Uncertainty-Aware Flight Scheduling for Airport Throughput and Flight Delay Optimization

XIAODAO CHEN
China University of Geosciences, Wuhan, China

HAO YU
Shenzhen Airlines, Shenzhen, China

KUN CAO
East China Normal University, Shanghai, China

JUNLONG ZHOU, Member, IEEE
Nanjing University of Science and Technology, Nanjing, China

TONGQUAN WEI, Senior Member, IEEE
East China Normal University, Shanghai, China

SHIYAN HU, Senior Member, IEEE
University of Essex, Colchester U.K.

The continuous growth in the demand for air transportation exceeds the capacity of existing infrastructure, usually leading to unreliable flight schedules, i.e., long flight delays and uncertainties in arrival/departure and taxi times. We tackle the problem in this paper by designing an air traffic control algorithm, which can accommodate both airport throughput and flight quality of service in terms of flight delay on a given runway. The flight scheduling problem is formulated as an integer linear programming, and then converted to a multiobjective optimization problem which enables the computation of tradeoff between scheduling resolution and time complexity. Based on the multiobjective optimization, a heuristic algorithm considering uncertainties in flight arrival/departure time and taxi time is designed to achieve an improvement in airport throughput and a reduction in flight delay. Extensive simulations show that compared to benchmarking schemes, the proposed uncertainty-aware flight scheduling algorithm can improve the airport throughput and flight delay by up to 12.02% and 31.4%, respectively.

Manuscript received December 16, 2017; revised April 1, 2019; released for publication May 22, 2019.

DOI. No. 10.1109/TAES.2019.2921193

Refereeing of this contribution was handled by R. Sabatini.

This work was supported in part by the Shanghai Municipal Natural Science Foundation under Grant 16ZR1409000, in part by the National Natural Science Foundation of China under Grant 61802185, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180470, and in part by the Fundamental Research Funds for the Central Universities under Grant 30919011233.

Authors' addresses: X. Chen is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China, E-mail: (cxdao@yahoo.com); H. Yu is with the Boeing Fleet Group 4, Shenzhen Airlines, Shenzhen 518128, China, E-mail: (a09794@shenzhenair.com); K. Cao and T. Wei are with the Department of Computer Science and Technology, MOE Engineering Research Center for Software/Hardware Co-Design Technology and Application, East China Normal University, Shanghai 200241, China, E-mail: (52174506005@stu.ecnu.edu.cn; tqwei@cs.ecnu.edu.cn); J. Zhou is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, E-mail: (jlzhou@njust.edu.cn); S. Hu is with the Department of Electrical and Computer Engineering, University of Essex, Colchester CO4 3SQ, U.K., E-mail: (shiyan.hu@essex.ac.uk). *(Corresponding author: Tongquan Wei.)*

I. INTRODUCTION

The Federal Aviation Administration (FAA) forecasts a long term growth in the demand for air travel and delivery driven by world economy [1]. The number of general aviation hours flown, the revenue passenger miles, and the enplanements are projected to increase 3% a year through 2030, and there will be one billion passengers in U.S. in 2021. This increase in the demand for air transportation will result in numerous operation and maintenance issues including airport capacity overload, safety degradation, flight delays, aircraft fuel costs, and the degradation of passenger service quality [2]–[4]. Aviation authorities have been seeking methods that make a better use of existing infrastructures to resolve the situation while maintaining the required level of safety [5]. Similar to automated vehicle traffic control [6], [7], air traffic control (ATC) is a popular mechanism to prevent collisions. ATC organizes and expedites the flow of traffic such that it benefits both airports and airline companies. Nevertheless, ATC is still performed by human operators, which is error prone and cost inefficient [8]. In addition, it was reported that FAA has reached a decision to close 149 ATC towers in U.S. due to budget cuts [9]. As a result, it is imperative to design and deploy automated intelligent ATC systems in international and domestic airports.

Aircraft takeoff and landing are two key operations in an airport. In an airport equipped with intelligent ATC systems, aircraft scheduling is responsible for sequencing aircraft in takeoff and landing operations. A minimum separation distance is required between every pair of consecutive aircraft to avoid the interference from the wake-vortex of the leading aircraft. The separation distance is mandatory and specified by aviation authorities. It varies for different types of aircraft, which are generally classified by weight, i.e., heavy, medium, and light aircraft. The separation distance also can be modeled using a separation time interval. In this paper, it is assumed that in a busy airport resources are limited and there is only one runway for both landing and takeoff operation. Thus, the concerned aircraft scheduling problem deals with sequencing the aircraft on a runway to maximize airport throughput and minimize flight delays under the constraint of FAA-specified separation distance. Airport throughput is defined as the number of passengers delivered by aircrafts in an airport per unit time. Table I

TABLE I
Minimum Leading Time of Flights and Aircraft Capacity

Minimum leading time (seconds)				
		Trailing traffic		
		Boeing 707	Boeing 727	Boeing 747
Leading traffic	Boeing 707	70	100	72
	Boeing 727	70	80	72
	Boeing 747	181	200	96
Aircraft passenger capacity (# of passengers)				
	Boeing 707	Boeing 727	Boeing 747	
	219	189	605	

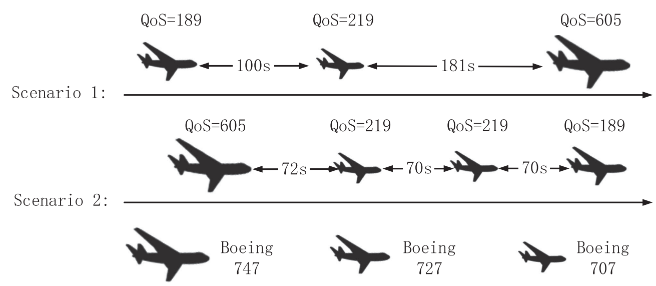


Fig. 1. Scheduling three types of aircraft in two scenarios produces different QoS.

gives an example of the minimum separation distances and capacity of different types of flights.

First-come first-served (FCFS) is the scheduling principle in many practical real-time scheduling algorithms that have been developed for runway operations. Extensive research effort has been devoted to the study of FCFS-based scheduling schemes. For example, D'Ariano *et al.* [10] and Samà *et al.* [11] studied the problem of sequencing aircraft takeoff and landing operations at congested airports. Several FCFS scheduling principle-based heuristics were designed and compared with branch-and-bound algorithm. In fact, the FCFS principle enforces a sequential order that can achieve scheduling efficiency, scheduling fairness, and controller preference [12]. However, the throughput of a runaway may be reduced due to relatively large spacing requirements of FCFS sequence of order [13]. As a result, the position shifting technique which resequences aircraft in contrast to FCFS discipline was proposed in [14] to compute optimal aircraft sequences with minimum delays.

Although position shifting techniques can effectively reduce schedule makespans and mean delay time, they fail to operate in real-time due to large computation overheads, hence they are not well suited for aircraft scheduling. To overcome drawbacks of position shifting methods, Malaek and Naderi described an efficient real-time algorithm in [13] for scheduling single and multiple runways. The presented algorithm is comparable to FCFS algorithms in terms of accommodating practical issues of real-time scheduling while enjoying optimality of minimizing makespans similar to that of position shifting methods. However, this combination of FCFS and position shifting does not take into account the uncertainty in aircraft departure and landing times.

In a real-world scenario, departure/landing and taxi times of aircraft often deviate from their predefined values. This uncertainty in departure/landing and taxi times results in inefficient airport usage and long aircraft delays [15]. Few research works have explored aircraft scheduling mechanisms considering uncertainties in landing/departure time and taxi time. Bosson and Sun [16] presented a comprehensive optimization model that minimizes the total time in the air and on the surface, and maximizes the punctuality performance of flights. They also tackled uncertainties in flight arrival/departure time by using a multistage stochastic programming technique. However, the sequence of flight takeoff/landing is not considered in this paper. Khanmohammadi *et al.* [17] presented a systematic approach to schedule landings of aircraft. The first step of the approach predicts the uncertainty in aircraft arrival time using an adaptive network based fuzzy inference system, and the second step of the approach prioritizes the arriving flights using a fuzzy decision making procedure. However, this approach does not consider the deviation of arrival times from predefined values. Moreover, the uncertainty in departure time and taxi time is not investigated.

We assume that each flight has a weight with respect to a time unit that indicates the quality of service (QoS) the flight can achieve during the interval. The QoS could be in the form of minimal flight delay or maximum passenger satisfactions. Fig. 1 illustrates two scenarios of flight scheduling, and gives the QoS obtained by different types of aircraft. These types of aircraft include Boeing 707 [18], Boeing 727 [19], and Boeing 747 [20]. The minimum time interval between any two types of aircrafts and passenger capacity of flights are given in Table I. For the sake of easy presentation, in this illustration example, we assume all passengers of a flight are satisfied with the flight and QoS of the flight is given by the number of passengers on board. Scenario 1 shows that 3 flights can be assigned to the runway while scenario 2 shows that 4 flights can be assigned to the same runway under the constraint of flight time interval. The throughput of the two scheduling solutions is 3.7 and 5.8 passengers per second, respectively, and the corresponding QoS of the scheduling solutions is 1031 and 1232, respectively. It is clear that scheduling solution 2 outperforms the solution 1 in terms of airport throughput and flight QoS.

In this paper, we concentrate on flight scheduling that jointly optimizes airport throughput and flight QoS, and propose an uncertainty-aware flight scheduling algorithm that sequences the arrival/departure order of flights on a given runway. The flight scheduling problem is first formulated as an integer linear program and then converted to a multiobjective optimization problem for achieving a trade-off between airport throughput and flight QoS. The major contributions are summarized as follows.

1) A flight traffic control algorithm for a given runway is proposed to accommodate both optimization objectives of airport throughput and flight QoS in terms of delay.
2) A stochastic flight scheduling mechanism is designed through considering uncertainties in flight arrival/departure time and taxi time. The proposed

TABLE II
Definitions of Main Notations Used in This Paper

Notation	Definition
Γ	A set of flights to be scheduled
T	Scheduling horizon
f_i	Flight i in the set Γ
A_i	Scheduling decision variable for flight i during the scheduling horizon T
β_i	Resultant throughput of scheduling flight i during the scheduling horizon T
τ	Scheduling time slot
α_i	Scheduling decision variable for flight i
α_i^τ	Scheduling decision variable for flight i in time slot τ
r_i	Release time for flight i when it becomes ready for landing/departing
d_i	Deadline for flight i when it must finish landing or departing process
l_i	Duration length of time for flight i when it employs the runway
w_i^τ	The QoS weight of flight f_i during the interval τ
$[r_i, d_i - l_i]$	The feasible interval in which a flight departure or landing is to be scheduled
θ_{ij}	The minimum interval between flight τ_i and τ_j to avoid interference from the wake-vortex of the leading aircraft
$f_1(\alpha_i)$	Throughput of flight i given in Eqn. (6)
$f_2(\alpha_i)$	QoS of flight i given in Eqn. (6)
$\lambda^i = (\lambda_1, \lambda_2)^T$	Tchebycheff weight vector of subproblem i
$r^* = (r_1^*, r_2^*)^T$	The reference point vector that gives the optimal solutions to two scalar objectives
Υ^i	Neighboring set of subproblem i containing Q closest weight vectors of vector λ^i
S	The non-dominated solution
$\varphi = \{\varphi_r, \varphi_t\}$	A variable modeling timing uncertainty in flight release and taxi time
φ_r	Adaptation variable for flight release time
φ_t	Adaptation variable for flight taxi time
σ	Deadline miss rate of scheduled flights
$R_{min,i}$	The earliest release time of flight f_i
$R_{max,i}$	The latest release time of flight f_i
$L_{min,i}$	The minimum taxi time of flight f_i
$L_{max,i}$	The maximum taxi time of flight f_i

scheme can handle uncertainties in air traffic and produce uncertainty-aware flight schedules.

3) Simulation experiments based on synthesized flight plan data have been used to verify the proposed scheme, which outperforms the benchmarking methods in airport throughput and flight delay by up to 12.02% and 31.4%, respectively.

The rest of this paper is organized as follows. Section II formulates the flight scheduling problem using integer linear program, Section III proposes the uncertainty-aware flight scheduling by accommodating uncertainties in arrival/departure time and taxi time. The proposed scheme is verified in Section IV and Section V concludes this paper.

II. INTEGER LINEAR PROGRAMMING (ILP)-BASED PROBLEM FORMULATION

The problem of scheduling landing or departing flights on a runway can be formulated as an ILP. The ILP aims to optimize the airport throughput, flight delays, and passenger satisfaction. Refer to Table II for notations used in this paper. Consider a set of flights Γ to be scheduled on a runway of an airport. The set Γ is assumed to contain N flights, i.e., $\Gamma = \{f_1, f_2, \ldots, f_N\}$, where f_i denote the ith flight in the set. The characteristic of flight f_i can be represented using a tuple $f_i = \{r_i, l_i, d_i\}$, where r_i denotes the release time of flight i when it becomes ready for landing/departing, d_i indicates deadline of flight i when it must finish landing or departing process, and l_i gives the taxi time, i.e., the duration length of time for flight i when it employs the runway.

Let T be the scheduling horizon that denotes the number of time units ahead of which the scheduling decision on flights is to be made. For flight $f_i \in \Gamma$, A_i is defined as a binary decision variable for the flight during the scheduling horizon T. A_i is set to 1 if flight f_i is scheduled in the horizon T and is reset to 0 otherwise. A parameter β_i is also introduced for flight f_i to indicate the resultant throughput of scheduling flight f_i in the horizon T. Based on the above description, the throughput of an airport having N flights to be scheduled is given by $\sum_{i=1}^{N} \beta_i \cdot A_i$, as shown in the first item of (1).

For each time unit $\tau \in \mathbf{T} \triangleq [1, 2, \ldots, T]$ in the horizon of scheduling, the variable α_i^τ denotes the scheduling decision for flight f_i during the interval τ. It is clear that α_i^τ is a binary scheduling decision variable. When flight f_i is scheduled to use the runway in the interval τ_i, α_i^τ is set to 1. Otherwise, it is set to 0.

Let w_i^τ be the QoS weight of flight f_i during the interval τ, and $[r_i, d_i - l_i]$ be the feasible interval in which a flight departure or landing is about to be scheduled. The QoS of flight f_i can be expressed as $\sum_{\tau=r_i}^{d_i-l_i} w_i^\tau \cdot \alpha_i^\tau$, as shown in the second item of (1). Note that the length of the feasible interval $[r_i, d_i - l_i]$ is in general greater than the time unit τ, and is typically set to some multiple of the τ.

The integer linear program for scheduling flights in the horizon T can then be formulated as follows:

$$\max \quad \sum_{i=1}^{N} \left(\beta_i \cdot A_i + \sum_{\tau=r_i}^{d_i-l_i} w_i^\tau \cdot \alpha_i^\tau \right) \tag{1}$$

$$\text{s.t.} \quad A_i \in \{0, 1\} \quad \forall f_i \in \Gamma \tag{2}$$

$$\sum_{i=1}^{N} \alpha_i^\tau = A_i, \forall \alpha_i^\tau \in \{0, 1\}, r_i \le \tau \le d_i - l_i \tag{3}$$

$$\alpha_i^\tau = 0 \quad \forall (\tau < r_i) \cup (d_i - l_i < \tau) \tag{4}$$

$$\alpha_i^\tau + \sum_{j \in \Gamma} \sum_{\tau'=\tau}^{\tau+l_i+\theta_{i,j}} \alpha_j^{\tau'} \le 1$$
$$\forall r_i \le \tau \le d_i - l_i \quad \forall f_i \in \Gamma, \forall \tau \in \mathbf{T}. \tag{5}$$

The objective function is given in (1), indicating a joint optimization for the airport throughput and flight QoS, the latter of which is in the form of flight delays or passenger satisfaction. For any flight $f_i \in \Gamma$, if it is scheduled in the horizon T [i.e., $A_i = 1$ as given in (2)], it must be scheduled at τ within the feasible scheduling interval $[r_i, d_i - l_i]$ (3). In other words, it cannot be scheduled outside the feasible scheduling interval (4). Let θ_{ij} denote the minimum interval between flight f_i and f_j to avoid interference from the wake-vortex of the leading aircraft. Only one flight can be scheduled in the duration of $(\tau + l_i + \theta_{i,j})$, where l_i is the length of time for which flight f_i uses the runway starting from τ (5). Equation (5) implicitly indicates that the maximum waiting time of flight f_i is $(d_i - l_i - r_i)$. In other words, flights with longer waiting time than $(d_i - l_i - r_i)$ are forced to stay in the waiting list, thus, will not be scheduled.

Note that the length of time unit τ determines the resolution of scheduling, which in turn has significant impact on scheduling accuracy and computational efficiency. A short duration of τ results in better scheduling solution since more time slots gives a finer resolution along the time horizon. In this case, the ILP problem size increases remarkably as a result of large amount of scheduling variables. Consequently, the computational complexity increases strikingly and it is infeasible for an ILP solver to derive an optimal schedule solution.

To handle the tradeoff between scheduling resolution and time complexity, a multiobjective evolutionary algorithm based on decomposition (MOEA/D) is developed to improve the computational efficiency without compromising scheduling accuracy, as discussed in Section III.

III. UNCERTAINTY-AWARE FLIGHT SCHEDULING ALGORITHM

The proposed algorithm consists of two parts. The first part is a deterministic flight scheduling algorithm and the second part is an uncertainty-aware flight scheduling algorithm. The deterministic flight scheduling algorithm is an MOEA/D-based approach. The uncertainty-aware flight scheduling algorithm is a stochastic programming based method which can efficiently handle the uncertainty of flight landing/departure and taxi time.

A. MOEA/D-Based Deterministic Flight Scheduling

MOEA/D is first proposed in [21] for the optimization of multiobjective problems. Unlike traditional multiobjective evolutionary algorithm in [22], the MOEA/D algorithm decomposes a multiobjective problem into a number of scalar subproblems, which are in turn optimized simultaneously. This technique has been widely used in the area of avionics [23].

The MOEA/D method is employed in this paper for the optimization of flight scheduling. The airport throughput given in the first term of (1) and flight QoS given in the second term of (1) are defined as two scalar optimization objectives. The two objectives denoted by $h_1(\beta_i)$ and $h_2(\alpha_i)$ are given as follows:

$$h_1(\beta_i) = \sum_{i=1}^{N} \beta_i \cdot A_i$$
$$h_2(\alpha_i) = \sum_{i=1}^{N} \sum_{\tau=r_i}^{d_i-l_i} w_i^\tau \cdot \alpha_i^\tau \tag{6}$$

where α_i is the scheduling decision variable for flight i, β_i is the resultant throughput of scheduling flight f_i in the horizon T, and w_i^τ is the QoS weight of flight f_i during the interval τ. Our goal is to optimize both the objectives given in (6).

Algorithm 1: MOEA/D-Based Flight Scheduling.

Input: Multiobjective problem, # of subproblems M, # of neighboring weight vectors Q, and uniformly distributed Tchebycheff vector λ for M subproblems

1: Set nondominated solution $S =$ NULL;
2: Set neighboring solution $\Upsilon =$ NULL;
3: For m subproblems, calculate Euclidean distance between any two vectors of a subproblem;
4: For each λ^i, pick its Q closest weight vectors to form its neighboring set Υ^i;
5: Generate an initial flight schedule using (1) with large τ and constant β and w;
6: Take schedule of (1) as the reference point;
7: For every subproblems, randomly select subproblems from Υ^i to generate a new solution sample;
8: Update $(r_1^*, r_2^*, \ldots, r_m^*)^T$ if new solution sample dominates;
9: Update neighboring solution Υ^i;
10: Update nondominated solutions to S;
11: Check if the convergence condition satisfied. If it is satisfied, S gives the solution; Otherwise, go to **step** 7;

The MOEA/D algorithm decomposes the flight scheduling problem into M subproblems by using the Tchebycheff approach [24]. The ith scalar optimization problem can be written in the form of

$$\min g(x|\lambda, r^*) = \max_{1\le i\le m} \{\lambda^i |h_i(x) - r_i^*|\} \tag{7}$$

where λ^i denotes the Tchebycheff weight vector of subproblem i. $r^* = (r_1^*, r_2^*, \ldots, r_m^*)^T$ is the reference point set where r_i^* in this set denote the optimal solution to $\max\{h_i(x)\}$. Thus, the reference point vector r^* essentially gives the optimal solutions to our two optimizing objectives.

Refer to Algorithm 1, which summarizes the MOEA/D-based flight scheduling algorithm. The algorithm proceeds iteratively. In each iteration, the MOEA/D method minimizes the scalar optimization problem given in (6), the optimal solution of which is a Pareto optimal solution of (1). After M weight vectors are obtained using Tchebycheff approach based decomposition, the MOEA/D calculates the Euclidean distance between any two weight vectors, and places the Q closest vectors of the vector λ^i in the set Υ^i. It is clear that Υ^i denotes the neighboring set of subproblem i containing Q closest weight vectors of vector λ^i.

The algorithm then runs the ILP-based approach given in (1) assuming a large time slot τ, and fixed β and ω. The airport throughput and flight QoS β and ω are deemed to be constant for a flight. Using a large time slot, the algorithm can compute an initial flight schedule fast. The initial flight schedule is in turn taken as the reference point of the MOEA/D algorithm in step 6.

Step 7 randomly select two subproblems from neighborhood Υ^i of subproblem i to generate a new solution.

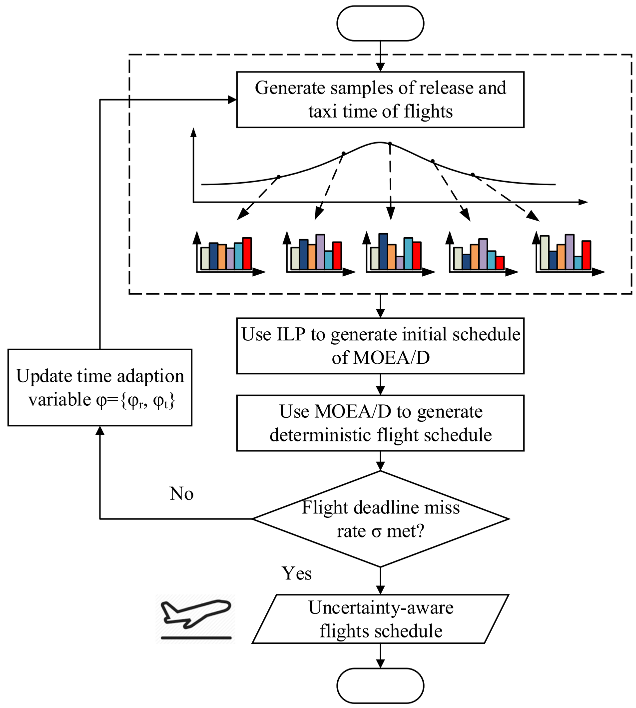


Fig. 2. Stochastic programming based flight scheduling approach.

This new solution is used to update the reference vector $r^* = (r_1^*, r_2^*, \ldots, r_m^*)^T$ if it dominates the existing solution. The neighboring set Υ^i of subproblem i is updated in step 9 accordingly. The nondominated solution is placed in S in step 10. This process repeats until the stop criterion is checked and satisfied in step 11. That is, process stops when the airport throughput and flight delay are satisfied.

B. Uncertainty-Aware Stochastic Flight Scheduling Algorithm

Due to undesirable weather conditions and other nondeterministic events, flights may experience delays for their departure/landing time or taxi time. This type of delays is often unpredictable, leading to timing violation of scheduled flights. In this case, a flight may not be able to land or takeoff before its scheduled deadline. As a result, the airport throughput is negatively impacted and flight QoS is degraded. To handle the timing uncertainty induced design issue, a stochastic programming based scheduling scheme is proposed in this section to generate a flight schedule that can adapt to timing uncertainties.

Fig. 2 illustrates the design flow of timing uncertainty-aware flight scheduling. This flow is motivated from our previous works [25], [26], while the works in [25] and [26] are for multiprocessor system scheduling which does not consider the unique constraints in flight scheduling problem. A timing adaptation variable $\varphi = \{\varphi_r, \varphi_t\}$ is defined to model the uncertain property of flight release and taxi time, and σ is utilized to denote the deadline miss rate of scheduled flights. As shown in Fig. 2, the timing adaptation variable $\varphi = \{\varphi_r, \varphi_t\}$ is iteratively calculated to obtain a flight schedule, such that the deadline miss rate of the schedule meets design requirements. In each iteration, ILP-based flight scheduling algorithm generates an initial flight schedule for the MOEA/D-based algorithm, which in turn produces a deterministic schedule for a given value of timing adaption variable $\varphi = \{\varphi_r, \varphi_t\}$. After this step, the Monte Carlo simulation is utilized to evaluate the timing adaption variable φ with respect to flight deadline miss rate. If the miss rate reaches a predefined threshold value, the procedure stops and a uncertainty-aware flight schedule is generated. Otherwise, the process moves to the next iteration with an updated value of $\varphi = \{\varphi_r, \varphi_t\}$. The following sections describe details of the proposed uncertainty-aware flight scheduling algorithm.

1) *Timing Adaptation Enabled Parallel Flight Scheduling:* The timing adaption variable φ plays a critical role in the stochastic programming. It is a tuple that can be written as $\varphi = \{\varphi_r, \varphi_t\}$, where φ_r and φ_t indicate the adaptation variable for flight release time and taxi time, respectively. The variables φ_r and φ_t take values in the range of [0, 1]. Let $R_{\min,i}$ and $R_{\max,i}$ denote the earliest and latest release time of flight f_i, respectively, and $L_{\min,i}$ and $L_{\max,i}$ be the minimum and maximum taxi time of the flight, respectively. Considering the adaptation variable φ_r and φ_t for flight f_i, the actual release and taxi time of the flight is then given as follows:

$$r_i = \varphi_r \cdot R_{\min,i} + (1 - \varphi_r) \cdot R_{\max,i} \tag{8}$$

$$l_i = \varphi_t \cdot L_{\min,i} + (1 - \varphi_t) \cdot L_{\max,i}. \tag{9}$$

The boundary values of release and taxi time of a flight indicate the corner case of the flight, the scheduling based on which will generate a conservative solution. Hence, in this paper we propose an adaptive algorithm that iteratively tunes the variable φ_r and φ_t to match the uncertainties in flight release and taxi time. In each iteration, the algorithm solves the optimization problem, and checks if the current φ_r and φ_t approximate the uncertainty in release and taxi time of the flight. Note we use Monte Carlo simulation technique for this check, which is detailed in Section III-B2.

Due to the stochastic property of φ_r and φ_t, the values of the two variables can be derived by using a step search method combined with Monte Carlo simulation technique. φ_r and φ_t are initialized to 0, and the step size is set to a value between 0 and 1. We use φ_r to illustrate the derivation of values of the two random variables. Let the step size of φ_r be 0.1. Then, the variable can take the values of $0, 0.1, 0.2, \ldots, 1.0$. For each of these values, a Monte Carlo simulation is conducted, the first value at which the flight schedule meets design requirements is deemed to be the value that represents the stochastic property of flight release time. This value can be used to generate approximate release time of flights according to (8), and the resultant flight schedule is supposed to meet design requirements. The value of φ_t can be derived in the same way. The step search procedure is essentially parallel, that is, we can search the values of the random variables in the interval of [0, 0.5] and [0.5, 1] simultaneously or even recursively. Thus, the key role of the step search method is to speed up the search process by using parallel platforms such as multicores or multiprocessors. The Monte Carlo simulation method used to derive timing adaptation variables are detailed in the next section.

2) *Monte Carlo Simulation Based Timing Adaption Variable Evaluation:* One of the goals of the proposed uncertainty-aware flight scheduling algorithm is to derive the timing adaptation variables such that the deadline miss rate σ of the generated flight schedule is satisfied. To this end, a Monte Carlo simulation is performed to iteratively evaluate the concerned timing adaptation variables φ_r and φ_t.

As illustrated in Fig. 2, the proposed stochastic programming-based flight scheduling approach derives values of timing adaptation variables φ_r and φ_t using Monte Carlo simulation. It first generates 10 000 samples of release time and taxi time of flights based on their distribution of probability. Gaussian distribution of probability is used to model the uncertainty in release and taxi time. Note that the proposed approach is not restricted to the Gaussian distribution. Once samples of flight release and taxi time are generated, an initial flight schedule is generated for each sample by using the ILP-based technique given in Section II, which is in turn fed to the MOED/D-based algorithm for further refinement. The deadline miss rate σ of a flight schedule is calculated as the ratio of the number of samples where flight deadlines are satisfied to the total number of samples. If the current deadline miss rate meets the design requirement, the resultant flight schedule and corresponding timing adaptation variables are the desired ones and the algorithm exits. Otherwise, the proposed algorithm updates the timing adaptation variables (φ_r and φ_t) for the next iteration. It has been shown that a simulation of 10 000 samples is quite sufficient to get stable results [25]. The obtained timing adaptation variables essentially represents the stochastic property of flight release and taxi time, thus, can be used to estimate the actual release and taxi time according to (8) and (9).

IV. NUMERICAL EVALUATION

We have conducted extensive simulation experiments to validate the proposed scheme in terms of improvements in airport throughput and flight QoS. In this section, we first give simulation settings for validation, then describe performance metrics for evaluation, and finally present and analyze the results.

A. Simulation Settings

We collected real plan data of arrival/departure flights of Eastern China Airline. These data are from PVG and SHA of Shanghai, two of the largest airports in China. Since the two airports are also used by major domestic and international airlines and we have no access to flight plan data of these airlines, we synthesized the flight plan data of the two airports in different scheduling horizon based on real flight plan data of Eastern China Airline. Because arrival data and departure data are equivalent in terms of validating the proposed scheme, we only use departure data of flights in the simulation experiments for the sake of simplicity. The release time of a flight is extracted from the synthesized flight plan data. The taxi time of the flight is assumed to follow a normal distribution of the probability with the mean of 10 min [27], [28]. Since a flight that departures within 15 min of the scheduled time is deemed to be punctual [29], [30], we define the deadline of the flight as its release time plus its taxi time and 15 min offset. The proposed uncertainty-aware parallel flight sequencing algorithm is implemented in C#. The simulation was performed on a machine with Intel Core i7-4720HQ 2.6 GHz CPU and 8GB memory.

B. Performance Metrics

In this section, we introduce two metrics to evaluate the proposed flight sequencing algorithm. These two metrics are airport throughput under given punctuality rate and QoS of a flight, as described below.

1) *Airport Throughput:* The throughput of an airport indicates the number of passengers delivered by aircraft in unit time (one day). In this paper, we use throughput under given punctuality rate to evaluate the proposed scheme. The punctuality rate is defined as the ratio of the number of flights arriving/departing within 15 min of the scheduled arrival/departure time to the total number of scheduled flights. It has been shown that the punctuality rate of the most punctual airlines in the world is about 90%. For example, based on percentage of punctual flights, the Hawaiian Airlines was ranked first with the punctuality rate of 93% in North America airlines in 2015 and BMI regional was the most punctual airline in the U.K. in 2013, with the punctuality rate of 92% [29], [30]. However, the punctuality rate of airlines in Asia is about 10% lower than that of airlines in North America and Europe. In particular, the punctuality rate of three largest airports in China, PVG/SHA in Shanghai and PEK in Beijing, is below 40% in the first half of 2013 according to flight data of FlightStats [30]. A wide range of punctuality rates is used in this experiment to investigate the impact of various flight sequencing schemes on airport throughput and flight QoS.

2) *Flight QoS:* The QoS of a flight mainly depends upon passenger satisfaction, which has many contributing factors such as cost and fees, in-flight services, boarding/deplaning/baggage, flight crew, and punctuality of the flight. From the viewpoint of an airport, the punctuality of a flight is the key factor that determines the efficiency of the airport operation. Hence, we define the flight QoS as a normalized function of the period of delay. When a flight does not have any delay, its QoS is deemed to be 1. Otherwise, the QoS of the flight is a value in the range between 0 and 1 and degrades with increase in delay. In other words, the QoS of a flight is inversely proportional to the delay beyond the scheduled arrival/departure deadline of the flight.

C. Experimental Results and Analysis

We compared the proposed uncertainty-aware flight sequencing scheme with four benchmarking approaches,

TABLE III
Comparison Between the Proposed Algorithm and Benchmarking Methods in Terms of the Average Throughput (# of Passengers Delivered in Unit Time) for Target Punctuality Rate of 0.4 (the Lowest Rate in the World)

Flight set	$\varphi_r = 0$ $\varphi_t = 0$	$\varphi_r = 0$ $\varphi_t = 1$	$\varphi_r = 1$ $\varphi_t = 0$	$\varphi_r = 1$ $\varphi_t = 1$	$0 < \varphi_r < 1$ $0 < \varphi_t < 1$
size	Θ_{00}	Θ_{01}	Θ_{10}	Θ_{11}	$\Theta_{Proposed}$
100-150	37231	39282	40444	42239	41706
150-200	49049	51719	53224	55556	54864
200-250	65577	69128	71137	74237	73319
250-300	80937	85357	87855	91719	90574

TABLE IV
Comparison Between the Proposed Algorithm and Benchmarking Methods in Terms of the Average Throughput (# of Passengers Delivered in Unit Time) for Target Punctuality Rate of 0.7 (the Average Rate in the World)

Flight set	$\varphi_r = 0$ $\varphi_t = 0$	$\varphi_r = 0$ $\varphi_t = 1$	$\varphi_r = 1$ $\varphi_t = 0$	$\varphi_r = 1$ $\varphi_t = 1$	$0 < \varphi_r < 1$ $0 < \varphi_t < 1$
size	Θ_{00}	Θ_{01}	Θ_{10}	Θ_{11}	$\Theta_{Proposed}$
100-150	39836	40886	41465	42385	42113
150-200	52441	53791	54568	55747	55394
200-250	70199	71971	72949	74497	74041
250-300	86603	88828	90091	92035	91459

TABLE V
Compare the Proposed Algorithm and Benchmarking Methods in Terms of the Average Throughput (# of Passengers Delivered in Unit Time) for Target Punctuality Rate of 0.9 (the Highest Rate in the World)

Flight set	$\varphi_r = 0$ $\varphi_t = 0$	$\varphi_r = 0$ $\varphi_t = 1$	$\varphi_r = 1$ $\varphi_t = 0$	$\varphi_r = 1$ $\varphi_t = 1$	$0 < \varphi_r < 1$ $0 < \varphi_t < 1$
size	Θ_{00}	Θ_{01}	Θ_{10}	Θ_{11}	$\Theta_{Proposed}$
100-150	41659	41999	42187	42486	42397
150-200	54740	55211	55463	55876	55754
200-250	73220	73816	74146	74667	74513
250-300	90432	91174	91600	92249	92056

TABLE VI
Compare the Proposed Algorithm and Benchmarking Methods in Terms of the Average Flight QoS for Target Punctuality Rate of 0.4 (the Lowest Rate in the World)

Flight set	$\varphi_r = 0$ $\varphi_t = 0$	$\varphi_r = 0$ $\varphi_t = 1$	$\varphi_r = 1$ $\varphi_t = 0$	$\varphi_r = 1$ $\varphi_t = 1$	$0 < \varphi_r < 1$ $0 < \varphi_t < 1$
size	Φ_{00}	Φ_{01}	Φ_{10}	Φ_{11}	$\Phi_{Proposed}$
100-150	63.06%	77.50%	85.56%	100.00%	94.46%
150-200	63.17%	77.53%	85.64%	100.00%	94.48%
200-250	63.23%	77.58%	85.66%	100.00%	94.48%
250-300	63.11%	77.45%	85.66%	100.00%	94.47%

TABLE VII
Compare the Proposed Algorithm and Benchmarking Methods in Terms of the Average Flight QoS for Target Punctuality Rate of 0.7 (the Average Rate in the World)

Flight set	$\varphi_r = 0$ $\varphi_t = 0$	$\varphi_r = 0$ $\varphi_t = 1$	$\varphi_r = 1$ $\varphi_t = 0$	$\varphi_r = 1$ $\varphi_t = 1$	$0 < \varphi_r < 1$ $0 < \varphi_t < 1$
size	Φ_{00}	Φ_{01}	Φ_{10}	Φ_{11}	$\Phi_{Proposed}$
100-150	81.50%	88.73%	92.77%	100.00%	97.23%
150-200	81.52%	88.76%	92.76%	100.00%	97.23%
200-250	81.63%	88.78%	92.85%	100.00%	97.24%
250-300	81.54%	88.71%	92.84%	100.00%	97.23%

TABLE VIII
Compare the Proposed Algorithm and Benchmarking Methods in Terms of the Average Flight QoS for Target Punctuality Rate of 0.9 (the Highest Rate in the World)

Flight set	$\varphi_r = 0$ $\varphi_t = 0$	$\varphi_r = 0$ $\varphi_t = 1$	$\varphi_r = 1$ $\varphi_t = 0$	$\varphi_r = 1$ $\varphi_t = 1$	$0 < \varphi_r < 1$ $0 < \varphi_t < 1$
size	Φ_{00}	Φ_{01}	Φ_{10}	Φ_{11}	$\Phi_{Proposed}$
100-150	93.83%	96.23%	97.60%	100.00%	99.07%
150-200	93.88%	96.24%	97.64%	100.00%	99.08%
200-250	93.86%	96.25%	97.61%	100.00%	99.08%
250-300	93.82%	96.20%	97.62%	100.00%	99.07%

which are constructed by using uncertainty characteristics of both flight release time and taxi time. To be specific, the four benchmarking methods indicate the scenario where adaptation variables for flight release time and taxi time are ($\varphi_r = 0, \varphi_t = 0$), ($\varphi_r = 0, \varphi_t = 1$), ($\varphi_r = 1, \varphi_t = 0$), and ($\varphi_r = 1, \varphi_t = 1$), respectively. The airport throughput of the four benchmarking methods are hence denoted by Θ_{00}, Θ_{01}, Θ_{10}, and Θ_{11}, respectively, and the flight QoS of the four methods are denoted by Φ_{00}, Φ_{01}, Φ_{10}, and Φ_{11}, respectively. The case where $0 < \varphi_r < 1$ and $0 < \varphi_t < 1$ indicates the stochastic scenario our proposed method is supposed to deal with. We denote by Θ_{Proposed} and Φ_{Proposed} the airport throughput and flight QoS of the proposed algorithm, respectively.

It can be derived from historical flight data in [29] and [30] that the highest, average, and lowest punctuality rate of airports throughout the world is about 90%, 70%, and 40%, respectively. We hence conduct simulation experiments for the three cases in terms of airport throughput and flight QoS.

Table III compares the average throughput of the proposed algorithm with that of four benchmarking schemes for flight sets with varying sizes under the given target punctuality rate of 40%. Since the punctuality of an airport is the percentage of flights arriving/departing within 15 min of scheduled time of arrival/departure over a period of 30 days, the throughput given in Table III is also averaged over 30 days assuming different number of flights in an individual day. As given in the table, the average throughput achieved by the proposed algorithm ($0 \leq \varphi_r \leq 1, 0 \leq \varphi_t \leq 1$) is close to that of the approach designing for the best case ($\varphi_r = 1, \varphi_t = 1$), and can be up to 12.02% higher than that of the approach designing for the worst case ($\varphi_r = 0, \varphi_t = 0$), 6.17% higher than that of the approach designing for the corner case ($\varphi_r = 0, \varphi_t = 1$), and 3.12% higher than that of the approach designing for the corner case ($\varphi_r = 1, \varphi_t = 0$).

In addition, Tables IV and V compare the average throughput of the proposed algorithm with that of four benchmarking schemes for flight sets with varying sizes under the given target punctuality rate of 70% and 90%, respectively. From the results given in the table, we can draw the same conclusion that the proposed algorithm ($0 \leq \varphi_r \leq 1, 0 \leq \varphi_t \leq 1$) achieves similar throughput when compared to the approach designing for the best case ($\varphi_r = 1, \varphi_t = 1$), and higher throughput when compared to the approaches designing for the worst case ($\varphi_r = 0, \varphi_t = 0$) and the corner cases ($\varphi_r = 0, \varphi_t = 1$) and ($\varphi_r = 1, \varphi_t = 0$).

Table VI gives the QoS of flights achieved by the proposed algorithm and benchmarking methods averaged over a period of 30 days under the target punctuality rate of 40%. The results listed in the table indicate that the QoS of flights achieved by the proposed algorithm is high and much better than that of benchmarking methods. Specifically, the proposed algorithm ($0 \leq \varphi_r \leq 1, 0 \leq \varphi_t \leq 1$) achieves a high QoS (above 94.4%), which is close to that of the approach designing for the best case ($\varphi_r = 1, \varphi_t = 1$). When compared to the approach designing for the worst case ($\varphi_r = 0, \varphi_t = 0$), the QoS achieved by the proposed

TABLE IX
Simulation Result Comparison With the Actual Flight Data

Basic Information					Actual Flight Data				Simulation Result			
Aircraft ID number	Call sign	Type of Flights	Sched. Dept. Order	Exp. Dept. Time	Dept. Order	Dept. Time	Thrpt.	QoS	Dept. Order	Dept. Time	Thrpt.	QoS
001	AAR8738	Large	1	09:25	2	09:35	415	90.00%	2	09:35	415	90.00%
002	HSF1097	Large	2	09:30	1	09:30	384	100.00%	1	09:30	384	100.00%
003	KAL1138	Large	3	09:35	3	09:45	351	90.00%	3	09:45	351	90.00%
004	AAR8708	Large	4	09:45	5	10:08	373	77.00%	5	10:02	402	83.00%
005	KAL1260	Large	5	09:45	4	10:06	364	79.00%	6	10:05	368	80.00%
006	JJA124	Large	6	09:50	6	10:18	290	72.00%	8	10:15	302	75.00%
007	HAN232	Large	7	09:55	7	10:25	396	70.00%	11	10:25	396	70.00%
008	JJA126	Large	8	10:00	8	10:28	282	72.00%	4	10:00	392	100.00%
009	B5976	Small	9	10:10	10	10:30	143	80.00%	7	10:12	175	98.00%
010	B6716	Small	10	10:10	9	10:29	154	81.00%	10	10:16	179	94.00%
011	B6831	Small	11	10:15	12	10:38	152	77.00%	9	10:15	198	100.00%
012	B8571	Small	12	10:25	13	10:45	192	80.00%	13	10:30	228	95.00%
013	B8226	Small	13	10:25	11	10:35	181	90.00%	12	10:25	202	100.00%
014	B6330	Small	14	10:35	14	10:54	168	81.00%	15	10:50	176	85.00%
015	AAR8610	Large	15	10:40	16	11:10	397	70.00%	14	10:45	539	95.00%
016	KAL1140	Large	16	10:50	15	11:00	414	90.00%	16	10:50	460	100.00%
Tot./Avg.							4656	81.19%			5167	90.94%

algorithm can be up to 31.4% higher. When compared to the approach designed for the corner cases ($\varphi_r = 0, \varphi_t = 1$) and ($\varphi_r = 1, \varphi_t = 0$), the QoS achieved by the proposed algorithm can be up to 17.02% and 8.9% higher, respectively. The flight QoS of the proposed algorithm and benchmarking methods under the target punctuality rate of 70% and 90% are presented in Tables VII and VIII, respectively. The results in the table also demonstrate the effectiveness of the proposed algorithm in terms of improving the flight QoS.

To further validate the proposed uncertainty-aware flight sequencing scheme, we compared simulation results of the proposed approach with actual flight data. As given in Table IX, both large aircraft 4 and 5 are scheduled to takeoff at 09:45 following the large aircraft 3 at 9:35. However, their actual takeoff time is 10:06 and 10:08, respectively, which has a large variation of almost 20 min due to uncertainty in flight release time. Taking uncertainty of flight release time into account, the proposed approach predicts that the takeoff time of aircraft 4 and 5 is 10:02 and 10:05, respectively. It can also be seen from the table that results of the proposed sequencing scheme approximate to actual flight data with respect to airport throughput and flight QoS in the presence of uncertainty. For example, the total throughput is 5167 in the simulation and 4656 in real flight data, and the average QoS is 90.94% in the simulation and 81.19% in real flight data. The discrepancy between actual and simulation data is within 10%.

Overall, the proposed algorithm can achieve a higher throughput of the airport and a better QoS of the flight as compared to benchmarking methods, both of which have been clearly demonstrated in the Tables III–IX. The higher throughput and the better QoS of the proposed algorithm benefit from the consideration of uncertainties in flight arrival/departure time and taxi time, which are handled by a stochastic programming based scheduling scheme developed in the proposed algorithm.

V. CONCLUSION

We tackle the flight scheduling problem of sequencing the arrival/departure order of flights on a given runway under the uncertainty of timing uncertainty in flight departure/landing or taxi time. The goal of this work is to design algorithms that optimize both airport throughput and flight QoS. We formulate the flight scheduling problem as an integer linear program; and we transform the integer linear program into a multiobjective optimization problem to achieve a tradeoff between airport throughput and flight QoS. We also design a stochastic flight scheduling algorithm that considers uncertainties in flight arrival/departure time and taxi time, thus the proposed algorithm can be adapted to uncertainties in air traffic and produce more resilient flight schedules. Our proposed stochastic flight scheduling algorithm is shown to improve the airport throughput and flight QoS by up to 12.02% and 31.4% as compared to benchmarking schemes, respectively.

REFERENCES

[1] Federal Aviation Administration FAA aerospace forecast fiscal years 2010–2030
Washington, DC, USA, 2010. [Online]. Available: http://www.faa.gov/data_research/aviation/aerospace_forecasts/2010-2030/media/2010%20Forecast%20Doc.pdf

[2] K. Cai, J. Zhang, M. Xiao, K. Tang, and W. Du
Simultaneous optimization of airspace congestion and flight delay in air traffic network flow management
IEEE Trans. Intell. Transp. Syst., vol. 18, no. 11, pp. 3072–3082, Nov. 2017.

[3] Y. Zhang, R. Su, Q. Li, C. G. Cassandras, and L. Xie
Distributed flight routing and scheduling for air traffic flow management
IEEE Trans. Intell. Transp. Syst., vol. 18, no. 10, pp. 2681–2692, Oct. 2017.

[4] A. Bicchi and L. Pallottino
On optimal cooperative conflict resolution for air traffic management systems
IEEE Trans. Intell. Transp. Syst., vol. 1, no. 4, pp. 221–231, Dec. 2000.

[5] J. Villarroel and L. Rodrigues
An optimal control framework for the climb and descent economy modes of flight management systems
IEEE Trans. Aerosp. Electron. Syst., vol. 52, no. 3, pp. 1227–1240, Jun. 2016.

[6] E. Meissner, T. Chantem, and K. Heaslip
Optimizing departures of automated vehicles from highways while maintaining mainline capacity
IEEE Trans. Intell. Transp. Syst., vol. 17, no. 12, pp. 3498–3511, Dec. 2016.

[7] D. Desiraju, T. Chantem, and K. Heaslip
Minimizing the disruption of traffic flow of automated vehicles during lane changes
IEEE Trans. Intell. Transp. Syst., vol. 16, no. 3, pp. 1249–1258, Jun. 2015.

[8] H. Erzberger and R. A. Paielli
Concept for next generation air traffic control system
Air Traffic Control Quart., vol. 10, no. 4, pp. 355–378, 2002.

[9] Federal Aviation Administration Press release—FAA makes tower closing decision
2013. [Online]. Available: http://www.faa.gov/news/press_releases/news_story.cfm?newsId=14414

[10] A. D'Ariano, P. D'Urgolo, D. Pacciarelli, and M. Pranzo
Optimal sequencing of aircrafts take-off and landing at a busy airport
In *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Funchal, Portugal, Sep. 19–22, 2010, pp. 1569–1574.

[11] M. Samà, A. D'Ariano, and D. Pacciarelli
Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports
Procedia-Social Behav. Sci., vol. 80, pp. 531–552, 2013.

[12] H. Erzberger
Design principles and algorithms for automated air traffic management
Knowl.-Based Funct. Aerosp. Syst., vol. 7, no. 2, pp. 1–31, 1995.

[13] S. Malaek and E. Naderi
A new scheduling strategy for aircraft landings under dynamic position shifting
In *Proc. IEEE Aerosp. Conf.*, 2008, pp. 1–8.

[14] H. Balakrishnan and B. Chandran
Scheduling aircraft landings under constrained position shifting
In *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, 2006, Paper 6320.

[15] I. Rusnak
Optimal guidance laws with uncertain time-of-flight
IEEE Trans. Aerosp. Electron. Syst., vol. 36, no. 2, pp. 721–725, Apr. 2000.

[16] C. S. Bosson and D. Sun
Optimization of airport surface operations under uncertainty
J. Air Transp., vol. 24, no. 3, pp. 84–92, 2016.

[17] S. Khanmohammadi, C. Chou, H. W. L. III, and D. Elias
A systems approach for scheduling aircraft landings in JFK airport
In *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2014, pp. 1578–1585.

[18] Wikipedia, "Boeing 707.
2019. [Online]. Available: https://en.wikipedia.org/wiki/Boeing_707

[19] Wikipedia Boeing 727.
2019. [Online]. Available: https://en.wikipedia.org/wiki/Boeing_727

[20] Wikipedia Boeing 747.
2019. [Online]. Available: https://en.wikipedia.org/wiki/Boeing_747

[21] Q. Zhang and H. Li
MOEA/D: A multiobjective evolutionary algorithm based on decomposition
IEEE Trans. Evol. Comput., vol. 11, no. 6, pp. 712–731, Dec. 2007.

[22] J. Zhou *et al.*
Resource management for improving soft-error and lifetime reliability of real-time MPSoCs
IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., p. 1, 2018.

[23] A. Gardi, R. Sabatini, and T. Kistan
Multiobjective 4D trajectory optimization for integrated avionics and air traffic management systems
IEEE Trans. Aerosp. Electron. Syst., vol. 55, no. 1, pp. 170–181, Feb. 2019.

[24] K. Miettinen
Nonlinear Multiobjective Optimization, vol. 12. Berlin, Germany: Springer, 2012.

[25] T. Wei, X. Chen, and S. Hu
Reliability-driven energy-efficient task scheduling for multiprocessor real-time systems
IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 30, no. 10, pp. 1569–1573, Oct. 2011.

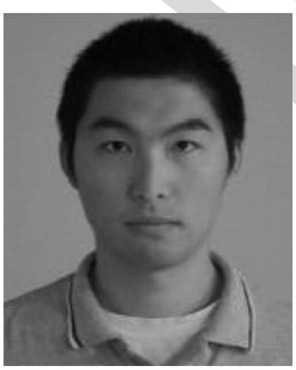
[26] K. Cao *et al.*
Affinity-driven modeling and scheduling for makespan optimization in heterogeneous multiprocessor systems
IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., p. 1, 2018.

[27] H. Idris, J.-P. Clarke, R. Bhuva, and L. Kang
Queuing model for taxi-out time estimation
Air Traffic Control Quart., vol. 10, no. 1, pp. 1–22, 2002.

[28] N. Shen, H. Idris, and V. Orlando
Estimation of departure metering benefits at major airports using queuing analysis
In *Proc. IEEE/AIAA 31st Digit. Avionics Syst. Conf.*, 2012, pp. 4E5–1–4E5–14.

[29] Statista The portal for statistics.
2019. [Online]. Available: https://www.statista.com/

[30] Flightstats Flight data services and applications.
2019. [Online]. Available: http://www.flightstats.com/go/Home/home.do



Xiaodao Chen received the B.Eng. degree in telecommunication from Wuhan University of Technology, Wuhan, China, in 2006, the M.Sc. degree in electrical engineering and Ph.D. degree in computer engineering from Michigan Technological University, Houghton, USA, in 2009 and 2012, respectively.

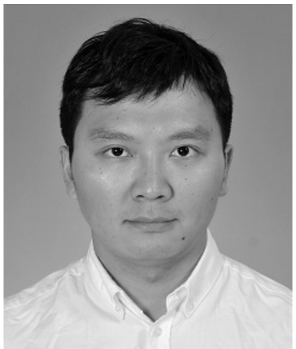
He is currently an Associate Professor with the School of Computer Science, China University of Geosciences, Wuhan, China.

Hao Yu received the master degree in safety engineering from China University of Geosciences, Wuhan, China. He is currently a Boeing 737 Captain with Shenzhen Airlines. He has more than 4600 h commercial flight experience and his research interest is in air traffic management.

Kun Cao is currently working toward the Ph.D. degree in computer science with the Department of Computer Science and Technology, East China Normal University, Shanghai, China.

His current research interests are in the areas of high performance computing, multiprocessor systems-on-chip, and cyber physical systems.

Mr. Cao was the recipient of the Reviewer Award from Journal of Circuits, Systems, and Computers in 2016.



Junlong Zhou (S'15–M'17) received the Ph.D. degree in computer science from East China Normal University, Shanghai, China, in 2017.

He was a Visiting Scholar with the University of Notre Dame, Notre Dame, IN, USA, during 2014–2015. He is currently an Assistant Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include real-time embedded systems, cloud computing and IoT, and cyber physical systems, where he has authored or coauthored more than 40 refereed papers.

Tongquan Wei (M'11–SM'19) received the Ph.D. degree in electrical engineering from Michigan Technological University, Houghton, MI, USA, in 2009. He is currently an Associate Professor with the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests are in the areas of Internet of Things, edge computing, cloud computing, and design automation of intelligent systems and cyber physical systems. He has authored or coauthored numerous papers in these areas, most of which are published in premium conferences and journals.

Dr. Wei serves as a Regional Editor for the *Journal of Circuits, Systems, and Computers* since 2012. He also served as the Guest Editor of the IEEE TII SS on Building Automation, Smart Homes, and Communities, the ACM TESC SS on Embedded Systems for Energy-Efficient, Reliable, and Secure Smart Homes, and the ACM TCPS SS on Human-Interaction-Aware Data Analytics for Cyber-Physical Systems.

Shiyan Hu (SM'10) received the Ph.D. degree in computer engineering from Texas A&M University, Uvalde, TX, USA, in 2008.

He is currently the Chair and a Professor of Cyber-Physical Systems (CPS) with the University of Essex, Colchester, U.K. He was an Associate Professor and the Director of Center for Cyber-Physical Systems, Michigan Tech., Houghton, MI, USA. He was also a Visiting Professor with IBM Research (Austin) in 2010, and a Visiting Associate Professor with Stanford University from 2015 to 2016. His research interests include CPS, CPS security, smart energy CPS, data analytics, and computer-aided design of VLSI circuits, where he has authored or coauthored more than 100 refereed papers.

Dr. Hu is an ACM Distinguished Speaker, an IEEE Systems Council Distinguished Lecturer, an IEEE Computer Society Distinguished Visitor, a recipient of the 2017 IEEE Computer Society TCSC Middle Career Researcher Award, the 2014 National Science Foundation (NSF) CAREER Award, and the 2009 ACM SIGDA Richard Newton DAC Scholarship. He is a Fellow of IET.