Received March 12, 2020, accepted March 27, 2020, date of publication March 31, 2020, date of current version April 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984513

Stochastic Runway Scheduling Problem With Partial Distribution Information of Random Parameters

MING LIU[1], **(Senior Member, IEEE), BIAN LIANG**[1], **MAORAN ZHU**[1], **AND CHENGBIN CHU**[2,3]

[1]School of Economics and Management, Tongji University, Shanghai 200092, China
[2]School of Economics and Management, Fuzhou University, Fuzhou 350108, China
[3]ESIEE Paris, Université Gustave Effeil, 93162 Noisy-le-Grand Cedex, France

Corresponding author: Chengbin Chu (chengbin.chu@gmail.com)

This work was supported in part by the National Science Foundation of China (NSFC) under Grant 71531011, Grant 71771048, and Grant 71571134, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT Air traffic flow management is one of the most important operations in terminal airports heavily relying on advanced intelligence transportation techniques. This work considers a two-stage runway scheduling problem given a set of flights with uncertain arrival times. The first-stage problem is to identify a sequence of aircraft weight classes (e.g., Heavy, Large and Small) that minimizes runway occupying time (i.e., makespan). Then the second-stage decision is dedicated to scheduling the flights as punctually as possible after their arrival times realized, which translates into determining a sequence of flights for each aircraft category such that the total deviation time imposed on the flights is minimized. Instead of an exactly known probability distribution, information on uncertain parameters is limited (i.e., ambiguous), such as means, mean absolute deviations and support set of random parameters derived from historical data. Under this information on the random parameters, an ambiguous mixed-integer stochastic optimization model is proposed. For such a problem, we approximately construct a worst-case discrete probability distribution with three possible realizations per random parameter, and adopt a hybrid sample average approximation algorithm in which genetic algorithms are used to replace commercial solvers. To illustrate the effectiveness and efficiency of the proposed model and algorithm, extensive numerical experiments are carried out.

INDEX TERMS Runway scheduling, stochastic optimization, ambiguity set, approximation, genetic algorithm.

I. INTRODUCTION

The demand of passengers on air transport is increasing with the growth of national economy. In China, for example, the passenger traffic increases from 138.3 millions of 2005 to 488.0 millions of 2016 with 12.22% average annual growth rate [1]. Globally, the annual travelers total of 2018 was up by about 6.5% compared to 2017, and the traffic volume is expected to reach over 6.4 billion by 2030 [2]. The ever growing demand of air transport is growingly increasing the pressure on airport operations due to restricted airdrome capacity and air traffic controller's ability. The ever growing traffic bring about high congestion, the main reason of flight delay, in the terminal area due to the limited airport resources.

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Chen.

To improve the efficiency of transportation system, some intelligence support projects have been developed, such as routing planning [3], ride-on-demand [4] and car GPS [5], [6] in vehicle traffic, Next Generation Air Transportation System (NextGen) [7] and Single European Sky Air Traffic Management (ATM) Research (SESAR) [8] in the domain of aviation optimization. Recently, ATM Technology Demonstration-1 (ATD-1) [9] project was performed by the NASA Ames Research Center, and Time-Based Flow Management (TBFM) is one of the technologies embedded in the ATD-1, which is used operationally for scheduling arrivals to major airports based on airport conditions, airport capacity, required spacing, and weather condition [10]. As an integral part of operational optimization of the TBFM, the *runway scheduling problem* (RSP) is defined as follows. The service time on a runway for each plane, in a given set of flights,

TABLE 1. Separation requirements (in seconds).

		Trailing aircraft		
		Small	Large	Heavy
Leading aircraft	Small	82	69	60
	Large	131	69	60
	Heavy	196	157	96

whether it is arriving, leaving or a mixed-mode, should be determined within a specified planning horizon. Meanwhile, a predetermined time window and separation requirements between the leading and following aircraft should be satisfied. In line with [11], the minimum separation time matrix associated with the aircraft weight class (i.e., aircraft category) is given in Table 1.

A great majority of operation circumstances in the existing literature are assumed deterministic (i.e., flights' arrival times are known in prior). However, some researchers argue that there are considerable uncertainty in the RSP (e.g. [12], [13]). For example, fluctuations in weather, complexity of airport surface operations and human factors involving in air traffic management all play important roles to the punctuality of flights. So inevitable is it for decision models that implicitly or explicitly take the uncertainties into account to develop more adoptable and practical scheduling schemes for the TBFM.

To handle the RSP with uncertain characteristics, stochastic programming (SP) approaches (or recourse models) are usually adopted. Specific probability distribution models are always used to portray the randomness of uncertain parameters, in the belief that historical data is abundant and regularity. It is generally assumed that the distribution pattern of arrival time is exactly known and the uncertainty is described in scenario-based way. The strategy to implement the two-stage stochastic optimization approach in a runway scheduling comparing with the deterministic method is shown in Figure 1. The two-stage optimization approach for runway scheduling deals with the tactical level decisions that generate a robust aircraft weight class sequence (i.e., here-and-now decisions) by using some possible scenarios while actual arrival time is unknown. After the uncertain parameter of arrival time is revealed we are allowed to take recourse actions (e.g., real scheduling of specific flights) with the aim of minimizing deviation time. Furthermore, the two-stage optimization approach planned ahead of flights arriving contributes to relieving decision making workload during real-time scheduling.

In fact, however, practical applications in runway scheduling may add one difficulty to general two-stage stochastic programming models. Specifically, there is so limited uncertain information available in a short period (e.g., the history data of the same flight within one season) so that it is impossible to know the specific probability of random parameters. In addition, though large amounts of data are available, there are even no suitable probability distribution models to fit these information well due to the lack of closed

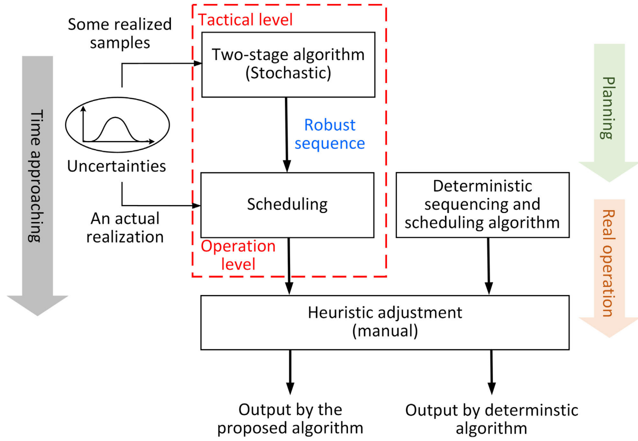


FIGURE 1. An illustration of the application to runway scheduling by two-stage optimization approach.

form expression. For these reasons, we propose an ambiguous two-stage stochastic programming model involving integer decision variables, in which the probability distribution on random parameters is partially known. There is no specific probability distribution function to express the randomness of parameters. In the stochastic programming (SP) literature ambiguous recourse models are called *minmax* problems, whereas they are called *distributionally robust optimization* in the robust optimization (RO) problems [14]. No matter the optimization methods used, the goal of ambiguous recourse models is to find a robust solution with minimizing the expected cost of *worst-case* on all possible probability distributions of random parameters.

Motivated by the more practical approach for practitioners, in this work, we consider the case of ambiguous distribution characterized by partial distribution information on flight's arrival time, and concentrate on solving the ambiguous stochastic runway scheduling problem (ASRSP) by using heuristic algorithm to boost the TBFM technique used in intelligence air transportation system. The efficient scheduling of landing aircraft on a single runway here is mainly discussed. The objective of the ASRSP is twofold, the first goal is to minimize runway occupying time, and the second goal is to minimize the total deviation time for all flights in second stage. The ambiguity set we used contains information on means, mean absolute deviations (MAD) and support set of the random parameters. These information can be estimated from historical data.

To our best knowledge, it is the first time to consider an ambiguous stochastic optimization problem with random aircraft arrival time in RSPs. The main contributions of this work are summarized as follows:

1) We address the ambiguous two-stage stochastic programming runway scheduling problem which aims to improve the runway operations with partial distribution information available. In this work, a worst-case discrete distributions of uncertain parameters is approximately constructed in case of partial

distribution information. The proposed model in favor of improving runway scheduling effectiveness by comparing with the current practice.

2) We develop a hybrid sample average approximation (HSAA) solution algorithm for resolving the nature of uncertainty. In contrast with general SAA algorithm, the improvement includes a nested genetic algorithm GA) used to replace commercial solvers that fail to solve large problems and clustering sampling methods applied to choose more representative scenarios. The HSAA can solve problem with up to 60 flights and 50 scenarios.
3) Extensive numerical experiments are carried out to illustrate the computing efficiency and effectiveness of the proposed solution algorithm by comparing with standard solvers. Comparisons between random sampling and clustering sampling are performed to demonstrate the effectiveness of HSAA algorithm.

The remainder of this article is organized as follows. in Section II, a literature review of the RSP is given. The mathematical formulation and solution approach for the ASRSP are given in Section III and IV respectively. In Section V, we carry out a lot of numerical experiments to demonstrate the performance of the proposed model and algorithm. Finally, Section VI concludes this work and indicates some related directions for future research.

II. LITERATURE REVIEW

There are two mainstream modeling methods for RSPs. One popular approach is heuristic or meta-heuristic designed to meet the stringent computation time feature required by the real-time nature. Another is exact-based perspective, which consists of two classic types: dynamic programming (DP) and mixed-integer programming (MIP) [15].

It is general to interpret the RSP as a classic machine scheduling problem with sequence-dependent set-up times and makespan/total tardiness objective function. In the context of RSP, runways are equivalent to machines, aircraft equal to jobs and time separations between consecutive airplanes on the same runway correspond to the set-up times. The first MIP model of the RSP on a single runway is presented by [16], and a branch-and-bound (B&B) algorithm is developed to solve the single RSP. Then Beasley *et al.* [17] generalizes the single-runway RSP to multiple runways scenario, and this extension is the most cited MIP model to date. Considering the speed of fuel consumption and safety requirements, the RSP formulation is further extended by [18] via taking runway-dependent time windows and separation times constraints into account. Focusing on the implicit homogeneity assumption on inbound aircraft, Briskorn and Stolletz [19] proposes a modification of the MIP formulation that explicitly takes aircraft weight class into consideration. Recently, a time-indexed formulation is developed by [15] to achieve a good trade-off between solutions quality and computing effort. In addition, some other surrounding features and operation characteristics are considered in the enhanced RSP, such as runway scheduling problem with holding pattern [20], integrated airport taxiway routing and runway scheduling [21], rolling planning horizon runway scheduling [20] and mixed-mode RSP [22]. In the domain of model's solving efficiency, branch-and-price (B&P) algorithm, dynamic constraint generation algorithm, heuristically pruning rules, polynomial-time heuristic algorithm are developed [23].

For DP approaches, Dear and Sherif [24] firstly introduces constrained position shifting (CPS) constraints into aircraft sequencing and scheduling problem for high density terminal airports. Then, Balakrishnan and Chandran [11] develops a unified framework for the RSP with CPS constraints. To solve the RSP efficiently with different weight classes, limit time windows and multiple runways characteristics, Lieder *et al.* [25] presents a DP algorithm which uses a new dominance criterion that eliminates states from the state space. Similarly, considering the additional diagonal separation constraints in safety requirement, Lieder and Stolletz [26] presents a DP approach for the mixed-mode aircraft scheduling. However, in the above endeavors, the optimization objective is always single. The interests of different stakeholders are considered simultaneously by [27] and [28] with a multi-objective DP algorithm.

Though the MIP and DP approaches are prevailing for the RSP, the computing times are too prolonged to the real-time application. In terms of the dilemma, there are some scholars dedicating to designing heuristic/metaheuristic algorithms with high computational efficiency for real-time scheduling. Beasley *et al.* [29] and Atkin *et al.* [30] respectively employ a population heuristic algorithm and a hybrid metaheuristic approach to aid the runway scheduling at London Heathrow Airport. Pinol and Beasley [18] designs two heuristic algorithms, the so-called scatter search and bionomic algorithm for the multi-runway case of the static RSP. Recently, a hybrid metaheuristic algorithm based on tabu search and variable neighbourhood search is developed by [31], which is utilized to solve a integrated aircraft scheduling and routing problem. Simulated annealing (SA) method is utilized by [2] to delivers quick schedules for a mixed-mode RSP considering the CPS constraints and the dynamic nature of airport.

Uncertain factors, as opposed to the deterministic operational environment, existing in the runway scheduling operations are gradually attracting the attention of interested researchers. Such as, the robust RSP [23], [32], [33], and two-stage stochastic decision procedures [12]. However, the probability distribution of uncertain parameters, in [12], is assumed exactly known, and the scale of problem that can be solved is small. There are few works to tackle the assumption that the distribution of uncertain parameters is unknown in the RSPs. In other research domains, to enhance the robustness of the solutions under the uncertain probability distribution of random parameters, a class of distributionally robust models (DRM) is proposed. DRM describes the uncertainty of input parameters by an ambiguity set consisting of partially distribution information. The ambiguity distribution set is different according to the known parameter information. The existing methods for constructing the ambiguity set of distribution includes: exact first and second moments, directional deviations and mean absolute deviation (MAD).

The computation using the exact algorithm in the two-stage optimization is costly as the complexity of the computation increases with the number of scenarios. Moreover, the computing time required increases dramatically with the size of the model due to the RSP is NP-hard problem [34]. For these reasons, several meta-heuristic approaches have been recently proposed in the RSP model, such as, Simulated Annealing (SA) [2], Genetic Algorithm [18], [29], Artificial Bee Colony (ABC) algorithm [23].

III. PROBLEM FORMULATION

In this section, we firstly introduce an ambiguity set depending on the partially known descriptive statistics of uncertain arrival time and establish the two-stage mathematical formulation of ASRSP. Then we present the approximate worst-case discrete probability distribution for the ASRSP in accordance with given information.

A. AMBIGUOUS TWO-STAGE STOCHASTIC MATHEMATICAL MODEL WITH PARTIAL DISTRIBUTION INFORMATION

Instead of using a exactly known probability distribution (e.g., the inter-arrival time is assumed as exponential distribution), in our work we relax this assumption that the probability distribution of random parameter is exactly known. It is more intuitive for practitioners that the distribution information of aircraft arrival time is partial known and it belongs to one ambiguity set $\mathbb{F}$ containing all possible distributions $\mathbb{P}$.

These descriptive statistical information consist of mean value μ, mean absolute deviation (MAD) d of random parameter and support set $[a, b]$. The reasons why we use these descriptive statistics to construct the ambiguity set not only is these data can be easily collected from tower control department of airport, but also we can, under this information on the random parameters, use a result of of Ben-Tal and Hochman [35] to prove that the approximate worst-case distributions are discrete with at most three possible realizations per random parameter.

Let $e_i, i = 1, 2, \ldots, |I|$ denotes the arrival time of aircraft i, where I is the set of all flights. The partial distribution information on aircraft arrival time e_i constituting the ambiguity set are detailed as follows:

1) Mean values: $u_i = \mathbb{E}_{\mathbb{P}}[e_i]$, μ_i denotes the mean value of each flight $i \in I$.
2) Mean absolute deviations (MAD): $d_i = \mathbb{E}_{\mathbb{P}}[|e_i - \mu_i|]$, d_i denotes the MAD of each flight $i \in I$.
3) Support sets: $[a_i, b_i]$, a_i and b_i respectively represent the lower and upper bound of e_i, where $i \in I$.
4) Covariance between e_i and e_j: $cov_{i,j} = \mathbb{E}_{\mathbb{P}}\left[(e_i - \mu_i)(e_j - \mu_j)\right]$, where $i, j \in I$.

In the literature, such as Balakrishan and Chandran [11] and Solak et al. [12], the schedules are generated assuming a Poisson arrival process corresponding to exponential inter-arrival times. The Poisson arrival process means that flight arrivals occur independently in the disjoint time interval. Due to our work is trying to extend the work of [12], we adopt the independent assumption throughout the paper.

Assumption 1: The estimated arrival times of all flights in a time horizon are independent, that is, the covariance matrix of random arrival time e_i is a diagonal matrix.

Based on the above four blocks, and independent assumption of flight arrival time, the ambiguity set $\mathbb{F}$ is defined in the following.

$$\mathbb{F} = \left\{ \mathbb{P} \left| \begin{array}{ll} \mathbb{E}_{\mathbb{P}}[e_i] = \mu_i, & \forall i \in I \\ \mathbb{E}_{\mathbb{P}}[e_i - \mu_i] = d_i, & \forall i \in I \\ \text{Prob}_{\mathbb{P}}\left(e_i \in [a_i, b_i]\right) = 1, & \forall i \in I \\ e_i \perp e_j, & \forall i \neq j \end{array} \right. \right\} \tag{1}$$

where $e_i \perp e_j$ means that e_i and e_j are stochastically independent.

In line with [12], although the unit of measurement for the runway occupancy time and total deviation time is identical, the impact of these values to airport operations is different. The airport emphasizes on minimizing the runway occupying time (i.e., makespan), but airlines are pursuing the shortest deviation time. Therefore, it is better to use economic cost instead of time to assess the costs of runway occupancy and deviation time of flights in an economical way.

Input Parameters:

- K: set of aircraft weight classes of all landing flights, indexed by k, l.
- I: set of landing flights that can be scheduled in a given planning time, indexed by i.
- P: set of positions in the flight landing sequence, indexed by p, $p \in \{1, 2, \ldots, |P|\}$, where $|P| = |I|$.
- I_k: the set of flights belong to aircraft weight class k, $k \in K$.
- n_k: the number of landing flights belong to aircraft class k.
- $d_{k,l}$: the safety separation requirements between weight class of leading and trailing aircraft, $k, l \in K$.
- e_i: the estimated arrival time of flight i, $i \in I$.
- τ_i: the earliest arrival time of flight i, $i \in I$ and $\tau_i \leq e_i$.
- M: a big enough number.
- λ: the slope of the runway utilization cost function.
- λ_i^+: the slope of cost function when flight i arrived too delay, $i \in I$.
- λ_i^-: the slope of cost function when flight i arrived too early, $i \in I$.
- η: the intercept of the runway utilization cost function.
- η_i^+: the intercept of cost function when flight i arrived too delay, $i \in I$.
- η_i^-: the intercept of cost function when flight i arrived too early, $i \in I$.

Decision Variables:

- $x_{p,k}$: (first-stage variable) = 1 if aircraft class k is assigned to position p, = 0 otherwise, $p \in P$, $k \in K$.

c_p: (first-stage variable) the cumulative separation times at position p, $p \in P$.
g: (first-stage variable) the runway occupying cost.
$y_{p,i}$: (second-stage variable) = 1 if flight i is assigned to position p, = 0 otherwise, $i \in I, p \in P$.
t_p: (second-stage variable) the time of position p, $p \in P$.
t_i: (second-stage variable) the landing time of flight i, $i \in I$.
θ_i^+: (second-stage variable) the delay time of flight i, $i \in I$.
θ_i^-: (second-stage variable) the early time of flight i, $i \in I$.
h_i^+: (second-stage variable) the cost of delay time.
h_i^-: (second-stage variable) the cost of early time.

Given the ambiguity set and the notations, the formulation of two-stage ambiguous stochastic RSP can be described as follows:

$$\text{(ASRSP)} \quad \min \; g + \sup_{\mathbb{P}\in\mathbb{F}} \mathbb{E}_{\mathbb{P}}[Q(x,e)] \tag{2a}$$

$$\text{subject to} \sum_{k\in K} x_{p,k} = 1, \quad \forall p \in P \tag{2b}$$

$$\sum_{p\in P} x_{p,k} = n_k, \quad \forall k \in K \tag{2c}$$

$$d_{k,l}(x_{p,k} + x_{p+1,l} - 1) \le c_{p+1} - c_p, \quad \forall p \in P\backslash\{|P|\}, \quad \forall k,\ l \in K \tag{2d}$$

$$\lambda c_{|P|} + \eta \le g \tag{2e}$$

$$x_{p,k} \in \{0,1\}, \quad 0 \le c_p, \quad \forall p \in P, \quad k \in K \tag{2f}$$

where $\mathbb{F}$ represents an ambiguity set for probability distribution $\mathbb{P}$, the second-stage recourse function $Q(x, e)$ for evaluating the cost of deviation time is defined as a function of the first-stage variable x and random parameter e. When flight arrival time vector e is revealed, the second-stage problem is given as follows:

$$Q(x,e) = \min \sum_{i\in I} (h_i^+ + h_i^-) \tag{2g}$$

$$\text{subject to} \sum_{i\in I_k} y_{p,i} = x_{p,k}, \quad \forall p \in P, \quad k \in K \tag{2h}$$

$$\sum_{p\in P} y_{p,i} = 1, \quad \forall i \in I \tag{2i}$$

$$\tau_i y_{p,i} \le t_p, \quad \forall p \in P,\ i \in I \tag{2j}$$

$$c_{p+1} - c_p \le t_{p+1} - t_p, \quad \forall p \in P\backslash\{|P|\} \tag{2k}$$

$$t_p - M(1 - y_{p,i}) \le t_i, \quad \forall p \in P, \quad i \in I \tag{2l}$$

$$t_i \le t_p + M(1 - y_{p,i}), \quad \forall p \in P, \quad i \in I \tag{2m}$$

$$\theta_i^+ - \theta_i^- = t_i - e_i, \quad \forall i \in I \tag{2n}$$

$$\lambda_i^+ \theta_i^+ + \eta_i^+ \le h_i^+, \quad \forall i \in I \tag{2o}$$

$$\lambda_i^- \theta_i^- + \eta_i^- \le h_i^-, \quad \forall i \in I \tag{2p}$$

$$y_{p,i} \in \{0,1\}, \quad 0 \le t_p,\ t_i,\ \theta_i^+, \theta_i^-, \quad \forall p \in P,\ i \in I \tag{2q}$$

In above formulation, objective (2a) minimizes the sum of runway occupying cost and the worst-case expected deviation cost. Constraint (2b) enforces that only one aircraft weight class is assigned to each position. Constraint (2c) ensures that all number of each aircraft weight class assigned in the sequence is equal to the number of flights which belong to the aircraft weight class. The separation time requirement between consecutive flights is satisfied by constraint (2d). For example, if aircraft category k is assigned to position p ($x_{p,k} = 1$) and followed by aircraft category l ($x_{p+1,l} = 1$), the safe separation distance between position p and $p + 1$ should not be less than $d_{k,l}$. The cost of runway occupying time is evaluated by constraint (2e) in economical way. The possible aircraft weight class assignments are given by the first-stage decision variable $x_{p,k}$, and the second-stage assignment decision is linked to the first-stage through constraint (2h). Constraint (2i) ensures each flight to be assigned exactly one position in the sequence in the second-stage decision. Constraint (2j) guarantees that each flight can not land before its earliest time τ_i. Separation requirements between consecutive flights are forced by constraint (2k). Constraints (2l) and (2m) determine exact landing time for flight i. Constraints (2n)-(2p) calculate the second-stage cost of deviation time for each flight i. The domains of all variables are ensured by (2f) and (2q) together.

B. APPROXIMATE WORST-CASE DISCRETE DISTRIBUTION

In ASRSP, the unspecific distribution $\mathbb{P}$ of aircraft arrival time is main challenge for solving this problem. Recently, Postek *et al.* [14] proposes a discrete distribution approximation, based on marginal distributions of random parameters for the two-stage ambiguous stochastic integer programs with mean-MAD information. Inspired by the research endeavor, we adopt the methodology used in [14] to this work.

1) TWO-STAGE AMBIGUOUS CONTINUOUS RECOURSE MODEL

First, we relax the integer decision variables in the second stage, the model of ASRSP translates into a two-stage ambiguous continuous recourse model. In accordance with [14], for the ambiguity set $\mathbb{F}$ in equation (1), the worst-case distribution $\mathbb{P}_{\bar{e}}$ turns out to be same for every first-stage decision so that the relaxed ambiguous recourse model in (2a) reduces to

$$\min \; g + \sup_{\mathbb{P}_{\bar{e}}\in\mathbb{F}} \mathbb{E}_{\mathbb{P}_{\bar{e}}}[Q(x,\bar{e})] \tag{3}$$

subject to constraint (2b)-(2f) and (2h)-(2p)

$$0 < y_{p,i} < 1, \quad 0 \le t_p,\ t_i,\ \theta_i^+,\ \theta_i^-, \quad \forall p \in P, \quad i \in I \tag{4}$$

where each component of $\bar{e}$ follows a known discrete distribution with almost three realization. Its proof combines the fact that second-stage value function is convex with results from Ben-Tal and Hochman [35], who provides closed-form expressions for the worst-case expectations maximizing and minimizing the expectations of convex and concave functions. This result is detailed below.

Proposition 2: According to Ben-Tal and Hochman [35], Given function $f(e)$ of one-dimensional random variable e is convex, and the ambiguity set $\mathbb{F}$ of the distribution $\mathbb{P}$ of e. There is

$$\sup_{\mathbb{P}\in\mathbb{F}} f(e) = p_1 f(a) + p_2 f(\mu) + p_3 f(b), \tag{5}$$

where

$$p_1 = \frac{d}{2(\mu - a)}, \quad p_2 = 1 - \frac{d}{2(\mu - a)} - \frac{d}{2(b-\mu)}, \quad p_3 = \frac{d}{2(b-\mu)} \tag{6}$$

The worst-case distribution $\mathbb{P}_{\bar{e}}$ is a three-point distribution on $\{a, \mu, b\}$ with probability values p_1, p_2, and p_3 respectively. Equation (5) considers only one sample random variable with dimension $n_e = 1$. For $n_e > 1$, covariance between different variables should be taken into consideration, as this proposition is extended multi-dimensions situation.

$$p_1^i = \frac{d_i}{2(\mu_i - a_i)}, \quad p_2^i = 1 - \frac{d_i}{2(\mu_i - a_i)} - \frac{d_i}{2(b_i - \mu_i)}, \quad p_3^i = \frac{d_i}{2(b_i-\mu_i)}, \quad i = 1, 2, \ldots, |I| \tag{7}$$

According to equation (7), ambiguity set $\mathbb{F}$ and independent assumption of flight arrival time, we can obtain the joint probability distribution of all aircraft arrival time e_i. Then the worst-case expectation of $f(e)$ equation (8) is calculated based on equation (6) by enumerating over all $3^{|I|}$ scenarios of outcomes a_i, μ_i, b_i of each elements e_i.

$$\sup_{\mathbb{P}\in\mathbb{F}} f(e) = \sum_{\alpha\in\{1,2,3\}^{|I|}} \prod_{i=1}^{|I|} p_{\alpha_i}^i f(\varepsilon_{\alpha_1}^1, \ldots, \varepsilon_{\alpha_{|I|}}^{|I|}), \tag{8}$$

where

$$\varepsilon_{\alpha_1}^i = a_i, \quad \varepsilon_{\alpha_2}^i = \mu_i, \ \varepsilon_{\alpha_3}^i = b_i, \ \forall i \in I \tag{9}$$

The worst-case distribution $\mathbb{P}_{\bar{e}}$ of random parameter $\bar{e}_i$ is a marginal distribution with $\mathbb{P}\{e_i = a_i\} = \frac{d_i}{2(\mu_i - a_i)}, \mathbb{P}\{e_i = \mu_i\} = 1 - \frac{d_i}{2(\mu_i-a_i)} - \frac{d_i}{2(b_i-\mu_i)}, \mathbb{P}\{e_i = b_i\} = \frac{d_i}{2(b_i-\mu_i)}, \quad i = 1, 2, \ldots, |I|$.

Example 3: Suppose there are two flights, $I = \{1, 2\}$. The partial probability distribution information of each flight are $a_1 = 5, b_1 = 10, \mu_1 = 8, d_1 = 2; a_2 = 6, b_2 = 12, \mu_2 = 9, d_2 = 1$. According to (7), the three-points marginal distribution of e_1 over a_1, μ_1 and b_1 with mass $2/6$, $1/6$, and $3/6$, respectively. e_2 follows a three-points marginal distribution over a_2, μ_2 and b_2 with mass $1/6$, $4/6$, and $1/6$, respectively.

2) TWO-STAGE AMBIGUOUS MIXED-INTEGER RECOURSE MODEL

The difficulty of having integer decision variables y in the second stage is that the second-stage value function $Q(x, e)$ is generally not convex so that the result of *Proposition* 2 cannot be applied. There are two approaches to deal with this difficulty for two-stage ambiguous mixed-integer recourse models in [14]. The first is using a value function $\hat{Q}(x, e)$ that is convex to approximate $Q(x, e)$. The other one is keeping the $Q(x, e)$ but treats that $\mathbb{P}_{\bar{e}}$ as an approximate worst-case probability distribution in the ASRSP. In this article, we take the latter approximation. Therefore, corresponding solutions are approximate worst-case solutions with the approximate three-point discrete distribution.

It can be observed that there are $3^{|I|}$ possible scenarios in the approximate worst-case probability distribution. Practically speaking, it is intractable to enumerate all the scenarios costly for capturing the uncertainty. Hence, in our work, we propose a hybrid sample average approximation (HSAA) algorithm, by capturing as more scenarios as possible, to obtain more approximate solutions.

IV. SOLUTION APPROACH

Here, this paper presents a nested GA to substitute off-the-shelf solvers in the Optimization Step of HASS algorithm. Specifically, a nested GA consisting of outer genetic algorithm 1 (GA1) and inner genetic algorithm 2 (GA2) is used to solve the sample average approximate problem with N scenarios. GA2 will be applied to determine the optimal second-stage scheduling decision designed to minimize total costs of deviation time in line with the first-stage decision. In other words, GA2 play as a role of solver for estimating the second-stage function value. The framework of the nested GA is presented in Figure 5 of Appendix A. The details of two genetic algorithms are displayed in Appendix A.

In this section, we sequentially introduce general SAA algorithm framework, sampling methods. Finally, we conclude this section with the framework of HSAA algorithm.

A. SAMPLE AVERAGE APPROXIMATION ALGORITHM FRAMEWORK

The basic idea of the SAA algorithm is to replace the original distribution of random parameter e with a sample of size N, that is, $e^s, s \in \Omega, \Omega = \{1, 2, \ldots, N\}$, where N is much smaller than the reality number of scenarios of e. Then the determinately equivalent SAA problem (SAAP) of original problem (2a)-(2q) can be written as follows:

$$\text{(SAAP)} v = \min \ g + \frac{1}{N}\sum_{s=1}^{N} Q(x, e^s)$$
$$\text{for each scenario } s \in \Omega$$
$$\text{subject to constraint (2b)-(2f) and (2h)-(2q)} \tag{10}$$

In implementation of the SAA algorithm, a lower bound and a set of upper bounds of the real objective value v are

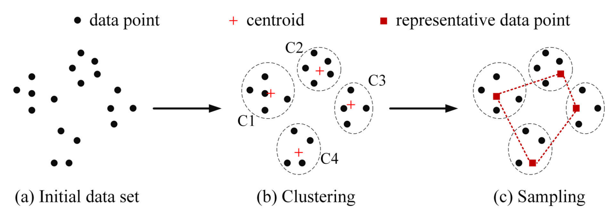


FIGURE 2. Illustration of clustering sampling procedure.

obtained. The general framework of SAA algorithm, which consists of five blocks, is detailed below.

1) (Sampling Step). Choose a sample size N and a number M denoting the number of replications. For $m = 1, 2, \ldots, M$ repeat sampling N scenarios from the population of approximate worst-case distribution.
2) (Optimization Step). For $m = 1, 2, \ldots, M$ repeatedly solve the SAAP by nested GA and store the optimal first-stage solution x_m and the corresponding objective value v_N^m. Compute statistical lower bound $\underline{v}$ and its variance $\sigma_{\underline{v}}^2$ by using $\underline{v} = \frac{1}{M}\sum_{m=1}^{M} v_N^m$ and $\sigma_{\underline{v}}^2 = \frac{1}{M(M-1)}\sum_{m=1}^{M}(v_N^m - \underline{v})^2$ respectively.
3) (Validation Step). For $m = 1, 2, \ldots, M$, generate a sample of size N' ($N' \gg N$). Estimate the objective value $v^s(x_m)$, $s = 1, 2, \ldots, N'$ for each individual in the sample with given first-stage solution x_m. Estimate the statistical upper bound $\overline{v}(x_m)$ and its variance $\sigma_{\overline{v}(x_m)}^2$ by using $\overline{v}(x_m) = \frac{1}{N'}\sum_{s=1}^{N'} v^s(x_m)$ and $\sigma_{\overline{v}(x_m)}^2 = \frac{1}{N'(N'-1)}\sum_{n=1}^{N'}\left(v^s(x_m) - \overline{v}(x_m)\right)^2$, respectively.
4) Calculate the confidence interval of lower bound and upper bound via $\underline{v} \pm t_{M-1,\alpha/2}\sigma_{\underline{v}}/\sqrt{M}$ and $\overline{v}(x_m) \pm t_{N'-1,\alpha/2}\sigma_{\overline{v}(x_m)}/\sqrt{N'}$ respectively, where the confidence level $\alpha = 0.05$ is usually adopted.
5) Select a desired upper bound from M number upper bound values according to some criteria. Compute the optimality gap by $gap = \frac{\text{UB}-\text{LB}}{\text{UB}} * 100\%$

B. K-MEANS AND K-MEANS++ CLUSTERING SAMPLING

In SAA algorithms, sampling is one very critical procedure in that the selected sample directly determines the objective value. To improve the accuracy of computational results with small or moderate sampling size, two clustering sampling methods applied in the HSAA algorithm to improve efficiency. Figure 2 shows a clustering sampling process.

The first clustering sampling method is based on K-means clustering technique, a set of scenarios are grouped into C minority separative clusters by K-means clustering method. For each cluster C_i, We choose a representative data point $s_i, i \in \{1, \ldots, N_k\}$, depending on the distance between data point x_j and centroid u_i. The nearest data point away from the centroid, marked by red solid circle, as shown in Figure 2-(b), is preferred as the representative scenario. Finally, we can obtain a set of typical scenarios Ω of size N_k that will used in the HSAA algorithm.

Instead of randomly selecting the k number centroids, in Kmeans++ clustering sampling, only one is randomly chosen from data set while the remaining $(k-1)$ cluster centers are selected as far from previous centroids as possible. Besides, the rest of the steps are similar to Kmeans case.

C. HYBRID SAMPLE AVERAGE APPROXIMATE (HSAA) ALGORITHM

In the HSAA algorithm, in order to improve the computing efficiency, we develop a nested GA, which algorithm framework is given in Figure 5 of the Appendix A, to substitute commercial solvers (e.g., Cplex) in Optimization Step. Once a feasible first-stage solution obtained, GA2 is used to estimated statistical upper bound in the Validation Step. The detail of the HSAA algorithm framework is presented in **Algorithm 1**. In **Algorithm 1**, the estimated computing time can be estimated as follows. Assuming that the running time of GA2 is 0.01s, for a problem with $popsize_1 = 50$, $maxgen_1 = 30$, $N = 30$, $|K| = 3$, the maximal computation time is about $50 \cdot 30 \cdot 30 \cdot 3 \cdot 0.01\text{s} = 1350\text{s}$. Repeated sampling 10 times, the total computation time for the Optimization Step in the HSAA algorithm is 13500s. In the Validation Step, if the sample size $N' = 1000$, the running time is $10 \cdot 1000 \cdot 3 \cdot 0.01\text{s} = 300\text{s}$.

Algorithm 1 Hybrid Sample Average Approximation Algorithm

Input: replication M, sample size N, big sample size N'; first-stage
Output: solution, lower bound and upper bounds;

```
for m = 1 : M do
    S_N ← generate a sample of size N by using sampling techniques;
    Solve the SAAP with sample S_N by nested GA;
    x_m ← a feasible first-stage solution;
end for
Estimated the statistical lower bound and associated confidence interval;
for m = 1 : M do
    S_N' ← generate a sample of size N' (N' ≫ N);
    for s ∈ S_N' do
        Solve the s-th second-stage problem Q(x_m, e^s) with given first-stage decision x_m by GA2;
    end for
    Estimate the statistical upper bound and associated the confidence interval;
end for
Select a upper bound with given criterion;
```

V. NUMERICAL EXPERIMENT

In this section, we conduct a series of numerical experiments to explain: (1) the efficiency and effectiveness of the proposed optimization model and solution algorithm. (2) how the duplication sampling number M and sample size N/N_k affect the performance of the HSAA algorithm; (3) the effect of different sampling methods on objectives values. We implement all the numerical experiments in MATLAB 2014b by using PC with Inter(R) Core(TM) i5-6500U CPU @3.2GHz and 4 GB RAM.

In Subsection V-A, we detail our experimental data and parameters pretest results. The improvement of runway scheduling comparing current practice and off-the-shelf solvers are reported in Subsection V-B. In Subsection V-C, the performance of the HSAA algorithm on parameters M and N/N_k are discussed. Finally, we analyze the impact of pending scenarios size N, used in clustering sampling methods, on computational results.

A. DATA DESCRIPTION AND PARAMETER PRETEST

The unit cost of runway utilization is taken $\lambda = 10$, the unit cost of delay and early is $\lambda_i^+ = 2{\cdot}k_i$ and $\lambda_i^- = k_i$, respectively, where $k_i = 1, 2, 3$, respectively represents weight class small, large and heavy. The distribution of the flight weight classes for all instances is uniform. The means μ_i, $i = 1, 2, ..., |I|$, of aircraft arrival time (time unit is seconds) are obtained according the arrival rate. Estimated arrival time e_i is contained in a $\mathbb{F}$ ambiguity set with $a_i = \mu_i - \mathrm{U}[300, 900]$, $b_i = \mu_i + \mathrm{U}[300, 900]$, $d_i = 300$, where U[300, 900] denotes the uniform distribution within interval [300, 900]. The aircraft's earliest landing time τ_i is no more than 30 minutes ahead of the estimated arrival time.

In the GA1 and GA2, selective pressure coefficient α, population size (*popsize*) and number of iterations (*maxgen*) are three key arguments, which have significant impact on convergence time and solution quality of the HSAA algorithm. According to parameter pretest results, ($\alpha_1 = 0.0001$, $\alpha_2 = 0.0001$) is chosen as the well-behaved selective pressure coefficients based on the trade-off between solution quality and computing time. A sound population size $popsize_2 = 50$ and iteration number $maxgen_2 \geq 30$ are preferred in the GA2. The solution quality of GA1 is guaranteed when $popsize_1 \geq 50$ and $maxgen_1 \geq 40$.

B. PERFORMANCE ANALYSIS OF THE NESTED GA AND TWO-STAGE OPTIMIZATION

With the aim of evaluating efficiency of the nested GA and effectiveness of the proposed two-stage optimization approach, Computational results by exact method (e.g. MIP with Cplex) and the current practice of First-Come-First-Serve (FCFS) (e.g., Balakrishan and Chandran [11] and Ng *et al.* [23]) way are taken as references for comparison of the landing cost and computation time. The description of test instances is shown in Table 2. Computational results of different problem size by the nested GA and the exact method using Cplex 12.8 from ILOG are presented in Table 3 and Table 4. After solving the instances through the proposed two-stage optimization, some far larger samples/scenarios are selected for evaluating the solution performance in more general situation. Based on these out-of-sample landing cost, on the other hand, we can get insights into how the two-stage optimization

TABLE 2. Instances used for model evacuation.

Instance	I1	I2	I3	I4	I5	I6
Flight size	15	15	30	30	60	60
Arrival rate (flights/hr)	30	60	30	60	30	60

TABLE 3. Comparision of nested GA and MIP with Cplex for the instances of small sample size.

N	Inst.	Nested GA			MIP with CPLEX			Gap
		Obj. value	Makespan	time (s)	Obj. value	Makespan	time (s)	
2	I1	29,336	15,690	59.4	26,708	13,970	837.6	0.09
	I2	25,913	13,680	40.5	24,883	12,670	677.6	0.04
3	I1	30,384	15,170	117.6	27,320	13,320	1,114.8	0.10
	I2	29,299	13,180	89.1	26,106	12,400	493.8	0.11
4	I1	29,173	15,600	181.3	28,476	13,350	1,385.4	0.02
	I2	25,928	12,960	157.1	24,459	12,960	1,481.3	0.06

can improve the runway scheduling efficiency compared with FCFS. Table 5 reports the out-of-sample results of deviation costs by the proposed modeling method and the landing cost by FCFS.

We first investigate the algorithm efficiency of HSAA, which uses the nested GA to substitute the exact method, with small sampling size (e.g., sampling size $N < 5$). These scenarios is obtained in random sampling way. As shown in Table 3, Cplex be able to obtain the landing cost for the instances "I1-I2" with 15 flights within 1,500 seconds. Column "Gap" records the relatively gap of objective value between nested GA and exact method, the maximum "Gap" value is 0.11 and its average is 0.07, indicating that the performance of nested GA is guaranteed. Particularly, the proposed nested GA is great better than the commercial solver in achieving better computational time at the cost of slight loss of solution quality.

Further, we test the solutions performance with large sampling size $N \in \{10, 20, 50\}$. The maximum computation time is 3,600 s. The time limit was chosen in accordance with the characteristic of the instances. These computational results are reported in Table 4. Symbol "-" means that there is no feasible solution before the computer is out of memory. As for the instances with 15 flights and sampling size $N = 10$, exact solver is not able to obtain a global optimal solution within one hour. On the contrary, the proposed nested GA yields more close-to-optimal solutions within 1,600 s comparing to the baseline solutions obtained by Cplex. In addition, the working computer is out of memory for the exact method since the number of flights exceeds 15 or sampling size $N > 10$. The computational results suggest that the exact method would be taken into account when the number of arriving flights and stochastic scenarios is small. Otherwise, the proposed nested GA is preferable for practical usage.

To demonstrate the effectiveness of the two-stage optimization model, we compare the first-stage solutions performance with current practice FCFS by using out-of-sample tests. Inherited from the first-stage solutions by solving the

TABLE 4. Comparision of nested GA and MIP with Cplex for the instances of large sample size.

N	Inst.	Nested GA Obj. value	Nested GA Makespan	Nested GA time (s)	MIP with CPLEX Obj. value	MIP with CPLEX Makespan	MIP with CPLEX time (s)
10	I1	27,242	15,080	244	27,393	13,580	>3,600
	I2	23,088	12,180	210	27,775	12,930	>3,600
	I3	49,171	31,140	1,570	158,953	29,020	>3,600
	I4	49,127	25,540	1,349	-	-	-
	I5	120,133	41,830	3,922	-	-	-
	I6	130,631	39,190	2,583	-	-	-
20	I1	28,946	15,080	527	-	-	-
	I2	22,806	12,730	406	-	-	-
	I3	51,286	30,910	2,876	-	-	-
	I4	50,281	25,600	2,214	-	-	-
	I5	116,267	41,470	7,734	-	-	-
	I6	110,806	39,810	5,637	-	-	-
50	I1	29,450	15,080	570	-	-	-
	I2	23,197	12,730	666	-	-	-
	I3	50,877	30,100	4,862	-	-	-
	I4	50,825	25,310	4,560	-	-	-
	I5	106,918	41,730	12,147	-	-	-
	I6	113,876	38,930	8,626	-	-	-

SAAP (10) with small sample size $N = 10$, we use $N' \in \{500, 1000, 10000\}$ scenarios from the approximate distribution to validate the deviation cost in the second stage. We directly evaluate the average landing cost by the same N' scenarios for current practice in last two columns of Table 5. It can be found that the deviation costs by exact method are larger than the nested GA. This observation is similar for the objective values in Table 4, which further demonstrates the conclusion that the nested GA outperforms Cplex with large sample sizes. Moreover, the average deviation costs by using FCFS are far grater than that by using the two-stage optimization regardless of which solution method is used. For example, the total deviation time costs of instance "I1" by using the proposed model are 12,665 and 13,058 respectively for meta-heuristic and exact solution method, which are much smaller than the FCFS of 12,486,596. The large differences between our proposed approach and current practice intuitively illustrate that the two-stage optimization can improve runway scheduling efficiency, especial in reducing deviation costs, compared with FCFS fashion. In additional, compare the column "Makespan" in Table 4 and Table 5, the average runway occupying cost of FCFS is around 1.5 times that of the two-stage optimization in instances "I5-I6" with a flight size of 60. This observation further indicates that the two-stage optimization is better than FCFS in terms of reducing both runway occupying time and total deviation time when the number of landing flight is large.

C. PERFORMANCE ANALYSIS OF THE HSAA ALGORITHM

To measure the performance of the HSAA algorithm, we firstly introduce two optimility gaps here based on statistical lower and upper bounds. The illustration of "optimality gaps" is presented in Figure 3. For the convenient, we use LB_E (UB_E) to denotes lower (upper) bound estimator,

TABLE 5. Comparision of the out-of-sample results.

N'	Inst.	Nested GA Deviation	CPLEX Deviation	FCFS Deviation	FCFS Makespan
500	I1	12,655	13,058	12,486,596	14,445
	I2	12,755	13,056	9,736,515	14,593
	I3	14,756	71,217	27,330,525	29,940
	I4	14,745	-	9,677,218	29,552
	I5	19,978	-	57,180,583	60,731
	I6	13,707	-	9,197,727	60,606
1000	I1	12,602	13,081	12,711,499	14,472
	I2	12,940	13,031	10,008,447	14,584
	I3	14,369	68,698	27,609,427	29,902
	I4	14,856	-	9,832,524	29,532
	I5	20,037	-	57,506,457	60,712
	I6	13,783	-	9,385,972	60,634
10000	I1	12,714	13,183	12,579,481	14,469
	I2	12,907	13,128	9,843,750	14,623
	I3	14,720	69,557	27,466,502	29,899
	I4	14,813	-	9,693,677	29,567
	I5	19,826	-	57,348,147	60,744
	I6	13,692	-	9,210,364	60,567

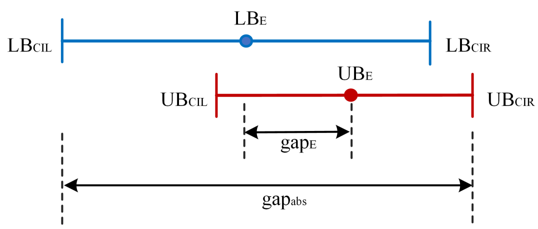


FIGURE 3. The graphic definition about two optimality gap values.

TABLE 6. Impact of duplication number *M* on HSAA by random sampling.

M	LB (95% conf. int.)	UB (95% conf. int.)	gap_E	gap_{abs}	time (s)
5	67,467±2,694	69,835±150	3.39%	7.45%	15,165
10	67,231±1,724	67,867±130	0.94%	3.66%	33,014
15	67,906±1,032	68,494±128	0.86%	2.55%	47,456
20	67,804±940	68,082±146	0.41%	2.00%	63,333
25	67,678±942	68,259±143	0.85%	2.44%	80,660
30	67,421±638	67,801±145	0.56%	1.71%	95,366

confidence interval left and right values are abbreviated as CIL and CIR, respectively. $gap_E = \frac{UB_E - LB_E}{UB_E} \cdot 100\%$ means the nominal optimality gap based on expectation values; $gap_{abs} = \frac{UB_{CIR} - LB_{CIL}}{UB_{CIR}} \cdot 100\%$ reports the absolute optimality gap value. Following experiments are based on instance "I3".

1) IMPACT OF DUPLICATION NUMBER *M* ON ALGORITHM CONVERGENCE

Set sample size $N = 30, N' = 10000$, we test duplication number $M \in \{5, 10, 15, 20, 25, 30\}$ with random sampling to get insights into how the duplication number M influences the HSAA algorithm. These results are presented in Table 6. In Table 6, column "LB" ("UB") record lower (upper) bound information at 95% confidence interval. Gap metrics are displayed in column 4-5 respectively. Finally, last column "time" records the computation time by seconds.

TABLE 7. Impact of sampling size N/N_k on HSAA algorithm with different sampling methods.

N/N_k	gap_E			gap_{abs}			time (s)		
	Random	Kmeans	Kmeans++	Random	Kmeans	Kmeans++	Random	Kmeans	Kmeans++
5	4.12%	1.76%	3.59%	7.24%	5.04%	6.43%	26,804	30,804	28,005
10	3.43%	2.48%	2.35%	5.70%	4.37%	4.41%	32,362	35,951	33,783
15	0.55%	2.01%	0.38%	2.66%	4.21%	2.05%	35,081	37,700	36,092
20	1.35%	2.74%	1.56%	2.89%	5.10%	3.38%	40,476	43,632	41,827
30	0.72%	0.13%	1.23%	2.89%	2.49%	3.16%	51,913	51,560	53,092
40	0.72%	0.58%	-0.29%	2.75%	2.60%	1.38%	62,294	59,264	58,920
50	0.41%	0.10%	1.38%	2.12%	1.80%	3.08%	68,389	63,548	67,175
60	-0.16%	0.63%	0.26%	1.27%	1.97%	2.07%	62,780	69,049	78,465
Average	**1.39%**	**1.31%**	**1.31%**	**3.44%**	**3.45%**	**3.25%**	**47,512**	**48,938**	**49,670**

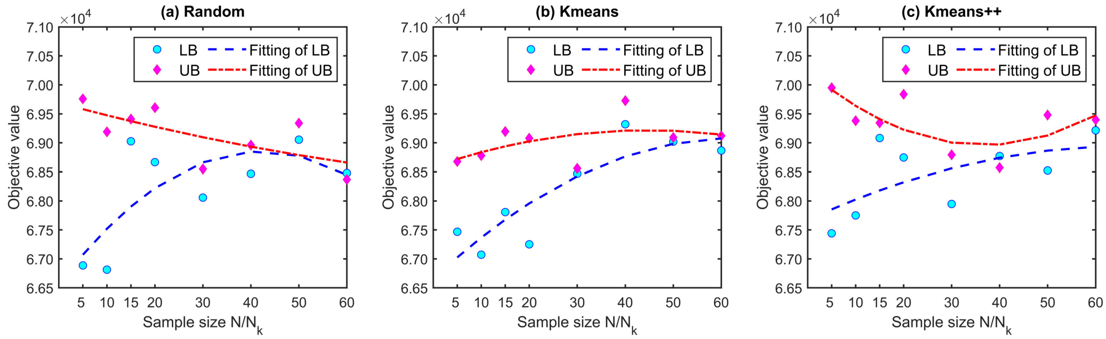


FIGURE 4. Curve fitting of LB and UB values on sample size with different sampling methods.

It can be observed from Table 6 that the length of the confidence interval of LB is decreasing as duplication number M increases. The value range of the lower bound reduces from [64773, 70161] to [66783, 68059] when M increases from 5 to 30. The shorter the confidence interval, the more accurate the lower bound. Moreover, the upper bound "UB" generally decreases when the duplication number M increases. Hence, as it can be seen from Table 6, the consideration of a larger M allows for generating a solution with a smaller optimality gap and a tighter lower bound confidence interval. However, more computing time is required to solve the SAA problem when the duplication number M increases. This observation indicates that a good compromise between the quality of solution and the computational effort should be made. Indeed, when $M = 15$, a near-optimal solution with $\text{gap}_E < 1\%$ and $\text{gap}_{abs} < 3\%$ is obtained within an acceptable computational time (less than 13 h). Though the quality of solutions considering more larger M are slightly better than the case with $M = 15$, the computational time needed is higher as well. In practice, it doesn't make sense to seek a slight increase in the accuracy of solution at expense of a large amount of computational time.

2) IMPACT OF SAMPLING SIZE N/N_k

Given M, how many scenarios are enough to obtain guaranteed solutions. To illustrate this influence by different sampling size and sampling methods, we analyze a set of sample sizes $N/N_k \in \{5, 10, 15, 20, 30, 40, 50, 60\}$, and randomly generate $N = 1000$ scenarios from the approximate distribution as pending sample for clustering sampling methods. According to the discussion in Subsection V-C.1, Take $M = 15$ based on the trade-off between solution quality and computing time. Experimental results are pictured in Figure 4 and presented in Table 7.

Figure 4 shows how the statistical lower and upper bounds change as sample size increases. All sampling cases converge toward an optimal solution as the number of sample size increases. Figure 4 clearly indicates that the gap_E becomes tight enough for further considerations at a sample size of 40 scenarios. The tighter gap mainly attribute to the lower bound increases shown in Figure 4.

It can be observed from Table 7 that as sample size used in HSAA algorithm increases, the values of gap_E by using the clustering sampling method in general are tighter than the random sampling way. When sampling size is of large (e.g., $N/N_k \geq 40$), we can observe that gap_E is converging to 0. The maximal gap_E among all sampling cases is smaller than 5%, and the average gap_E over different sampling approaches are of 1.39%, 1.31% and 1.31%, respectively. In addition, the observation of gap_{abs} is similar for gap_E, which decreases quickly and converges to 2% with sampling size increasing up to 60. The maximum and minimum gap_{abs} among all sampling methods are less than 8% and 2%, respectively. The gap_{abs} of the Random and K-means sampling cases is less than 3% since the sample size is greater than 30, and the gap_{abs} reduces to less than 4% in the K-means++ sampling

TABLE 8. Impact of different N on HSAA algorithm by using clustering sampling methods ($N_k = 20$).

N	gap_E		gap_{abs}		time (s)	
	Kmeans	Kmeans++	Kmeans	Kmeans++	Kmeans	Kmeans++
100	4.13%	2.99%	6.68%	5.51%	41,486	43,588
300	2.08%	1.70%	4.66%	4.15%	41,260	43,185
500	2.67%	1.21%	6.03%	3.60%	42,923	41,967
1000	2.74%	1.56%	5.10%	3.38%	44,632	41,827
1500	2.69%	0.86%	4.45%	3.30%	43,574	42,692
2000	1.31%	1.49%	3.81%	3.54%	41,695	41,943
Average	**2.60%**	**1.63%**	**5.12%**	**3.91%**	**42,595**	**42,534**

case when sample size $N_k \geq 15$. The average values of gap_{abs} are of 3.44%, 3.45% and 3.25%, respectively. The differences of computing times among different sampling methods used in the HSAA algorithm are nearly omitted. Hence, what can be concluded is that the K-means++ sampling technique outperforms the Random and K-means sampling methods in terms of the quality of solutions.

3) IMPACT OF N USED IN CLUSTERING SAMPLING METHOD ON COMPUTATIONAL RESULT

We discuss here that the impact of pending scenarios size N on solution quality by using clustering sampling in HSAA algorithm. What we care about are: (1) how many alternative scenarios is adequate to extract representative sample with size N_k used in the Optimization Step of the HSAA algorithm; (2) which clustering sampling method makes the HSAA algorithm perform better.

Table 8 reports the solution quality and computation times for each sample size N with $N_k = 20$. It can be intuitively observed that both gap_E and gap_{abs} of K-means++ sampling method are less than the values by using K-means sampling method with the exception of $N = 2000$. The average values of gap_E and gap_{abs} are of 2.60% and 5.12% by using the K-means clustering sampling, and are of 1.63% and 3.91% for the K-means++ sampling, respectively. These results suggests that K-means++ sampling behaves better than K-means sampling way in obtaining a tighter objective value with nearly same computational burden. Additionally, as shown in Table 8, the larger the number of alternative scenarios, the smaller the optimality gap. It is intuitive that the number of alternative scenarios size should be not less than 500 for a tight optimality gap.

VI. CONCLUSION

This work studies an ambiguous two-stage stochastic runway scheduling problem with partial distribution information available, in which the uncertainty mainly arises from the random aircraft arrival times. Contrast to the past researches, we relieve the assumption that the probability distribution of random parameters is exactly known, but consider the case that the distribution information of uncertain parameters is partial known. These information consist of means, support set and MAD. Based on the given information, a discrete probability distribution with three realizations per random parameter is approximately established. For solving the complicated problem, we develop a hybrid sample average approximate algorithm, into which a nested GA are embedded. In addition, the performance of three different sampling methods used to provided random scenarios set are analyzed. Numerical experiments are carried out to demonstrate the efficiency and validity of the proposed ambiguous two-stage optimization approach and solution algorithm. Computational results indicate that the runway scheduling efficiency and effectiveness by using the proposed model and algorithm are remarkable; on the other hand, clustering sampling methods, used in the HSAA algorithm, behave better than the random sampling way in improving approximate results. In general, this work further contributes to extending the TBFM, an auxiliary intelligence technique for air traffic management, to ambiguous stochastic environment in which only partial descriptive statistics information of uncertain elements are captured, and to proposing an efficient solution algorithm.

For the future research, it is preferable that the ambiguous stochastic runway scheduling will be extended to the all day time horizon, a rolling horizon scheduling plan with uncertain nature is necessary. To relieve the significant computational effort required, developing more efficient solution strategies for the Optimization Step, which is the computing bottleneck, is meaningful in the HSAA algorithm. Besides, it is quit considerable that extends the ambiguous two-stage runway scheduling problem to more general situations by taking into account the correlation of arrival time.

APPENDIX
THE DETAILS OF GA1 AND GA2
A. GENETIC ALGORITHM 1 (GA1)

The details of GA1 at the outer layer of the nested GA are presented below.

1) REPRESENTATION OF CHROMOSOME AND POPULATION INITIALISATION OF GA1

The appropriate encoding here is a 0-1 matric shown in Figure 6, the first to third rows respectively represent the assignment decisions of Small, Large and Heavy aircraft weight classes. Each column represents a gene unit. Initial population generate randomly with respecting constraints (2b) and (2c).

2) FITNESS EVALUATE AND SELECTION OPERATOR OF GA1

Linear transformation of fitness and tournament selection operator are adopted. Specifically, the estimation of select pressure difference is given in equation (11), where *popsize* denote population size, $objval_i$ and *bestobjval* are individual $i's$ objective value and the best value respectively. Fitness value of individual is evaluated by equation (12), in which **objval** is an objective value vector and f_i represents individual's fitness value. In equation (12) parameter $\alpha \in (0, 1)$ is used to adjust select pressure to ensure population

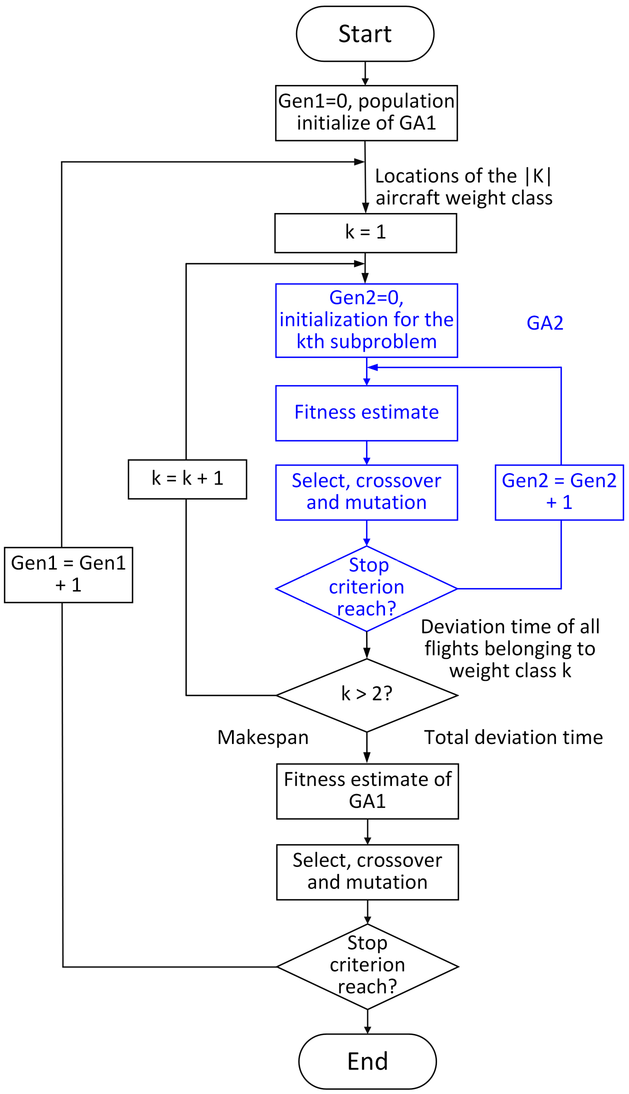


FIGURE 5. **An algorithm flowchart of the nested GA.**

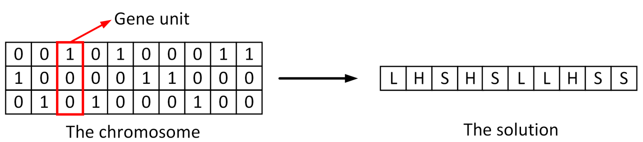


FIGURE 6. **Chromosome representation of GA1.**

diversity.

$$pressure = \frac{1}{popsize}\sum_{i=1}^{popsize} objval_i - bestobjval \quad (11)$$

$$f_i = -objval_i + max(objval) + \alpha \cdot pressure \quad (12)$$

3) ADAPTIVE PROBABILITIES OF CROSSOVER AND MUTATION

To avoid the premature local convergence arising from stationary crossover and mutation probabilities, adaptive probabilities of crossover and mutation approach proposed by [36] is applied here. The expression for p_c and p_m are Eqs. (13) and (14), respectively.

$$p_c = \begin{cases} k_1\dfrac{f_{max}-f'}{f_{max}-f_{avg}}, & f' \geq f_{avg} \\ k_3, & f' < f_{avg} \end{cases} \quad (13)$$

$$p_m = \begin{cases} k_2\dfrac{f_{max}-f}{f_{max}-f_{avg}}, & f \geq f_{avg} \\ k_4, & f < f_{avg} \end{cases} \quad (14)$$

where f_{max} and f_{avg} respectively denotes the maximum and the average population fitness value, f' refers to the larger fitness value between the paired parent chromosomes to be crossed, the individual fitness is represented by f, and weighting parameters $k_1, k_2, k_3, k_4 \leq 1$. In this work, we assign k_1 and k_3 a value of 1, and we use a value of 0.5 for k_2 and k_4.

4) CROSSOVER AND MUTATION OPERATION OF GA1

In GA1, two-points crossover operator is used to generate new offspring, the crossover procedure is detailed in Figure 7-(a). The repair operation of chromosome is necessary since the obtained proto-childs may be infeasible. For example, Proto-child1 is infeasible, because the constraint (2c) does not meets after crossover operation. In Proto-child1, the fixed number of aircraft belong to Small class should be 4 instead of 3. The proto-child1 and proto-child2 are revised by repairing the genes inherited from the parents. Figure 7-(b) illustrates the repair operation, randomly select a gene from a collection of genes that can be repaired and revise it so as to the feasibility constraints are met. Since we obtain the offspring, a mutation may happen when mutation probability p_m is satisfied. In GA1, two different genes are selected randomly, then the back gene unit is inserted into the rear position of the ahead one. This mutation manipulation can avoid to produce invalid mutation individuals.

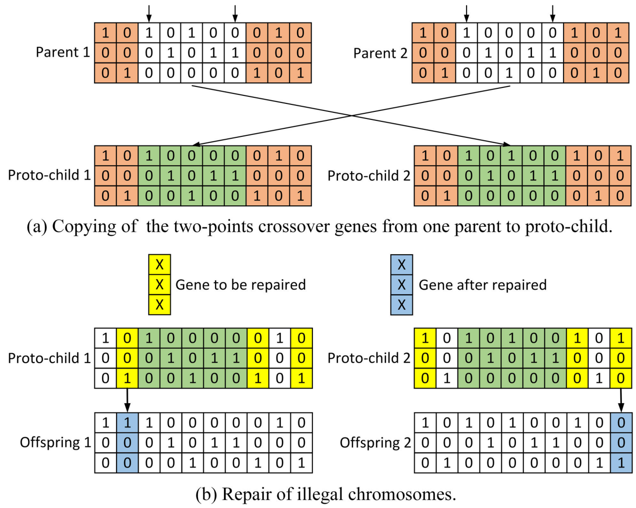
5) STOP CRITERIA

Iteration terminates when the maximum number of evolutionary generations (*maxgen*) reaches or the reduction of the best objective value is less than a threshold value (hereafter referred to as *thre*) within consecutive iterations (hereafter referred to as *iters*).

B. GENETIC ALGORITHM 2 (GA2)

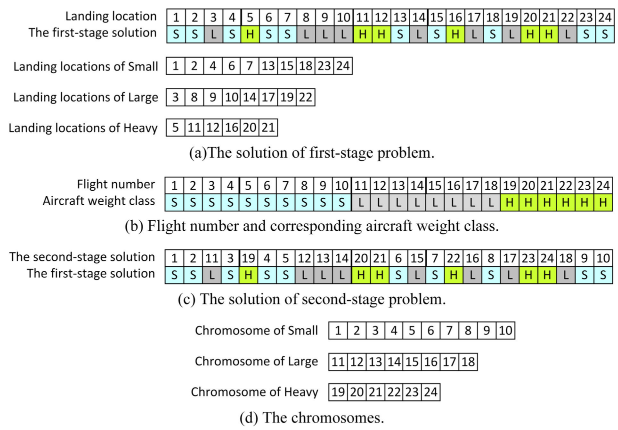
1) REPRESENTATION OF CHROMOSOME AND POPULATION INITIALISATION OF GA2

Figure 8-(a) shows the aircraft weight class assignment decision, the first row represents the landing locations (specific flight's landing positions), the aircraft weight classes (Small, Large or Heavy) on these locations are displayed in the second row. Landing locations of different aircraft weight classes are extracted from the first row and displayed in last three row. For example, the landing locations of small aircraft consist of positions 1, 2, 4, 6, 7,..., 23, 24. Figure 8-(b) details



(a) Copying of the two-points crossover genes from one parent to proto-child.
(b) Repair of illegal chromosomes.

FIGURE 7. Illustration of crossover and repair operations of GA1.



(a)The solution of first-stage problem.
(b) Flight number and corresponding aircraft weight class.
(c) The solution of second-stage problem.
(d) The chromosomes.

FIGURE 8. Illustration of chromosome representation of GA2.

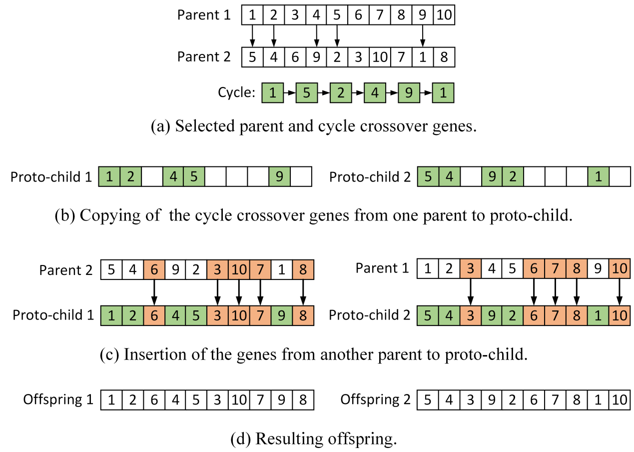
the flight number (a unique identification for each flight) of landing aircraft and corresponding aircraft weight class. According to the above scheduling characteristic, In GA2, a chain chromosome representation for flight scheduling over each aircraft weight class is employed. In this coding method, each chromosome is a sequence of flights to be landing. The detailed flight landing sequence of second-stage is shown in Figure 8-(c), the first row in Figure 8-(c) represents the flight number and the landing sequence. Figure 8-(d) illustrates the chromosome representation. In population initialization, individuals are randomly generated with the aim of covering the entire flights of particular aircraft weight category.

2) FITNESS EVALUATE AND SELECTION OPERATOR OF GA2

The fitness evaluate method and selection operator is same as the GA1.

3) CROSSOVER AND MUTATION OPERATION OF GA2

In GA2, Cycle Crossover (CX) is used. Figure 9 illustrates how to apply CX operator to generate new offspring. The detail implementations of CX comprises three steps. First,



(a) Selected parent and cycle crossover genes.
(b) Copying of the cycle crossover genes from one parent to proto-child.
(c) Insertion of the genes from another parent to proto-child.
(d) Resulting offspring.

FIGURE 9. Illustration of crossover operation of GA2.

randomly choose a gene (e.g., 1) from one parent (e.g., parent1) and extract an other gene (e.g., 5) located in the same position from another parent, then return to the parent1 and find out the corresponding position of 5, next select a gene (e.g., 2) located in position 5 from parent2. Repeat the previous work until a genes' loop is formed. As shown in Figure 9-(a), the selected genes' loop is $1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 9 \rightarrow 1$. Second, the proto-child1, as shows in Figure 9-(b), is generated by inserting the selected genes from parent1 to the corresponding positions. The proto-child2 is obtained in the same way. Third, as shown in Figure 9-(c), the remaining genes in parent2 and parent1 are inserted into proto-child1 and proto-child2 respectively. Finally, the resulting offspring are obtained. Double point mutation is a mutation operator in GA2, as mutation probability is met.

4) STOP CRITERIA

GA2 stops when the average fitness of population equals to the fitness of best individual.

REFERENCES

[1] VariFlight. *On-Time Rate Analysis: Global Airport Punctuality Rate Ranking*. Accessed: Oct. 3, 2018. [Online]. Available: https://data.variflight.com/

[2] A. Rodríguez-Díaz, B. Adenso-Díaz, and P. L. González-Torre, "Minimizing deviation from scheduled times in a single mixed-operation runway," *Comput. Oper. Res.*, vol. 78, pp. 193–202, Feb. 2017.

[3] C. Chen, D. Zhang, X. Ma, B. Guo, L. Wang, Y. Wang, and E. Sha, "Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1478–1496, Jun. 2017.

[4] S. Guo, C. Chen, J. Wang, Y. Liu, X. Ke, Z. Yu, D. Zhang, and D.-M. Chiu, "ROD-revenue: Seeking strategies analysis and revenue prediction in Ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, early access, Jun. 10, 2019, doi: 10.1109/TMC.2019.2921959.

[5] C. Chen, Y. Ding, Z. Wang, J. Zhao, B. Guo, and D. Zhang, "VTracer: When online vehicle trajectory compression meets mobile edge computing," *IEEE Syst. J.*, pp. 1–12, 2019.

[6] C. Chen, Y. Ding, X. Xie, S. Zhang, Z. Wang, and L. Feng, "TrajCompressor: An online Map-matching-based trajectory compression framework leveraging vehicle heading direction and change," *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 28, 2019, doi: 10.1109/JSYST.2019.2935458.

[7] F. A. Administration. *Next Generation Air Transportation System (NextGen)*. Accessed: Nov. 7, 2019. [Online]. Available: https://www.faa.gov/nextgen/

[8] *Eurocontrol. Single European Sky Atm Research (SESAR)*. Accessed: Nov. 7, 2019. [Online]. Available: https://www.sesarju.eu/

[9] N. A. R. Center. *Air Traffic Management (ATM) Technology Demonstration-1 (ATD-1): Terminal Sequencing and Spacing (TSS) and Flight Deck Interval Management (FIM)*. Accessed: Nov. 7, 2019. [Online]. Available: https://www.aviationsystemsdivision.arc.nasa.gov/research/tactical/atd1.shtml

[10] Y. Hong, B. Choi, and Y. Kim, "Two-stage stochastic programming based on particle swarm optimization for aircraft sequencing and scheduling," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1365–1377, Apr. 2019.

[11] H. Balakrishnan and B. G. Chandran, "Algorithms for scheduling runway operations under constrained position shifting," *Operations Res.*, vol. 58, no. 6, pp. 1650–1665, Dec. 2010.

[12] S. Solak, G. Solveling, J.-P.-B. Clarke, and E. L. Johnson, "Stochastic runway scheduling," *Transp. Sci.*, vol. 52, no. 4, pp. 917–940, Aug. 2018.

[13] J. C. Jones, D. J. Lovell, and M. O. Ball, "Stochastic optimization models for transferring delay along flight trajectories to reduce fuel usage," *Transp. Sci.*, vol. 52, no. 1, pp. 134–149, Jan. 2018.

[14] K. Postek, W. Romeijnders, D. den Hertog, and M. H. van der Vlerk, "An approximation framework for two-stage ambiguous stochastic integer programs under mean-MAD information," *Eur. J. Oper. Res.*, vol. 274, no. 2, pp. 432–444, Apr. 2019.

[15] P. Avella, M. Boccia, C. Mannino, and I. Vasilyev, "Time-indexed formulations for the runway scheduling problem," *Transp. Sci.*, vol. 51, no. 4, pp. 1196–1209, Nov. 2017.

[16] J. Abela, D. Abramson, M. Krishnamoorthy, A. De Silva, and G. Mills, "Computing optimal schedules for landing aircraft," in *Proc. 12th Nat. Conf. Austral. Soc. Oper. Res.*, Adelaide, Australia, Jul. 1993, pp. 71–90.

[17] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Scheduling aircraft Landings—The static case," *Transp. Sci.*, vol. 34, no. 2, pp. 180–197, May 2000.

[18] H. Pinol and J. E. Beasley, "Scatter search and bionomic algorithms for the aircraft landing problem," *Eur. J. Oper. Res.*, vol. 171, no. 2, pp. 439–462, Jun. 2006.

[19] D. Briskorn and R. Stolletz, "Aircraft landing problems with aircraft classes," *J. Scheduling*, vol. 17, no. 1, pp. 31–45, Feb. 2014.

[20] K. Artiouchine, P. Baptiste, and C. Dürr, "Runway sequencing with holding patterns," *Eur. J. Oper. Res.*, vol. 189, no. 3, pp. 1254–1266, Sep. 2008.

[21] G. Clare and A. G. Richards, "Optimization of taxiway routing and runway scheduling," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1000–1013, Dec. 2011.

[22] A. Ghoniem, H. D. Sherali, and H. Baik, "Enhanced models for a mixed arrival-departure aircraft sequencing problem," *Informs J. Comput.*, vol. 26, no. 3, pp. 514–530, Aug. 2014.

[23] K. K. H. Ng, C. K. M. Lee, F. T. S. Chan, and Y. Qin, "Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min-max regret approach," *Transp. Res. E, Logistics Transp. Rev.*, vol. 106, pp. 115–136, Oct. 2017.

[24] R. G. Dear and Y. S. Sherif, "An algorithm for computer assisted sequencing and scheduling of terminal area operations," *Transp. Res. A, Gen.*, vol. 25, nos. 2–3, pp. 129–139, Mar. 1991.

[25] A. Lieder, D. Briskorn, and R. Stolletz, "A dynamic programming approach for the aircraft landing problem with aircraft classes," *Eur. J. Oper. Res.*, vol. 243, no. 1, pp. 61–69, May 2015.

[26] A. Lieder and R. Stolletz, "Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways," *Transp. Res. E, Logistics Transp. Rev.*, vol. 88, pp. 167–188, Apr. 2016.

[27] J. Montoya, S. Rathinam, and Z. Wood, "Multiobjective departure runway scheduling using dynamic programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 399–413, Feb. 2014.

[28] J. A. Bennell, M. Mesgarpour, and C. N. Potts, "Dynamic scheduling of aircraft landings," *Eur. J. Oper. Res.*, vol. 258, no. 1, pp. 315–327, Apr. 2017.

[29] J. Beasley, J. Sonander, and P. Havelock, "Scheduling aircraft landings at london heathrow using a population heuristic," *J. Oper. Res. Soc.*, vol. 52, no. 5, pp. 483–493, 2001.

[30] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson, "Hybrid metaheuristics to aid runway scheduling at london heathrow airport," *Transp. Sci.*, vol. 41, no. 1, pp. 90–106, Feb. 2007.

[31] M. Samà, A. D'Ariano, F. Corman, and D. Pacciarelli, "Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas," *Transp. Res. C, Emerg. Technol.*, vol. 80, pp. 485–511, Jul. 2017.

[32] M. Niendorf, P. T. Kabamba, and A. R. Girard, "Stability analysis of runway schedules," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3380–3390, Dec. 2016.

[33] Y. Hong, B. Choi, K. Lee, and Y. Kim, "Dynamic robust sequencing and scheduling under uncertainty for the point merge system in terminal airspace," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2933–2943, Sep. 2018.

[34] L. Bianco, P. Dell'Olmo, and S. Giordani, "Scheduling models and algorithms for TMA traffic management," in *Proc. Modeling Simulation Air Traffic Manage.* Berlin, Germany: Springer, 1997, pp. 139–167.

[35] B.-T. A. and E. Hochman, "More bounds on the expectation of a convex function of a random variable," *J. Appl. Probab.*, vol. 9, no. 4, pp. 803–812, Dec. 1972.

[36] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 4, pp. 656–667, Apr. 1994.

MING LIU (Senior Member, IEEE) received the B.S. degree in management science and engineering and the Ph.D. degree in management science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2005 and 2010, respectively.

He is currently an Associate Professor with Tongji University, Shanghai. His research interests include logistics optimization and production scheduling.

BIAN LIANG received the B.S. degree in industry engineering from the Civil Aviation University of China, Tianjin, China, in 2016. He is currently pursuing the M.S. degree with the Economic and Management School, Tongji University. His research interest includes air operations management.

MAORAN ZHU received the B.S. degree in computer science and technology and the Ph.D. degree in management science and engineering from Tongji University, Shanghai, China, in 1995 and 2008, respectively.

He is currently an Associate Professor with Tongji University. His research interests include E-government information, cloud computing, and smart city.

CHENGBIN CHU received the B.S. degree in electrical engineering from the Hefei University of Technology, Hefei, China, in 1985, and the Ph.D. degree in computer science from the University of Metz, Metz, France, in 1990.

He was with the National Research Institute in Computer Science and Automation, Metz, from 1987 to 1996. He was a Professor with the University of Technology of Troyes, Troyes, France, from 1996 to 2008, where he was the Founding Director of the Industrial Systems Optimization Laboratory. He is currently a Professor with ESIEE Paris, Université Paris-Est, France. His research interests include operations research and modeling, analysis, and optimization of supply chain and production systems. He is currently an Associate Editor of the IEEE Transactions on Automation Science and Engineering and the IEEE Transactions on Industrial Informatics.