions. Another key feature of the convex approximation is the specially designed massive independent marginal function, which can be approximated in parallel.

Based on the massive independent approximation processes in the convex approximation based approach, a distributed computing framework is designed to further improve the computational efficiency. By taking the advantage of Spark, the distributed framework enables concurrent executions for the convex approximation processes such that multiple distributed computing facilities can be connected to solve the large scale time-consuming problem. As an extension from a basic cloud computing package, Hadoop MapReduce, Spark provides advanced features on in-memory computing and dynamical task allocation balancing. Simulations show that the Spark based framework can greatly improve the computational efficiency, which is about two times faster than the previous MapReduce based framework.

As artificial intelligent and advanced automation is highly involved into the Next Generation Air Transportation System, the incorporation of uncertainty into air traffic management decision making continues to gain interest in the ATM community. The combination of convex approximation and distributed computing framework will continue to provide more efficient supports for the modern air transportation system.

6.2 Future Work

This chance constrained optimization method and its computation platform provide a useful platform for handling uncertainties in air transportation. To push this platform further, the following are the primary future directions of the work presented in this dissertation.

- **Investigating the selection for service level.** The current implementation is to provide an optimal solution once the service level is defined. However, the selection of the service level is actually a key prior process if the idea of service level will be integrated into the future air transportation management system. The service level is highly related to many factors, such as the weather prediction, the safety requirement and the operational cost. How to formulate a proper selection process for the service level in practice operation will be an interesting problem.

- **Developing dynamic model with Markov decision process.** Currently, the joint capacity distribution is homogeneous for all the time, i.e. only one identical joint capacity distribution is used for every time step. It is desirable that different types of capacity distributions should be considered in the problem, which represent different weather conditions at a specific time. Moreover, as a time series process, the states of the model should be dynamic. Intuitively, one could expect better solutions when capacity distributions also rely on previously states. However, the current model considers the capacity distribution to be independent for each time step, which fails to model the dynamic connection of capacity distribution on time dimension. These two limitations are potentially solvable with a dynamic model based on Markov decision process, which is possible to be efficient by using the Spark-based distributed computing framework, if Markov chain Monte Carlo is involved.

- **Improving the rounding process for integer solution.** The current convex approximation method will produce a continuous solution, then a simple

Branch-and-Bound process is applied to find a close integer solution. Such an integer solution cannot guarantee to be optimal, probably sub-optimal. Moreover, the B&B process often takes time to traverse nodes along the developed search tree. Therefore, efforts can be extended to find a proper and efficient rounding process to get the final integer solution.

- **Exploring new applications of chance constrained model in air transportation.** This chance constrained optimization method and its computation platform are potentially helpful in their application to many domains in air transportation. This method and platform are not only limited to TFM, but can also be extended to other classical problems in ATM community, such as runway scheduling problem, rerouting problem, airport surface operations and airline management under uncertainties. These problems are often formulated as a MILP problem and most of the current work just ignores the uncertainties in the realistic operations. By taking the advantage of the chance constrained model and the distributed framework, it is highly possible to provide reliable, efficient, robust and optimal solutions for these problems.

REFERENCES

REFERENCES

[1] U.S. Department of Transportation Federal Aviation Administration Aviation Policy and Plans. Faa aerospace forecast: Fiscal years 2017-2037. Technical report, Federal Aviation Administration, 2017.

[2] Michael Ball, Cynthia Barnhart, Martin Dresner, Mark Hansen, Kevin Neels, Amedeo Odoni, Everett Peterson, Lance Sherry, Antonio A Trani, and Bo Zou. Total delay impact study: a comprehensive assessment of the costs and impacts of flight delay in the united states, nextor. 2010.

[3] Bureau of Transportation Statistics. *Airline On-Time Statistics and Delay Causes*, 2017. Avialble at `https://www.transtats.bts.gov`.

[4] Banavar Sridhar, Shon R Grabbe, and Avijit Mukherjee. Modeling and optimization in traffic flow management. *Proceedings of the IEEE*, 96(12):2060–2080, 2008.

[5] Jun Chen, Daniel DeLaurentis, and Dengfeng Sun. Dynamic stochastic model for converging inbound air traffic. *Journal of Guidance, Control, and Dynamics*, pages 2273–2283, 2015.

[6] Nikolas Pyrgiotis, Kerry M Malone, and Amedeo Odoni. Modelling delay propagation within an airport network. *Transportation Research Part C: Emerging Technologies*, 27:60–75, 2013.

[7] Michael Nolan. *Fundamentals of air traffic control*. Cengage learning, 2010.

[8] Amedeo R Odoni. The flow management problem in air traffic control. In *Flow control of congested networks*, pages 269–288. Springer, 1987.

[9] Mostafa Terrab and Amedeo R Odoni. Strategic flow management for air traffic control. *Operations research*, 41(1):138–152, 1993.

[10] Robert Hoffman and Michael O Ball. A comparison of formulations for the single-airport ground-holding problem with banking constraints. *Operations Research*, 48(4):578–590, 2000.

[11] Thomas Vossen, Michael Ball, Robert Hoffman, and Michael Wambsganss. A general approach to equity in traffic flow management and its application to mitigating exemption bias in ground delay programs. *Air Traffic Control Quarterly*, 11(4):277–292, 2003.

[12] Peter B Vranas, Dimitris J Bertsimas, and Amedeo R Odoni. The multi-airport ground-holding problem in air traffic control. *Operations Research*, 42(2):249–261, 1994.

[13] Dimitris Bertsimas and Sarah Stock Patterson. The air traffic flow management problem with enroute capacities. *Operations Research*, 46(3):406–422, 1998.

[14] Dimitris Bertsimas, Guglielmo Lulli, and Amedeo Odoni. The air traffic flow management problem: An integer optimization approach. In *Integer programming and combinatorial optimization*, pages 34–46. Springer, 2008.

[15] Dimitris Bertsimas, Guglielmo Lulli, and Amedeo Odoni. An integer optimization approach to large-scale air traffic flow management. *Operations Research*, 59(1):211–227, 2011.

[16] Dimitris Bertsimas and Sarah Stock Patterson. The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science*, 34(3):239–255, 2000.

[17] Padmanabhan K Menon, Gregory D Sweriduk, and Karl D Bilimoria. New approach for modeling, analysis, and control of air traffic flow. *Journal of Guidance, Control, and Dynamics*, 27(5):737–744, 2004.

[18] Carlos F Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994.

[19] Carlos F Daganzo. The cell transmission model, part ii: Network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.

[20] Dengfeng Sun, Issam S Strub, and Alexandre M Bayen. Comparison of the performance of four eulerian network flow models for strategic air traffic management. *Networks and Heterogeneous Media*, 2(4):569, 2007.

[21] Dengfeng Sun and Alexandre M Bayen. Multicommodity eulerian-lagrangian large-capacity cell transmission model for en route traffic. *Journal of Guidance, Control, and Dynamics*, 31(3):616–628, 2008.

[22] Yi Cao and Dengfeng Sun. Link transmission model for air traffic flow management. *Journal of Guidance, Control, and Dynamics*, 34(5):1342–1351, 2011.

[23] Dengfeng Sun, Alexis Clinet, and Alexandre M Bayen. A dual decomposition method for sector capacity constrained traffic flow optimization. *Transportation Research Part B: Methodological*, 45(6):880–902, 2011.

[24] P Wei, Y Cao, and D Sun. Total unimodularity and decomposition method for large-scale air traffic cell transmission model. *Transportation Research Part B: Methodological*, 53:1–16, 2013.

[25] Weigang Li, Marcos Vinicius Pinheiro Dib, Daniela Pereira Alves, and Antonio Marcio Ferreira Crespo. Intelligent computing methods in air traffic flow management. *Transportation Research Part C: Emerging Technologies*, 18(5):781–793, 2010.

[26] Joseph Rios and Kevin Ross. Massively parallel dantzig-wolfe decomposition applied to traffic flow scheduling. *Journal of Aerospace Computing, Information, and Communication*, 7(1):32–45, 2010.

[27] Yi Cao and Dengfeng Sun. A parallel computing framework for large-scale air traffic flow optimization. *Intelligent Transportation Systems, IEEE Transactions on*, 13(4):1855–1864, 2012.

[28] Yi Cao and Dengfeng Sun. Migrating large-scale air traffic modeling to the cloud. *Journal of Aerospace Information Systems*, 12(2):257–266, 2015.

[29] Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia. *Learning spark: Lightning-fast big data analysis*. O'Reilly Media, Inc., 2015.

[30] Octavio Richetta and Amedeo R Odoni. Dynamic solution to the ground-holding problem in air traffic control. *Transportation research part A: Policy and practice*, 28(3):167–185, 1994.

[31] Kan Chang, Ken Howard, Rick Oiesen, Lara Shisler, Midori Tanino, and Michael C Wambsganss. Enhancements to the faa ground-delay program under collaborative decision making. *Interfaces*, 31(1):57–76, 2001.

[32] Thomas WM Vossen and Michael O Ball. Slot trading opportunities in collaborative ground delay programs. *Transportation Science*, 40(1):29–43, 2006.

[33] Michael O Ball, Robert Hoffman, Amedeo R Odoni, and Ryan Rifkin. A stochastic integer program with dual network structure and its application to the ground-holding problem. *Operations Research*, 51(1):167–171, 2003.

[34] Avijit Mukherjee and Mark Hansen. A dynamic stochastic model for the single airport ground holding problem. *Transportation Science*, 41(4):444–456, 2007.

[35] Avijit Mukherjee and Mark Hansen. A dynamic rerouting model for air traffic flow management. *Transportation Research Part B: Methodological*, 43(1):159–171, 2009.

[36] Pei-chen Barry Liu, Mark Hansen, and Avijit Mukherjee. Scenario-based air traffic flow management: From theory to practice. *Transportation Research Part B: Methodological*, 42(7):685–702, 2008.

[37] Shubham Gupta and Dimitris J Bertsimas. Multistage air traffic flow management under capacity uncertainty: A robust and adaptive optimization approach. *Proceedings of the 51st AGIFORS Annual Symposium and Study Group Meeting*, 2011.

[38] Gillian Clare and Arthur Richards. Air traffic flow management under uncertainty: Application of chance constraints. In *Proceedings of the 2nd international conference on application and theory of automation in command and control systems*, pages 20–26. IRIT Press, 2012.

[39] Joseph Rios. Aggregate statistics of national traffic management initiatives. In *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*. American Institute of Aeronautics and Astronautics, Sep. 2010.

[40] Peng Wei, Jit-Tat Chen, Dominick Andrisani, and Dengfeng Sun. Routing flexible traffic into metroplex. In *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, Aug. 2011.

[41] Stephen Atkins. Observation and measurement of metroplex phenomena. In *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*. IEEE, Oct. 2008.

[42] Donald J Bowersox, David J Closs, M Bixby Cooper, et al. *Supply chain logistics management*, volume 2. McGraw-Hill New York, NY, 2002.

[43] Christian P Robert. *Monte carlo methods*. Wiley Online Library, 2004.

[44] Jonathan Cox and Mykel J Kochenderfer. Optimization approaches to the single airport ground-holding problem. *Journal of Guidance, Control, and Dynamics*, 38(12):2399–2406, 2015.

[45] Federal Aviation Administration. Faa operations and performance data. http://aspm.faa.gov, [retrieved 24 March 2017].

[46] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[47] Federal Aviation Administration. *San Francisco international airport capacity profile*, 2014. Avialble at `https://www.faa.gov/airports/planning_capacity/profiles/media/SFO-Airport-Capacity-Profile-2014.pdf`.

[48] Optimization Inc. Gurobi. *Gurobi optimizer reference manual, (accessed 27/03/16) [Online]. Available: www. gurobi. com*, 2016.

[49] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[50] Christopher A Provan, Lara Cook, and Jon Cunningham. A probabilistic airport capacity model for improved ground delay program planning. In *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, pages 2B6–1. IEEE, 2011.

[51] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge University Press, 2000.

[52] Jonathan Cox and Mykel J. Kochenderfer. Probabilistic airport acceptance rate prediction. In *AIAA Modeling and Simulation Technologies Conference*. American Institute of Aeronautics and Astronautics, Jan. 2016.

[53] Hamsa Balakrishnan and Bala G Chandran. Algorithms for scheduling runway operations under constrained position shifting. *Operations Research*, 58(6):1650–1665, 2010.

[54] Miltiadis A Stamatopoulos, Konstantinos G Zografos, and Amedeo R Odoni. A decision support system for airport strategic planning. *Transportation Research Part C: Emerging Technologies*, 12(2):91–117, 2004.

[55] Gustaf Sölveling and John-Paul Clarke. Scheduling of airport runway operations using stochastic branch and bound methods. *Transportation Research Part C: Emerging Technologies*, 45:119–137, 2014.

[56] Yeonju Eun, Inseok Hwang, and Hyochoong Bang. Optimal arrival flight sequencing and scheduling using discrete airborne delays. *Intelligent Transportation Systems,IEEE Transactions on*, 11(2):359–373, 2010.

[57] Michael V McCrea, Hanif D Sherali, and Antonio A Trani. A probabilistic framework for weather-based rerouting and delay estimations within an airspace planning model. *Transportation Research Part C: Emerging Technologies*, 16(4):410–431, 2008.

[58] Marcella Samà, Andrea DAriano, Paolo DAriano, and Dario Pacciarelli. Optimal aircraft scheduling and routing at a terminal control area during disturbances. *Transportation Research Part C: Emerging Technologies*, 47:61–85, 2014.

[59] Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[60] Andras Prékopa. Numerical solution of probabilistic constrained programming problems. *Numerical Techniques for Stochastic Optimization*, pages 123–139, 1988.

[61] Adrien Saumard and Jon A Wellner. Log-concavity and strong log-concavity: A review. *Statistics Surveys*, 8:45, 2014.

[62] Alexander Shapiro, Darinka Dentcheva, et al. *Lectures on stochastic programming: Modeling and theory*, volume 16. SIAM, 2014.

[63] Gurkaran Buxi and Mark Hansen. Generating probabilistic capacity profiles from weather forecast: A design-of-experiment approach. In *Proc. of USA/Europe Air Traffic Management Research & Development Seminar*, 2011.

[64] Jonathan Cunningham, Lara Cook, and Chris Provan. The utilization of current forecast products in a probabilistic airport capacity model. 2012.

[65] Jonathan Cox and Mykel J Kochenderfer. Probabilistic airport acceptance rate prediction. In *AIAA Modeling and Simulation Technologies Conference*, page 0165, 2016.

[66] Philip J Davis. *Interpolation and approximation*. Courier Corporation, 1975.

[67] GM Phillips and PJ Taylor. Approximation of convex data. *BIT Numerical Mathematics*, 10(3):324–332, 1970.

[68] EW Cheney. Introduction to approximation theory. *Chelsea, New York*, 1966.

[69] GW Stewart. Afternotes on numerical analysis. *University of Maryland at College Park*, 1993.

[70] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[71] Volpe Center Automation Applications Division. Aircraft situation display to industry: Functional description and interface control document. Technical report, Report no. ASDI-FD-001, Cambridge, Massachusetts, 2000.

[72] National Transportation Center Volpe. Enhanced traffic management system (etms). Technical report, Rep. VNTSC-DTS56-TMS-002,United States Department of Transportation, Cambridge, Massachusetts, 2000.

[73] Dengfeng Sun, Banavar Sridhar, and Shon R Grabbe. Disaggregation method for an aggregate traffic flow management model. *Journal of Guidance, Control, and Dynamics*, 33(3):666–676, 2010.

[74] Karl D Bilmoria, Sridhar Banavar, Gano B Chatterji, Kapil S Sheth, and Shon Grabbe. Facet: Future atm concepts evaluation tool. 2000.

[75] Jun Chen, Yi Cao, and Dengfeng Sun. Modeling, optimization, and operation of large-scale air traffic flow management on spark. *Journal of Aerospace Information Systems*, pages 1–13, 2017.

VITA

VITA

Jun Chen was born in the city of Zhongxiang, Hubei Province in Central China. He received a Bachelor degree of Science in Aeronautics Science and Engineering in 2012 from Beihang University, also known as Beijing University of Aeronautics and Astronautics, China. Then he entered the School of Aeronautics and Astronautics in Purdue University, West Lafayette, Indiana, and received the Master degree of Science in Aeronautics and Astronautics in 2014. After that, he continued to pursue his Ph.D. degree in the same school. His advisor is Professor Dengfeng Sun. His research area includes dynamics, control, machine learning and artificial intelligence, particularly in data-driven modeling, control and optimization for large-scale networked dynamical systems, with applications in mechanical and aerospace engineering such as air traffic control, traffic flow management, and autonomous vehicles. He is a member of AIAA and is an author of several journal papers. He is a recipient of the John L. and Patricia R. Rich Scholarship from Purdue University in 2017.