



Air traffic flow management under uncertainty using chance-constrained optimization



J. Chen^{a,*}, L. Chen^b, D. Sun^a

^aSchool of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana 47907, USA

^bDepartment of MIS, Operations Management and Decision Sciences, University of Dayton, Dayton, Ohio 45469, USA

ARTICLE INFO

Article history:

Received 8 September 2016

Revised 27 March 2017

Accepted 26 May 2017

Keywords:

Air traffic flow management
Chance-constrained optimization
Bernstein polynomial

ABSTRACT

In order to efficiently balance traffic demand and capacity, optimization of Air Traffic Flow Management (ATFM) relies on accurate predictions of future capacity states. However, these predictions are inherently uncertain due to factors, such as weather. This paper presents a novel computationally efficient algorithm to address uncertainty in ATFM by using a chance-constrained optimization method. 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, a polynomial approximation-based approach 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 is adopted to obtain a satisfactory solution, which is expected to be optimal. Numerical results are reported in order to evaluate the polynomial approximation-based approach by comparing it with the brute-force method. Moreover, since there are massive independent approximation processes in the polynomial approximation-based approach, a distributed computing framework is designed to carry out the computation for this method. 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.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

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 such that the demand and supply situation of ATFM is made worse. According to Sridhar et al. (2008), severe weather has been identified as the most important causal factor for traffic delays in the United States. Moreover, the weather forecast brings uncertainty into capacities, which also poses a significant challenge to ATFM. Strategic traffic flow management decisions made under uncertainty can cause nationwide severe congestion in the

* Corresponding author.

E-mail addresses: chen1241@purdue.edu (J. Chen), lchen1@udayton.edu (L. Chen), dsun@purdue.edu (D. Sun).

National Airspace System (NAS). This fact motivates the need for stochastic optimization algorithms for ATFM that account for capacity uncertainty.

In the past three decades, the flow management problem in air transportation has been studied by many researchers in order to address air traffic congestion. The first effort dates back to 1987, when Odoni was among the first to propose the mathematical formulation of ATFM (Odoni, 1987). Later, Bertsimas and Stock Patterson proposed a binary integer programming formulation that considered both airspace and airport capacities, known as the BSP model (Bertsimas and Patterson, 1998). 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 (Bertsimas and Patterson, 1998). Subsequently, Bertsimas presented several extensions of the BSP model to account for other features, such as rerouting (Bertsimas et al., 2008, 2011; Bertsimas and Patterson, 2000).

To overcome the computational limitation of the Lagrangian models, the Eulerian model of ATFM was proposed (Menon et al., 2004), which is inspired by the Daganzo Cell Transmission Model (Daganzo, 1994, 1995). 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. 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 (Sun and Bayen, 2008; Cao and Sun, 2011). Moreover, the distributed algorithms for these aggregate models have also been proposed using the dual decomposition method (Sun et al., 2011; Wei et al., 2013).

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, which is known as the Single Airport Ground Holding Problem (SAGHP). As one of the first attempts, Richetta and Odoni formulated a stochastic integer programming model for the SAGHP (Richetta and Odoni, 1994). Subsequently, Ball et al. proposed a modified version for the same problem, which solves for an optimal number of planned arrivals of aircraft during different time intervals (Ball et al., 2003). Recently, Mukherjee and Hansen proposed a model that incorporated dynamic rerouting into SAGHP (Mukherjee and Hansen, 2007, 2009). 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 (Liu et al., 2008). Moreover, the techniques are developed to determine probabilistic capacity profiles and scenario tree forecasts from historical data (Liu et al., 2008). 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 Gupta and Bertsimas (2011) 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 (Clare and Richards, 2012), 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.

This paper presents a novel polynomial approximation-based chance-constrained optimization method to address uncertainty in ATFM, which could provide a computationally efficient algorithm. 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, a polynomial approximation-based approach 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 the gradient. Thus, a first-order algorithm is adopted to obtain a satisfactory solution, which is expected to be optimal. Numerical results are reported in order to evaluate the polynomial approximation-based approach by comparing it with the brute-force method. Moreover, since there are massive independent approximation processes in the polynomial approximation-based approach, a distributed computing framework is designed to carry out the computation for this method. 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.

The rest of this paper is organized as follows. Section 2 introduces the chance-constrained ATFM problem. Section 3 introduces a polynomial approximation-based approach to overcome the limitation of the brute-force method. The main algorithm based on the polynomial approximation-based approach is presented with computational complexity in Section 4. Section 5 demonstrates the parallel computing framework for the approximation-based approach. Section 6 evaluates the

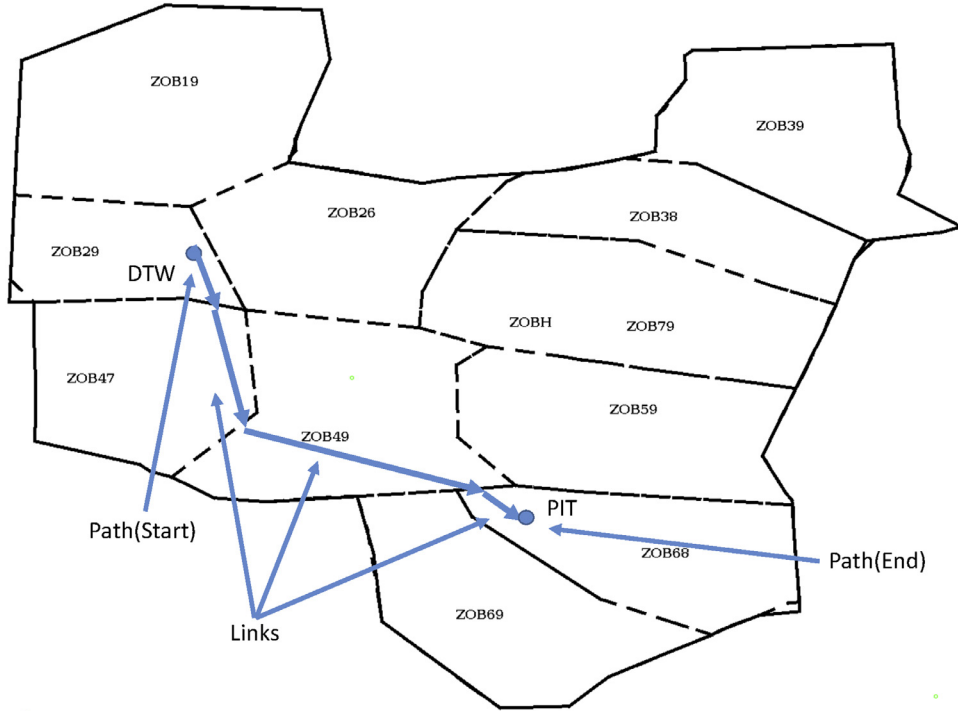


Fig. 1. Link transmission model.

performance of the approximation-based approach by comparing the results with the brute-force method. Section 7 concludes this paper.

2. Development of the chance-constrained model

2.1. Deterministic aggregate traffic flow management modeling

The proposed stochastic Traffic Flow Management (TFM) model is derived based on a previous deterministic Link Transmission Model (LTM) for ATFM (Cao and Sun, 2011). 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 Enhanced Traffic Management System (ETMS) (Division, 2000; Volpe, 2000). As shown in Fig. 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 (Sun and Bayen, 2008).

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 $f^k(t)$):

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

A typical deterministic TFM problem is formulated as an Integer Programming problem. By controlling the flow rate $q_i^k(t)$, delays are minimized, while sector counts are kept below the sector capacity $C_s(t)$.

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

s.t.

$$x_i^k(t + 1) = x_i^k(t) - q_i^k(t) + q_{i-1}^k(t) \tag{2}$$

$$\sum_{(i,k) \in Q_{s_i}} x_i^k(t) \leq C_s(t), \quad \sum_{(0,k) \in A_{arr}} q_0^k(t) \leq C_{arr}(t), \quad \sum_{(n^k,k) \in A_{dep}} q_{n^k}^k(t) \leq C_{dep}(t) \tag{3}$$

$$\sum_{t \in \mathbb{T}} q_0^k(t) = \sum_{t \in \mathbb{T}} q_{n^k}^k(t) = \sum_{t \in \mathbb{T}} f^k(t) \tag{4}$$

$$\sum_{t=T_0^k+T_1^k+\dots+T_i^k}^{T_i^k} q_i^k(t) \leq \sum_{t=T_0^k+T_1^k+\dots+T_{i-1}^k}^{T_i^k-T_{i-1}^k} q_{i-1}^k(t) \tag{5}$$

$$\sum_{t=0}^{T_0^k+T_1^k+\dots+T_{i-1}^k} q_i^k(t) = 0, \quad x_i^k(0) = 0 \tag{6}$$

$$x_i^k(t) \in \mathbb{Z}_+, \quad q_i^k(t) \in \mathbb{Z}_+ \tag{7}$$

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

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 (2)–(7) regulate traffic flow behaviors. Constraints (3) enforce en route and airport capacity constraints. Constraints (4) ensure that the accumulated departures equal to the accumulated arrivals. Constraints (5) show that every flight must dwell in a link for at least T_i^k minutes. Constraints (6) and (7) represent the initial states and integer constraints, respectively. We refer readers to Reference Cao and Sun (2011) for detailed discussions about these constraints.

The solution to the above TFM problem is the optimal traffic flow, as well as the flow control for each route. Specifically, vector $[x_1^k(t), x_2^k(t), \dots, x_{n^k}^k(t)]$ 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 $[x_1^k(t), x_2^k(t), \dots, x_{n^k}^k(t)]$, 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 Sun et al. (2010). After the disaggregation process, the flow controls are translated into delays imposed on individual flights in each sector.

2.2. 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 paper proposes to impose a probabilistic constraint on traffic flow capacities, as follows:

$$\mathbb{P} \left(\begin{array}{l} \sum_{(i,k) \in Q_s} x_i^k(t) \leq C_s(t), \quad \forall i, t \in \mathbb{T} \\ \sum_{(0,k) \in A_{arr}} q_0^k(t) \leq C_{arr}(t), \quad \forall i, t \in \mathbb{T} \\ \sum_{(n^k,k) \in A_{dep}} q_{n^k}^k(t) \leq C_{dep}(t), \quad \forall i, t \in \mathbb{T} \end{array} \right) \geq \alpha \tag{8}$$

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 $\alpha \in (0, 1)$. 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. Since the constraints (3) are all linear, the probabilistic constraint (8) can be simply written as:

$$\mathbb{P}(T(t)x(t) \leq \xi(t)) \geq \alpha \tag{9}$$

where $x(t)$ denotes the vector of the decision variables with the associated coefficient matrix $T(t)$ at time t , and $\xi(t)$ is defined as representing the random vectors at time t . Thus, the TFM optimization under the stochastic airspace capacities can be written as:

$$\min d = \sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} \sum_{1 \leq i \leq n^k} c_i^k x_i^k(t) \tag{10}$$

s.t.

$$x_i^k(t+1) = x_i^k(t) - q_i^k(t) + q_{i-1}^k(t) \tag{11}$$

$$\mathbb{P}(T(t)x(t) \leq \xi(t)) \geq \alpha \tag{12}$$

$$\sum_{t \in \mathbb{T}} q_0^k(t) = \sum_{t \in \mathbb{T}} q_{n^k}^k(t) = \sum_{t \in \mathbb{T}} f^k(t) \tag{13}$$

$$\sum_{t=T_0^k+T_1^k+\dots+T_i^k}^{T_i^k} q_i^k(t) \leq \sum_{t=T_0^k+T_1^k+\dots+T_{i-1}^k}^{T_i^k-T_1^k} q_{i-1}^k(t) \tag{14}$$

$$\sum_{t=0}^{T_0^k+T_1^k+\dots+T_{i-1}^k} q_i^k(t) = 0, \quad x_i^k(0) = 0 \tag{15}$$

$$x_i^k(t) \in \mathbb{Z}_+, \quad q_i^k(t) \in \mathbb{Z}_+ \tag{16}$$

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

The only difference from the deterministic model is that the capacity constraints (3) are replaced with the probabilistic capacity constraint (12). This problem is referred to *chance-constrained TFM optimization*.

Moreover, notice that the above TFM model is a linear model, except for the probabilistic constraint (12). Thus, the stochastic TFM problem can be written in the following concise form:

$$\begin{aligned} \min \quad & d = \sum_{t \in \mathbb{T}} c'x(t) \\ \text{s.t.} \quad & \mathbb{P}(T(t)x(t) \leq \xi(t)) \geq \alpha \\ & Ax \leq b \\ & x(t) \in \mathbb{Z}_+, t \in \mathbb{T} \end{aligned} \tag{17}$$

where c represents the vector of the weight coefficients; A and b are the coefficients corresponding to the original linear constraints.

3. Solving methods for chance-constrained TFM problem

The probabilistic programming indicates that some of the constraints may be violated at a well-controlled, very low chance. Not surprisingly, the probabilistic programming problem is difficult to solve for multiple reasons. First, it becomes too costly to evaluate the joint distribution function as the optimization solver moves toward the optimal solution. Second, the convexity of both the constraints and the feasible set needs to be preserved throughout the optimization. The convexity guarantees the global optimal solution, and only convex-preserved transformations are allowed. Third, the problem is computationally demanding. Thus, our novel method is one of the better solutions available for large-scale problems.

In this section, the brute-force method is introduced based on a previous chance-constrained model (Clare and Richards, 2012), 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 paper introduces a 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.

3.1. The brute-force method

In order to solve the chance-constrained ATFM problem, the brute-force MILP optimization model is proposed in Clare and Richards (2012), which is developed based on the Bertsimas-Stock Patterson model (Bertsimas and Patterson, 1998). 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 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 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 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.

Based on the same idea of the feasible capacity set, the probabilistic constraint (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 matrix for all combinations that can be built, which is an m -dimensional matrix. This m -dimensional matrix

Table 1
Example joint probability.

ξ_1, ξ_2	1	2
1	0.06	0.14
2	0.24	0.56

Table 2
Cumulative example joint probability.

sn_1, sn_2	1	2
1	1	0.7
2	0.8	0.56

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 (8) can be replaced by the following linear constraints:

$$\sum_{(i,k) \in Q_{s_i}} x_i^k(t) \leq \sum_{p_i \in M} \delta(p_i, t) I(p_i, j) \tag{18}$$

$$\sum_{p_i \in M} P(p_i, t) \delta(p_i, t) \geq \alpha \tag{19}$$

$$\sum_{p_i \in M} \delta(p_i, t) = 1 \quad \delta(p_i, t) \in \{0, 1\} \tag{20}$$

Constraint (18) enforces the number of aircraft assigned to each sector to be under the safe limits. Constraint (19) ensures that the chance constraint will be satisfied. Constraint (20) 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 TFM operation.

3.2. Polynomial approximation

3.2.1. Problem definition

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 Prékopa (1988), the author showed that under less restrictive assumptions, the chance-constrained model in Section 2.2 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 Saumard and Wellner (2014).

Definition 1. A function $f(z) \geq 0, z \in \mathbb{R}^m$ is said to be logarithmically concave (in short form, logconcave), 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 $\log f(z)$ is a concave function in \mathbb{R}^m .

Definition 2. A probability measure defined on the Borel sets of \mathbb{R}^m is said to be logarithmically concave (logconcave) 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 1. *If $\xi \in \mathbb{R}^m$ is a random variable, the probability distribution of which is logconcave, then the probability distribution function $F(x) = \mathbb{P}(\xi \leq x)$ is a logconcave function in \mathbb{R}^m .*

The proof of Theorem 1 and the rationale of Definitions 1 and 2 are presented in Shapiro et al. (2014), and this paper omits them. Theorem 1 is applied to the probabilistic constraint of model (17), and if the distribution of ξ is logconcave, we have

$$\begin{aligned} \mathbb{P}(T(t)x(t) \leq \xi(t)) \geq \alpha &\iff 1 - F(T(t)x(t)) \geq \alpha \iff \\ \log(1 - \alpha) \geq \log(F(T(t)x(t))) &\iff \log(F(T(t)x(t))) - \log(1 - \alpha) \leq 0 \end{aligned} \tag{21}$$

Thus, the new model can be written as

$$\begin{aligned} \min \quad & d = \sum_{t \in \mathbb{T}} c'x(t) \\ \text{s.t.} \quad & \\ & g_t(x) = \log(F(T(t)x(t))) - \log(1 - \alpha) \leq 0 \\ & Ax \leq b \\ & x(t) \in \mathbb{Z}_+, t \in \mathbb{T} \end{aligned} \tag{22}$$

With the exception of one constraint, $g_t(x) : \log(F(T(t)x(t))) - \log(1 - \alpha) \leq 0$, model (22) is a linear programming. Although Constraint (21) is nonlinear, it is convex, which makes model (22) a convex program with respect to x . For any feasible point x_0 of model (22), as long as we have the function value at x_0 , i.e., $g_t(x_0)$ and its gradient $\nabla g_t(x_0)$ (subgradient if $g_t(x)$ is nondifferentiable) at x_0 , then a first-order gradient algorithm can be adopted to obtain the optimal solution (Nesterov, 2013).

The assumption that ξ follows a logconcave 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 Van der Vaart (2000)), 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, logconcave. 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 logconcave. There are only a few commonly used distributions that are not logconcave, 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(Liu et al., 2008; Buxi and Hansen, 2011). For generating AAR distributions, the most common approach is the Weather Translation Model for Ground Delay Program Planning (Cunningham et al., 2012; Provan et al., 2011). Recently, a AAR Distribution Prediction Model is proposed based on the Bayesian network model, which could predict the distribution based on the weather forecasts(Cox and Kochenderfer, 2016). There is few literature on en route sector capacity distribution analysis, which is still an open area for future study.

With the logconcave distribution assumption, the formulation (22) is relaxed into a continuous problem. Moreover, if we define $g_t(x) = \log(F(T(t)x(t))) - \log(1 - \alpha) \leq 0$, then the formulation becomes a standard constrained optimization problem as follows:

$$\begin{aligned} \min \quad & f_0(x) = c'x \\ \text{s.t.} \quad & \\ & g_t(x) \leq 0, t \in \mathbb{T} \\ & Ax \leq b \\ & x \geq 0 \end{aligned} \tag{23}$$

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

We want to emphasize that all of the approximation method described in the rest of this section will be based on this standard model (23), and to be concise, all of the indices in this subsection are independent of the former ones in Section 2.

3.2.2. Details of the approximation

The key to solving model (23) is to effectively evaluate the gradient (or subgradient) of $g_t(x)$. 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, 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 with a probability of 1.

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

$$\ell^i \leq x^i \leq u^i, i = 1, \dots, n. \tag{24}$$

We are interested in

$$\nabla g_t(x) := \left[\frac{\partial g_t(x)}{\partial x^1}; \dots; \frac{\partial g_t(x)}{\partial x^n} \right] \tag{25}$$

and each component $\frac{\partial g_t(x)}{\partial x^i} : \mathbb{R} \rightarrow \mathbb{R}$ 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 univariate function with respect to $x^i \in [\ell^i, u^i]$. $g_t^i(x^i)$ is essentially the function $g_t(x)$ with $x^1, \dots, x^{i-1}, x^{i+1}, \dots, x^n$ 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 $g_t^i(x^i), i = 1, \dots, n$ 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.

In this paper, 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. The Bernstein polynomial of a function $\phi(y), y \in [0, 1]$ is

$$B_k(\phi; y) := \sum_{j=0}^k \binom{k}{j} y^j (1-y)^{k-j} \phi(j/k) \tag{26}$$

and

$$B_k(\phi; 0) = \phi(0), B_k(\phi; 1) = \phi(1). \tag{27}$$

Theorem 2 (Bernstein Theorem). *Let $\phi(y)$ be continuous on $[0, 1]$. Then*

$$\lim_{k \rightarrow \infty} B_k(\phi; y) = \phi(y) \tag{28}$$

any point $y \in [0, 1]$ and the limit (28) hold uniformly in $[0, 1]$. That is, given an $\epsilon > 0$, for all large enough k , we have

$$|\phi(y) - B_k(\phi; y)| \leq \epsilon, y \in [0, 1]. \tag{29}$$

The proof is in Davis (1975), and we omit it. This theorem 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. Thus, we discard the idea of directly approximating $g_t^i(x^i)$ by the Bernstein polynomial. Instead, we find the following important result.

Theorem 3. *There exists a sequence of component functions:*

$$\psi_0(y), \psi_1(y), \psi_2(y), \dots, \tag{30}$$

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

The proof is in Phillips and Taylor (1970). Since this result plays the central role of this paper, 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 2 to construct a 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

$$B_k(\phi''; y) = \sum_{j=0}^k \binom{k}{j} y^j (1-y)^{k-j} \phi''(j/k) \tag{31}$$

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 (31) 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

$$p''_k(y) = B_{k-2}(\phi''; y), p'_k(0) = \phi'(0), p_k(0) = \phi(0). \tag{32}$$

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

$$\beta''_{j,k}(y) = y^{j-2}(1-y)^{k-j}, \beta'_{j,k}(0) = \beta_{j,k}(0) = 0. \tag{33}$$

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

$$\beta_{0,k}(y) = \text{sign}[\phi(0)], \beta_{1,k}(y) = x\text{sign}[\phi'(0)]. \tag{34}$$

The relevance of the choice of functions (34) will be seen later. We then have

$$p_k(y) = \sum_{j=0}^k c_j^* \beta_{j,k}(y), \tag{35}$$

where $c_j^* \geq 0$ and $\beta''_{j,k}(y) \geq 0$ on $[0, 1]$. Now, given any $\epsilon > 0$, applying Theorem 2, we have

$$|B_{k-2}(\phi''; y) - \phi''(y)| \leq \epsilon \tag{36}$$

on $[0, 1]$. That is

$$|p''_k(y) - \phi''(y)| \leq \epsilon \tag{37}$$

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

$$\left| \int_0^y (p''_k(t) - \phi''(t)) dt \right| \leq \int_0^y |p''_k(t) - \phi''(t)| dt \leq \epsilon y \leq \epsilon. \tag{38}$$

Using (32), the inequality (38) gives

$$|p'_k(y) - \phi'(y)| \leq \epsilon \tag{39}$$

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

$$|p_k(y) - \phi(y)| \leq \epsilon \tag{40}$$

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

$$\psi_j(y) = \beta_{j,k}(y), \quad 2 \leq j \leq k, \quad \psi_0(y) = \text{sign}[\phi(0)], \quad \psi_1(y) = x\text{sign}[\phi'(0)] \tag{41}$$

where for $j \geq 2$,

$$\beta''_{j,k}(y) = y^{j-2}(1-y)^{k-j} = x^{j-2} \sum_{i=1}^{k-j} (-1)^i \binom{k-j}{i} y^i \tag{42}$$

and we have

$$\beta_{j,k}(y) = y^j \sum_{i=0}^{k-j} (-1)^i \frac{\binom{k-j}{i} y^i}{[(i+j)(i+j-1)]}. \tag{43}$$

□

This theorem 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

$$p_k(y) = \sum_{j=0}^k c_j^* \psi_j(y) \tag{44}$$

of degree k , regardless of the differentiability of $\phi(y)$. As long as all of the coefficients in (44) are non-negative, $c_j^* \geq 0, j = 0, 1, \dots, k$, $p_k(y)$ will be convex. We also need $k + 1$ non-negative coefficients c_0^*, \dots, c_k^* to construct $p_k(y)$. To obtain these coefficients, we need a set of points with coordinates $(y_i, \phi(y_i)), i = 1, 2, \dots, k + 1$. We solve the following problem

$$\min \left\{ \max_{i=1, \dots, k+1} \left\{ \phi(y_i) - \sum_{j=0}^k c_j^* \psi_j(y_i) \right\}, c_j^* \geq 0 \right\} \tag{45}$$

Definition 4. The $p_k(y)$ with coefficients c_0^*, \dots, c_k^* is called the best approximation of degree k .

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

Theorem 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:*

$$\max_{y \in [0,1]} |\phi(y) - p_k(y)| \leq \left(\frac{\pi}{2}\right)^r \frac{|\phi^{(r)}(\omega)|}{[(k-r+2) \dots k(k+1)]}, k \geq r \tag{46}$$

Table 3
Error bounds as the degree k of $p_k(y)$ increases.

k	4	5	6	7	8	9	10	11	12	20	40	50
Error bounds	1	0.67	0.48	0.36	0.28	0.22	0.18	0.15	0.13	0.048	0.012	0.008

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 Cheney (1966). This theorem 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

$$\frac{\left(\frac{\pi}{2}\right)^r \frac{|\phi^{(r)}(\omega)|}{[(k-r+3)\dots(k+1)(k+2)]}}{\left(\frac{\pi}{2}\right)^r \frac{|\phi^{(r)}(\omega)|}{[(k-r+2)\dots k(k+1)]}} = \frac{k-r+2}{k+2} < 1. \tag{47}$$

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.

From Table 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 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 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.

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

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

$$\left| \phi(y) - p_k(y) \right| = \frac{\phi^{(k+1)}(\omega)}{(k+1)!} \prod_{i=1}^{k+1} (y - y_i) \tag{48}$$

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

This proposition is in Stewart (1993, Lecture 20). Since we can apply Theorem 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 $\prod_{i=1}^{k+1} (y - y_i)$ by choosing the Chebyshev nodes on $[0, 1]$, as follows:

$$y_i := \frac{1}{2} - \frac{1}{2} \cos\left(\frac{2i-1}{2k+2}\pi\right), \quad i = 1, \dots, k + 1. \tag{49}$$

The technical detail regarding the minimization of the approximation error by adopting Chebyshev nodes is in (Stewart, 1993).

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

- Step 1. Determine the overall error bound $\epsilon > 0$.
- Step 2. Choose the degree k .
- Step 3. Calculate $k+1$ Chebyshev nodes y_i and coordinates $(y_i, \phi(y_i)), i=1, \dots, k+1$.
- Step 4. Solve the model (44) for the coefficients c_0^*, \dots, c_k^* and construct $p_k(y)$.
- Step 5. Use $p'_k(y)$ as an approximation of $\phi'(y)$.

4. Algorithms and computational complexity

4.1. 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 Nesterov (2013) 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 (23) can be transformed into the parametric max-type function:

$$f(d; x) = \max\{f_0(x) - d; g_t(x), t \in \mathbb{T}\}, \quad d \in \mathbb{R}^1, x \in Q. \tag{50}$$

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:

$$f(d; \bar{x}; x) = \max_{t \in \mathbb{T}} \{f_0(\bar{x}) + \langle f'_0(\bar{x}), x - \bar{x} \rangle - d; g_t(\bar{x}) + \langle g'_t(\bar{x}), x - \bar{x} \rangle\}. \tag{51}$$

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

$$f_\gamma(d; \bar{x}; x) = f(d; \bar{x}; x) + \frac{\gamma}{2} \|x - \bar{x}\|^2 \tag{52}$$

$$f^*(d; \bar{x}; \gamma) = \min_{x \in Q} f_\gamma(d; \bar{x}; x) \tag{53}$$

$$x_f(d; \bar{x}; \gamma) = \arg \min_{x \in Q} f_\gamma(d; \bar{x}; x) \tag{54}$$

$$g_f(d; \bar{x}; \gamma) = \gamma(\bar{x} - x_f(d; \bar{x}; \gamma)). \tag{55}$$

where $g_f(d; \bar{x}; \gamma)$ is the constrained gradient mapping of the problem (23).

The main algorithm for the chance-constrained problem is shown in Algorithm 1.

Algorithm 1 Chance-constrained TFM optimization based on polynomial approximation.

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

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

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

for the j th internal iteration:

Approximate $g_t(y_{r,j})$ and $\nabla g_t(y_{r,j})$ by the method in Section 3.2

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 $j^*(r) = \arg \min_{0 \leq j \leq j(r)} (f^*(d_r; x_{r,j}; L))$ 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 $f^*(d; \bar{x}; 0)$

$r=r+1$

4.2. 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 $\nabla g_t(x)$ 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 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 (45) needs to calculate $\psi_0(x^i), \dots, \psi_k(x^i)$. 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 $\psi_0(x^i), \dots, \psi_k(x^i)$ 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(45).

Model (45) 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 Boyd and Vandenberghe (2004). With the obtained c_0, \dots, 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 Nesterov (2013), we can get the following result:

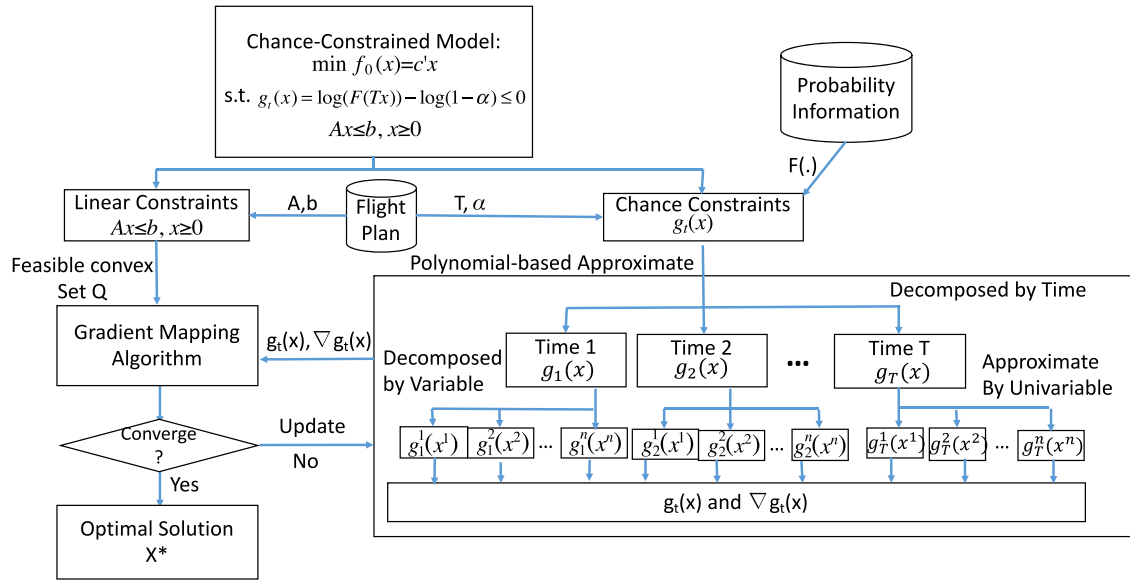


Fig. 2. Parallel framework.

Proposition 2. The gradient mapping method takes at most

$$\frac{1}{\ln(2(1 - \kappa))} \ln \frac{d_0 - d^*}{(1 - \kappa)\epsilon} \tag{56}$$

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 Nesterov (2013). In the proof, both κ and $d_0 - d^*$ are well-bounded values. Therefore, the iteration value of (56) 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.

5. Parallel computing framework

Recall that the construction of the polynomial approximation for each individual marginal function $g_t^i(x^i)$ is independent. Therefore, the approximation process for the chance constraints is highly suitable for performing parallel computing. A customized parallel computing framework for the chance-constrained TFM optimization, based on the polynomial approximation method, is shown in Fig. 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, the feasible convex set Q and the first-order information ($g_t(x)$ and $\nabla g_t(x)$) 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 parallel 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 problems. 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. After solving the Tn independent problems in parallel, the results can be gathered to provide first-order information ($g_t(x)$ and $\nabla g_t(x)$) 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 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.

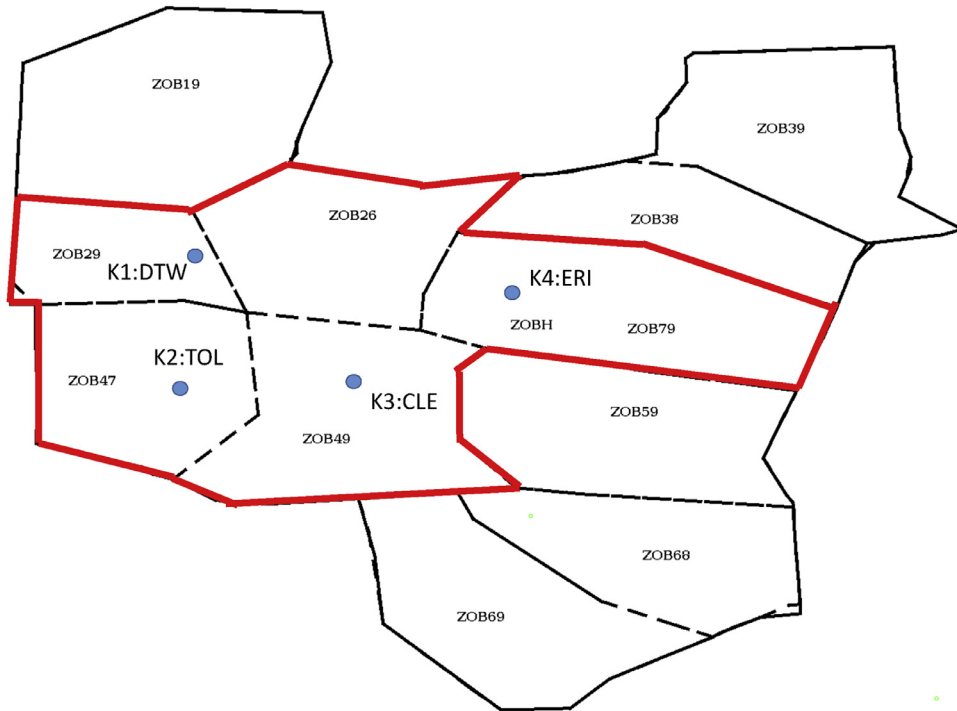


Fig. 3. Small-sized example.

Table 4
Flight schedule.

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

Table 5
Sector capacity distribution.

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

6. Simulation results

6.1. Model validation with a small-sized example

6.1.1. Example setup

As stated in Section 3.1, 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 Fig. 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, 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 5. The corresponding cumulative probability ($P(C \geq sn_j)$) is shown

Table 6
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

Table 7
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

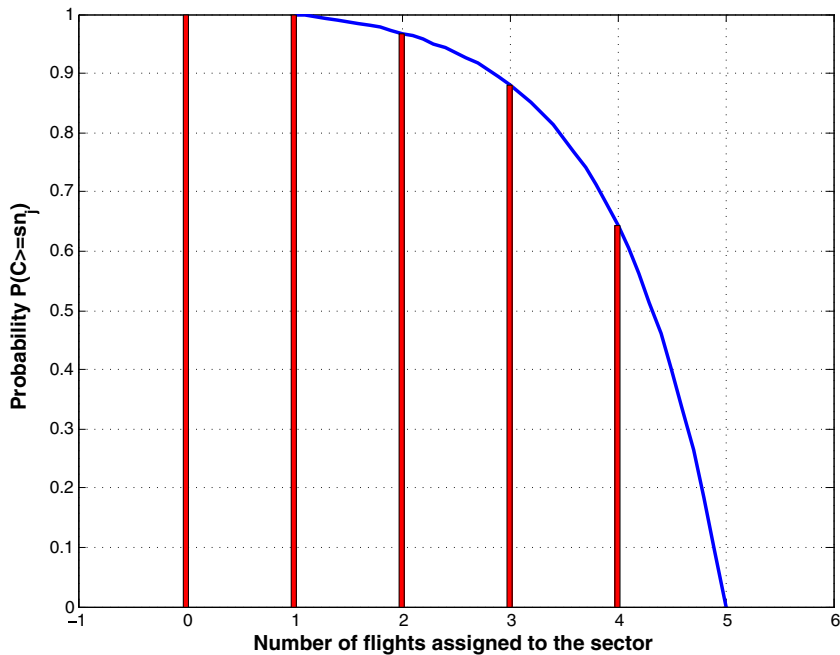


Fig. 4. Cumulative probability.

in Table 6, 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 change, and the method to form the feasible combination set of the MILP model is the same.

6.1.2. Result of MILP

The result of MILP is solved with the chance constraints’ significance level of 0.8 ($\alpha = 0.8$). The result is then collected to check the feasibility of the chance constraint, which is shown in Table 7. 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 solution. Later, we will use this accurate optimal solution to evaluate the approximation-based method.

6.1.3. Result of approximation-based approach

To perform the approximation-based approach, a log-concave continuous probability distribution is provided, as shown in Fig. 4, which has the exact cumulative probability ($P(C \geq sn_j)$) with the discrete one in Table 5. Therefore, the comparison is meaningful with the same probability information. With the same significance level of 0.8 ($\alpha = 0.8$), the result, based on

Table 8
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						

Table 9
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						

the approximation-based method, is shown in Table 8, which are real numbers before the rounding process. It is clear that the 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 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 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 (Gurobi, 2014), where the real-valued optimal solution is provided as an initial point. As shown in Table 9, 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 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, which is also confirmed by large-scale problem simulations in Section 6.2.

6.2. Large-scale ATFM simulation

This subsection presents a large-scale ATFM optimization, employing the proposed parallel computing framework. The traffic data are extracted from the ASDI, which provides historical traffic data, as well as flight plans. A 2-h NAS-wide instance is used, which represents the high-traffic period of a day, with 2326 paths and 3054 flights involved. The same log-concave probability distribution in Section 6.1 is adopted and adjusted, according to the real-sector capacity (usually 10 to 15). The optimization is performed on a small Spark cluster with six nodes, where Spark is a distributed computing platform (Karau et al., 2015). 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.

To further 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 are extracted from the 2-h NAS-wide instance above. Fig. 5 shows the relative objective error between the approximation-based approach and the accurate optimal solution. The error gap is kept below 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

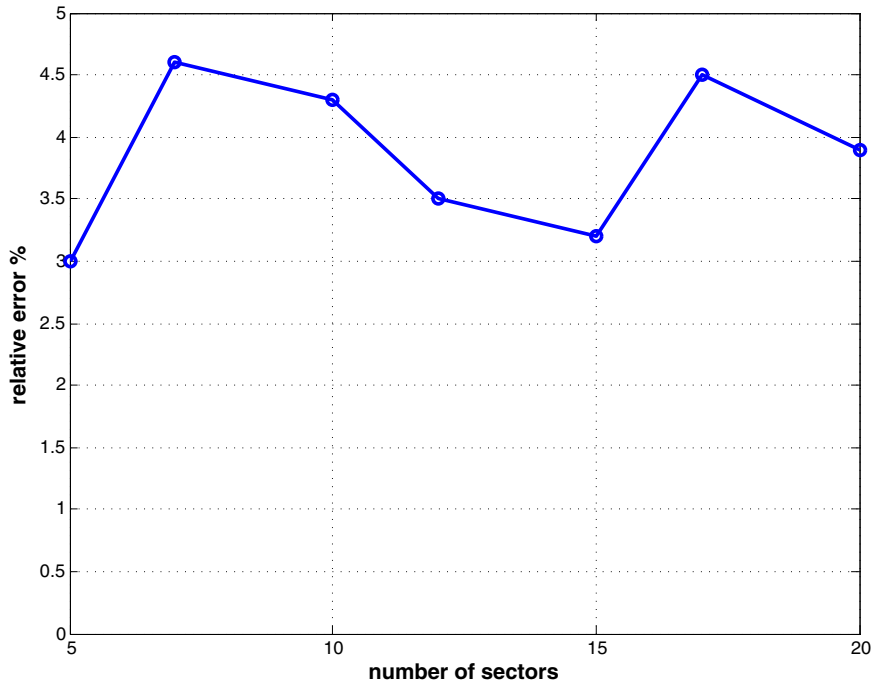


Fig. 5. Relative error with different ATFM problem sizes.

up to 20 sectors. 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.

The running time of the optimization with different paralleling configurations is demonstrated in Fig. 6. As a tuning parameter to control the concurrency level, the maximum number of executors allowed on a node 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 nodes. 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 nodes are launched. The speedup is also not linear by the fact that it is more and more difficult to achieve further speedup as more nodes are deployed. The runtime is reduced from 241 min with 1 nodes and 1 executor to 25 min with 6 nodes 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 parallel computing programming. Overall, it is clear that the parallel computing framework, indeed, improves the computational efficiency of the polynomial approximation-based approach.

7. Conclusions

This paper introduces a novel polynomial approximation-based chance-constrained optimization method to address uncertainty in ATFM, which could provide a computationally efficient algorithm. 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 is 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. Moreover, a distributed computing framework is designed based on the massive independent approximation processes in the polynomial approximation-based approach, which is demonstrated with the ability to improve computational efficiency. 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.

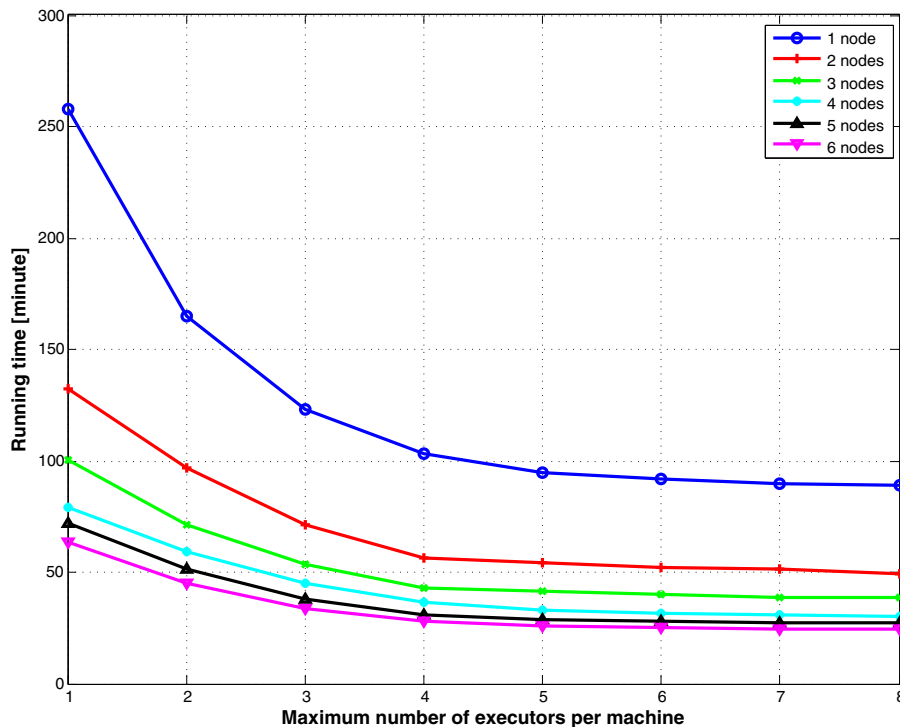


Fig. 6. Runtime of the NAS-wide AFTM optimization with the parallel computing framework.

References

- Ball, M.O., Hoffman, R., Odoni, A.R., Rifkin, R., 2003. A stochastic integer program with dual network structure and its application to the ground-holding problem. *Oper. Res.* 51 (1), 167–171.
- Bertsimas, D., Lulli, G., Odoni, A., 2008. The air traffic flow management problem: an integer optimization approach. In: *Integer programming and combinatorial optimization*. Springer, pp. 34–46.
- Bertsimas, D., Lulli, G., Odoni, A., 2011. An integer optimization approach to large-scale air traffic flow management. *Oper. Res.* 59 (1), 211–227.
- Bertsimas, D., Patterson, S.S., 1998. The air traffic flow management problem with enroute capacities. *Oper. Res.* 46 (3), 406–422.
- Bertsimas, D., Patterson, S.S., 2000. The traffic flow management rerouting problem in air traffic control: a dynamic network flow approach. *Transp. Sci.* 34 (3), 239–255.
- Boyd, S., Vandenberghe, L., 2004. *Convex Optimization*. Cambridge University Press.
- Buxi, G., Hansen, M., 2011. Generating probabilistic capacity profiles from weather forecast: A design-of-experiment approach. In: *Proc. of USA/Europe Air Traffic Management Research & Development Seminar*.
- Cao, Y., Sun, D., 2011. Link transmission model for air traffic flow management. *J. Guidance Control Dyn.* 34 (5), 1342–1351.
- Cheney, E., 1966. *Introduction to Approximation Theory*. Chelsea, New York.
- Clare, G., Richards, A., 2012. 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*. IRIT Press, pp. 20–26.
- Cox, J., Kochenderfer, M.J., 2016. Probabilistic airport acceptance rate prediction. In: *AIAA Modeling and Simulation Technologies Conference*, p. 0165.
- Cunningham, J., Cook, L., Provan, C., 2012. The utilization of current forecast products in a probabilistic airport capacity model.
- Daganzo, C.F., 1994. The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transp. Res. Part B* 28 (4), 269–287.
- Daganzo, C.F., 1995. The cell transmission model, part ii: network traffic. *Transp. Res. Part B* 29 (2), 79–93.
- Davis, P.J., 1975. *Interpolation and Approximation*. Courier Corporation.
- Division, V.C.A.A., 2000. *Aircraft Situation Display to Industry: Functional Description and Interface Control Document*. Technical Report, Report no. AS-DI-FD-001. Cambridge, Massachusetts.
- Gupta, S., Bertsimas, D. J., 2011. 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*.
- Gurobi, O. I., 2014. *Gurobi optimizer reference manual*, (accessed 27/03/14) [Online]. Available: www.gurobi.com.
- Karau, H., Konwinski, A., Wendell, P., Zaharia, M., 2015. *Learning Spark: Lightning-Fast Big Data Analysis*. O'Reilly Media, Inc.
- Liu, P.-c.B., Hansen, M., Mukherjee, A., 2008. Scenario-based air traffic flow management: from theory to practice. *Transp. Res. Part B* 42 (7), 685–702.
- Menon, P.K., Sweriduk, G.D., Bilimoria, K.D., 2004. New approach for modeling, analysis, and control of air traffic flow. *J. Guidance Control Dyn.* 27 (5), 737–744.
- Mukherjee, A., Hansen, M., 2007. A dynamic stochastic model for the single airport ground holding problem. *Transp. Sci.* 41 (4), 444–456.
- Mukherjee, A., Hansen, M., 2009. A dynamic rerouting model for air traffic flow management. *Transp. Res. Part B* 43 (1), 159–171.
- Nesterov, Y., 2013. *Introductory Lectures on Convex Optimization: A Basic Course*, 87. Springer Science & Business Media.
- Odoni, A.R., 1987. The flow management problem in air traffic control. In: *Flow control of congested networks*. Springer, pp. 269–288.
- Phillips, G., Taylor, P., 1970. Approximation of convex data. *BIT Numer. Math.* 10 (3), 324–332.
- Prékopa, A., 1988. Numerical solution of probabilistic constrained programming problems. *Numer. Tech. Stochastic Optim.* 123–139.
- Provan, C.A., Cook, L., Cunningham, J., 2011. A probabilistic airport capacity model for improved ground delay program planning. In: *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th. IEEE*, pp. 2B6–1.
- Richetta, O., Odoni, A.R., 1994. Dynamic solution to the ground-holding problem in air traffic control. *Transp. Res. Part A* 28 (3), 167–185.

- Saumard, A., Wellner, J.A., 2014. Log-concavity and strong log-concavity: a review. *Stat. Surv.* 8, 45.
- Shapiro, A., Dentcheva, D., et al., 2014. *Lectures on Stochastic Programming: Modeling and Theory*, 16. SIAM.
- Sridhar, B., Grabbe, S.R., Mukherjee, A., 2008. Modeling and optimization in traffic flow management. *Proc. IEEE* 96 (12), 2060–2080.
- Stewart, G., 1993. *Afternotes on Numerical Analysis*. University of Maryland at College Park.
- Sun, D., Bayen, A.M., 2008. Multicommodity eulerian-lagrangian large-capacity cell transmission model for en route traffic. *J. Guidance Control Dyn.* 31 (3), 616–628.
- Sun, D., Clinet, A., Bayen, A.M., 2011. A dual decomposition method for sector capacity constrained traffic flow optimization. *Transp. Res. Part B* 45 (6), 880–902.
- Sun, D., Sridhar, B., Grabbe, S.R., 2010. Disaggregation method for an aggregate traffic flow management model. *J. Guidance Control Dyn.* 33 (3), 666–676.
- Van der Vaart, A.W., 2000. *Asymptotic Statistics*, 3. Cambridge University Press.
- Volpe, N.T.C., 2000. Enhanced Traffic Management System (ETMS). Technical Report, Rep. VNTSC-DTS56-TMS-002. United States Department of Transportation, Cambridge, Massachusetts.
- Wei, P., Cao, Y., Sun, D., 2013. Total unimodularity and decomposition method for large-scale air traffic cell transmission model. *Transp. Res. Part B* 53, 1–16.