



# Solving slot allocation problem with multiple ATFM measures by using enhanced meta-heuristic algorithm

Jing Tian<sup>a</sup>, Xinchang Hao<sup>b,\*</sup>, Jibo Huang<sup>a</sup>, Jinglei Huang<sup>a</sup>, Mitsuo Gen<sup>c</sup>

<sup>a</sup> State Key Laboratory of Air Traffic Management System and Technology, Nanjing, China

<sup>b</sup> Changzhou Institute of Technology, Changzhou, China

<sup>c</sup> Fuzzy Logic Systems Institute and Tokyo University of Science, Tokyo, Japan

## ARTICLE INFO

### Keywords:

Air traffic flow management  
Slot allocation  
Priority-based rule  
Meta-heuristic optimization

## ABSTRACT

In Air Traffic Flow Management (ATFM), one key issue is to determine the Calculated Take Off Time (CTOT) for each flight. Due to the limited slot time resources, one single flight's slot decision could be influenced by plenty of other flights.

Few existing approaches in the previous literatures consider the interaction among various flights, so that the possible solution achieved will perform low stability and performance, especially under the condition of multiple measures. Aiming to solve the resource and flights coupling problems, in this paper, we present one heuristic method based on multi-dimension flight priority, with novel proposed methods of slot division and flight ranking. Meanwhile, in order to improve the optimality and flexibility of solutions, Estimation of Distribution Algorithm (EDA) is adopted. As one *meta*-heuristic algorithm, EDA is used to optimize the assignment sequence based on probability model, by which to provide an enhanced approach to solve slot allocation optimization problems. The numerical experiments have been conducted, the comparison results show that our approach has high calculation performance compared with conventional *meta*-heuristic algorithm-based approach, and better stability and applicability for re-allocation under the dynamic situation.

## 1. Introduction

In recent decades, civil aviation in China has developed a lot and the aviation efficiency and safety have raised more concern than ever. Air Traffic Flow Management (ATFM) was proposed to increase the operational performance of Air Traffic Management (ATM) and make the traffic control more effective. Main target of ATFM is to realize demand and capacity balancing, by using ATFM measures of increasing capacity and controlling the traffic volume. In sum, it is to control the aircraft flow under the capacity of airspace and controllers, so as to guarantee the safety of air traffic operations.

In China, till now, the primary ATFM measure is Miles in Trail (MIT), and in European and USA, many other ATFM measures are utilized, such as Ground Delay Program (GDP), Airspace Flow Management (AFP), Collaborative Trajectory Options Program (CTOP) and so on. For all the measures mentioned above, the core issue is called ground delay or ground delay policy, that means postponing some flights later than their scheduled departure time, to realize balancing between the traffic flow and capacity of airspace. Although CTOP hybrids ground delay and

rerouting, the basis is still ground delay.

The essential of ground delay is to transfer from air holding to ground delay, because ground delay is much cheaper and safer than air holding. If all the delay flights could be controlled before departure, benefits of both economic and social will be very high. As a result, the critical problem of ground delay is to determine which flights have to wait and how long they have to. In order to solve this problem, two main aspects should be well considered: parameters setting for ATFM measure, and slot allocation algorithm (Liu, Li, Yin, Zhu, & Han, 2017). In the domain of ATFM measure parameters setting, it contains, such as starting time and end time, operational capacity, exempted flights, etc. In the domain of slot allocation algorithm, it is to decide the CTOT for each flight (Ribeiro, Jacquillat, Antunes, Odoni, & Pita, 2018). It contains slot division, flight ranking, constraint checking on capacity and separation. Usually, the parameters are initially decided by ATFM supervisor, and then revised based on the feedback of solution made by slot assignment iteratively. Although partial parameters could be advised by some methods or systems, for solving the problem of ground delay, the core issue is how to achieve solutions with slot assignment

\* Corresponding author at: North Campus Wushan Road, Changzhou City, Jiangsu Province, China.

E-mail address: [haoxc@outlook.com](mailto:haoxc@outlook.com) (X. Hao).

<https://doi.org/10.1016/j.cie.2021.107602>

algorithm.

During the course of managing flight traffic, there are several challenges, which are highlighted as follows:

- (a) It is difficult to divide planning horizon into time slot with considering both efficacy with smaller delay and stability on flight CTOTs.
- (b) It is difficult to decide flight ranking, especially for multiple ATFM measures occurred simultaneously. The situation of multiple ATFM measures means that, when we try to calculate CTOTs for flights, there are more than one ATFM measures in operation, and these measures may influence the same flight. As a result, when we try to assign time slot for one flight, all related measures influenced should be considered. In other words, the relations among all ATFM measures have to be well modelled, which is more complicated and practical situation than conventional ones.
- (c) It becomes low calculation performance for constraint checking under multiple ATFM measures. Meanwhile, it becomes instability when a flight having CTOT is effected by a new ATFM measures, especially with too many measures appearing continuously, which is a practical situation of aviation operation in China.

Therefore, the issue of slot allocation problems are divided into three main aspects: we provide one novel mechanism to divide the planning horizon into time slots; we design one multi-dimension flight priority framework containing ATFM levels and flight own information and state; we propose one priority-based slot allocation algorithm with constraint checking.

The paper is structured as follows. In Section 2, we give a literature review of related topics, especially focus on research of ground delay policy and slot allocation algorithm. Section 3 describes slot allocation algorithm, including mechanism of slot division, multi-dimension flight priority and priority-based slot allocation heuristic method. Then one enhanced *meta*-heuristic algorithm is embedded into the framework we proposed to improve the searching ability. In Section 4, we provide some sensitivity analysis with different parameters of proposed model and case study. Finally, Section 5 summarizes the work and presents the future research direction.

## 2. Literature review

### 2.1. Ground delay policy

In 1960s, National Airspace System (NAS) could not satisfy the rapid increase of transportation, air holding was proposed to mitigate the arrival delay problems. However, too much air holding would decrease the safety level and increase the pressure of controllers. In order to solve this problem, the ATFM policy that is tailored towards that postponing some flights later than their scheduled departure time, to realize balancing the traffic flow and capacity of airport and airspace. The basic idea of ground delay policy is to convert air holding with high cost into ground delay with lower cost.

Thereafter, in order to improve the operational performance, Federal Aviation Administration (FAA) firstly developed one novel ATFM measures called GDP (Mukherjee, Hansen, & Grabbe, 2009), by giving each affected flight one CTOT. Next, in 2000s, some mechanisms of Collaborative Decision Making (CDM) were introduced, airports and airlines were integrated into GDP policy, which is named Ground Delay Program Enhancement (GDP-E). From 1980s, a great many of GDP researches have already been conducted by researchers and institutes, and several approaches and papers have been published (Fei, 2010). Andreatta and Romanin-Jacur (1987) firstly tried to build mathematic model to solve slot assignment problem in GDP, in which dynamic programming was used to assign departure time slots for flights, minimizing the total delay costs. Later, the multiple airports ground delay problem was firstly

modelled and solved by Vranas, Bertsimas, and Odoni (1994), with the use of several integer programming models to achieve different objectives. Collaborative Decision Making (CDM) was developed to enhance the GDP by Milner (1995), in which conducted the research and developed a dynamic model for slot allocation with airline engaged. Mukherjee, Hansen, and Grabbe (2012) proposed a dynamic stochastic programming model for slot allocation in ground delay, based on the weather forecasting information. Corolli, Lulli, and Ntaimo (2014) developed two kinds of stochastic programming model for airports, which is designed as a decision tool for flow manager. Gao (2014) proposed a flow management model with multi-restrictions based on analyzing the whole process of the aircraft. Multi flow management methods are used based on fair principles to develop rational strategies. Huang (2019) presented a flight sequencing model for multi airports with minimizing the total weighting delay by analyzing structure and operation features on multi-airport terminal areas.

### 2.2. Slot allocation algorithm

Before the theory of Collaborative Decision Making (CDM) adopted in ATFM, Grover Jack Algorithm is the most popular centralized slot allocation algorithm, which is based on First Come First Serve (FCFS). In order to overcome the demerit of slot missing and double punishment and improve the operational performance, Ball, Chen, Hoffman, and Vossen (2001) developed the algorithm of Ration-By-Schedule (RBS), using the Original Schedule Time of Arrival (OSTA) to replace FCFS. Vossen and Ball (2006) built the model of slot allocation and slot exchange with the cost parameters based on OPTIFLOW model, and solved them with the greedy algorithm. Based on RBS, Zhou (2006) proposed two kinds of slot assignment algorithms based on delay cost parameters to minimize the total delay cost. In recent years, Pellegrini, Bolic, Castelli, and Pesenti (2017) proposed an integer linear programming model (SOSTA) for optimization of the airport with simultaneous allocation of slots. Ribeiro, Jacquillat, and Antunes (2019) developed an original algorithm based on large-scale neighborhood search to solve the slot allocation problem at the largest schedule-coordinated airports, by combining a constructive heuristic and an improvement heuristic algorithm.

In the domain of re-allocation research field for dynamic and stochastic problems, the algorithm of RBS++ and Slot Credit Substitution (SCS) (Sankararaman, 2004) have been developed firstly by FAA based on RBS and Compression algorithm, and adopted in ATFM system in America, in order to solve the dynamic ground delay problems. Next, Vossen and Ball (2006) treated slot exchange as the bargaining problem, and built network flow model to improve the decision support for 2-flights-vs-2-flights slots exchange. Considering airport capacity uncertainty, Wang and Zhao (2020) presented a robust model for simultaneous slot allocation on an airport network in multiple calendar days, in which the robustness is represented by reducing the potential scheduling conflicts in the worst case. Katsigiannis and Zografos (2021) proposed a novel model and solution framework that considering airlines' flexibility preferences and integrated with constraints that enable the dynamic allocation of the airport's resources.

In sum, from the previous researches of ground delay policy, they could be divided into two main aspects: (1) based on the framework of single ATFM measure GDP, some works have tried to improve the performance of calculation efficiency or fairness, by using new ranking rule, slot exchange scheme, and so on; (2) without considering pattern of ATFM measure GDP, some works treated the slot assignment problem as one optimization problem with specific constraints, and solved it by using priority rule or *meta*-heuristic algorithm. However, there are few works conducting slot assignment optimization problems within the framework of existing multiple coupling ATFM measures, which are more complicated to solve but practical in aviation operation.

### 3. Integrated slot allocation algorithm

From previous sections, we have known that, there are some typical existing ATFM measures of ground delay, such as GDP, AFP and MIT. For GDP and AFP, the key character is the constraint of the total capacity of airport or sector per hour usually. For MIT, the constraints are the minimal time or space separations between each pair flights. As mentioned in Section 2, in order to overcome the flaws of the existing solutions, in this work, our target is to develop an integrated slot allocation algorithm, keeping the pattern of existing ATFM measures (pattern means the constraint and parameters) and searching feasible slot assignment solution with an uniform method, to realize the final solution with higher operational efficiency (e.g. lower delay) and applicability.

#### 3.1. Problem description

In Section 1, we have known that, before slot allocation, we have to divide the planning horizon as several time interval. Based on that, each flight will be assigned to a time interval and choose one time point during the time interval, finally one flight will get one time point named CTOT.

After the parameters of one ATFM measure are given, all related flights can be achieved by comparisons with their expected time over (ETO) the airspace and the starting/end time of the measure.

For rule-based slot allocation, how to decide the assignment sequence of all flights has to be well considered, because it will affect the quality and performance of final solutions. Especially when there are plenty of ATFM measures and one flight is impacted by two or more measures, it becomes too difficult to decide which flight has higher priority to assign the slot resource (see Fig. 1).

Next, we propose one heuristics method to generate satisfied solution, including 3 steps: Step 1: Slot division based on minimal-time-interval, in order to increase the time slot flexibility. Step 2: Flight ranking based on multi-dimension flight priority, which includes flight information and state, level of related ATFM measures. Step 3: Solution generation based on flight priority and constraint checking. The outline of flowchart is shown in Fig. 2.

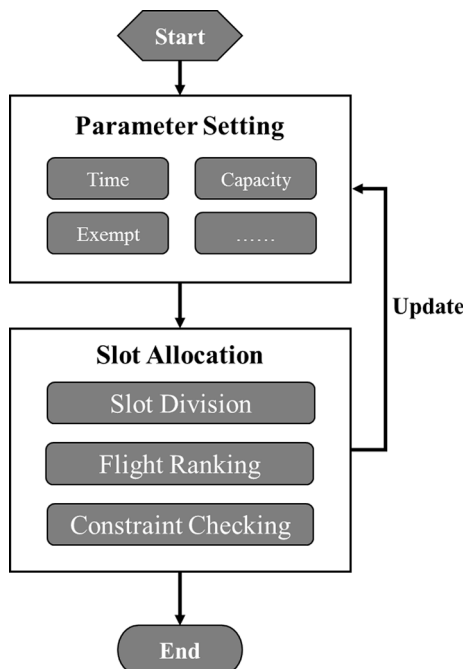


Fig. 1. Outline of solving slot allocation problem in ATFM.

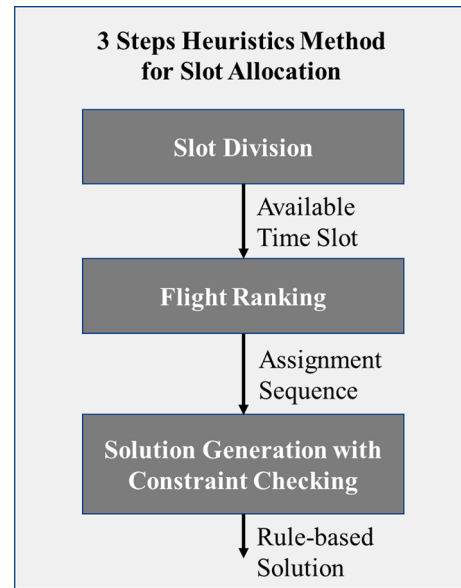


Fig. 2. Outline of 3 steps heuristics method for slot allocation.

#### 3.2. STEP-1: Slot division

Slot division is used to generate several time slots that can be viewed as one resource for flights assignment. In previous research and existing ATFM system, the main method of slot division is equally division. For example, the capacity of one airport decreases to 20 per hour, so that each time slot is  $60/20 = 3$  minutes. Therefore, the time slots are : (00:00–02:59), (03:00–05:59), ....., (57:00–59:59), with totally number of 20.

In Fig. 3, each black dot represents one position that can be assigned to one flight.

The merit of equally division is the distribution of flights are better. Because in each time interval, there is only one time slot, and only one flight can be assigned, so that the expected averaged separation is about 3 minutes. However, flow management should focus on the total capacity and delay, rather than the separation. For example, in the slot made by equally division, 2 flights have the same expect time over (e.g., 03:00), one flight can hold the time slot (03:00–05:59) with no delay, another one need to delay to the next time slot (06:00–08:59) with 3 minutes delay. Furthermore, suppose every time slot has already assigned to one flight, and there is flight  $f$  expected to delay 3 minutes due to airline problem. Two possible solutions can be happened. The flight  $f$  postpones to next time slot, so that all the flights after  $f$  have to move to next time slot in turn. The stability of solution is very low. Another condition is all other flights remain their position, flight  $f$  postpones to the final free time slot, that maybe 2 hours later, so the fairness is low.

In order to overcome the disadvantage, we designed slot division method based on minimal-time-interval. The steps are:

Step 1: According to previous researches and operational experience, for flow management, a time slot with 5 minutes is a suitable way. As a result, there are totally 12 time intervals per hour.

Step 2: We divide the total capacity  $C$  into each 5-minutes-time-

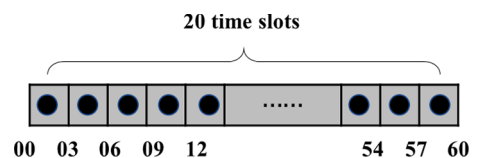


Fig. 3. Slot division based on equally division.

interval, and each time interval could hold more than one flight.

Step 3: Each 5-minutes-time-interval can hold  $C/12$  time slots. If the value is not integer, during each 15 minutes interval (0–15, 15–30, 30–45, 45–60), first 5-minutes-time-interval will get one more flight, else first and second 5-minutes-time-interval will get one more flight, to make the number in each time slot integer (see Fig. 4).

Take the same capacity of 20 for example, each 5-minutes-time-interval can hold  $20/12$ , which is not integer value. For each 15 minutes, the total number is  $20/4 = 5$ . Then during each 15 minutes, first 5-minutes-time-interval can hold 2 time slots and 2 flights, second 5-minutes-time-interval can hold 2 flights, and third 5-minutes-time-interval can hold only 1 flight, as shown in Fig. 5.

Here we explain the reason why we put “extra” flights in the first and second 5-minutes-time-interval. As time goes on, it is very common for some flights delay several minutes to the next time slot. If we put one flights in the first 5-minutes-time-interval, if this flight delay, the total number of flights during corresponding 15 minutes period is same. In practical aviation operation, flow manager can accept this deviation. So, in our method, we take this situation as legal solution and do not need any re-allocation. In other words, this kind of slot division mechanism can improve the stability when some flights postpone to next time slot.

### 3.3. STEP-2: Flight ranking

Flight priority is used to represent the importance of one flight, when they are selected to assign to time resource or time slot. In USA or European, there is one concept named Most Penalizing Regulation (MPR), and one flight's take off time will be calculated by one single ATFM measure, the constraints in other related measures can be omitted. However, in China, due to the limitation and constraints of airspace, 2 or more measures' requirements have to be satisfied at the same time.

As a result, we propose one framework of flight priority calculation, hybrid both flight-related ATFM measures and flight own information.

#### 3.3.1. Bi-level ATFM measure level

In order to mitigate the disruption of new-coming ATFM measures, especially regional (only affect flights in small area) and low important level measures, we propose bi-level ATFM measure framework (see Table 1).

For each ATFM measure, we define outer level  $N$  and inner level  $M$ . Final level is used to represent one measure's level based on outer and inner level.

##### (1) Inner Level

We classify the different inner level of ATFM measures based on the emergency and influence level. The number of measure level can be set as  $N$  (from 1 to  $N$ ), and in order to illustrate the framework more easily, in this work, we set the number of levels as 4, and the description of each level is shown in Table 2.

##### (2) Outer Level

We define different outer level of ATFM measures in order to mitigate the impact of new-coming ATFM measures. Because the measure with lower outer level can not change the departure time of flights that already impacted by higher outer level measure.

For example, there is one flight  $f$ , and one ATFM measure  $X_1$  with outer level 1. After slot allocation, flight  $f$  receives its CTOT as 10:00. Later, another ATFM measure  $X_2$  made with outer level 2, and  $X_2$ 's

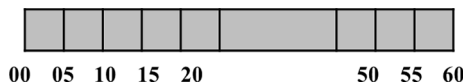


Fig. 4. Minimal-Time-Interval with 5 minutes.

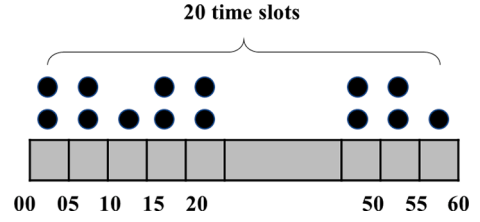


Fig. 5. Slot division based on Minimal-Time-Interval.

Table 1

ATFM measures with bi-level.

Outer Level	Inner Level	Final Level
1	1	1-1
	2	1-2
	.....	.....
	$M_1$	1- $M_1$
2	1	2-1
	.....	.....
	$M_2$	2- $M_2$
	.....	.....
.....	.....	.....
$N$	1	$N$ -1
	.....	.....
	$M_N$	$N$ - $M_N$

Table 2

Inner ATFM measure level.

Level ID	Name	Description
1	Top	The measure with: emergency event, such as earthquake, ground stop (GS) due to equipment breakdown, etc.
2	High	The measure with: impact area is large and the expected delay level of flights are high.
3	Normal	Default level.
4	Low	The measure with: impact area is small.

impact flights contain flight  $f$ . Now flight  $f$  has two measure  $X_1$  and  $X_2$ . Because  $X_2$ 's outer level is lower than  $X_1$ , when we calculate CTOT of flight  $f$ , the constraints of capacity limitation or separation requirement in  $X_2$  are omitted. In other words, flight  $f$  can keep its original CTOT, even after it receives new measure.

#### 3.3.2. Flight information and state

To determine flight priority, if this flight has higher measure level, it will have higher priority. Another factor should be considered is flight information and state.

- Flight information: whether belongs to VIP flights, whether belongs to ATFM measures exempt flight, whether belongs to international flight, etc.
- Flight state: for one airport, states of one flight include not arrive, land, park, push, startup, taxiing, departure, etc.

The information above could be standard to choose some ones as priority flights, so that they can assign time slot earlier than others. In this work, based on the rule of aviation operation and Chinese flow management reality, we select ATFM measures exempt flights and flights after push state as higher priority.

#### 3.3.3. Multi-dimension flight priority

We integrate bi-level ATFM measures levels and flight information and state, to generate multi-dimension priority matrix, as shown in Table 3.

In Table 3, it shows the flight priority matrix. For the same group in same level, finally we compare two flights on Scheduled Time Over



**Table 3**  
Multi-dimension flight priority.

Priority Group			Rank within Group
Outer Level 1	Flight Information Satisfied	Exempt Flights	by STO
	Flight State Satisfied	Departure Flights	by STO
		Push, Startup, Taxing Flights	by STO
	Final Level 1-1		by STO
	Final Level 1-2		by STO
Outer Level 2	.....		.....
	Final Level 1- $M_1$		by STO
	Flight Information Satisfied		by STO
	Flight State Satisfied		by STO
	Final Level 2-1		by STO
.....	.....		.....
	Final Level 2- $M_2$		by STO
	.....		.....

(STO). The earlier STO, the higher priority flight is. For one airport, we take Scheduled Landing Time (SLDT), which could be viewed as one special STO for airports.

Take five flights (Flight A-E) for example. As shown in Table 4, the information of these flights is listed.

Step 1: The highest measure level of 5 flights are 2-3, 2-2, 1-3, 2-2, and 2-2. Flight C belongs to outer level 1, and others belong to outer level 2. So, flight C is selected.

Step 2: In outer level 2, according to status sequence “Exempt, Departure, Push, Startup, Taxing, Others”, select flight A and flight B (A>B).

Step 3: For flight D and flight E, their STOs have to be compared. Flight E's STO is earlier, and the sequence is E > D.

Finally, the priority sequence is C > A > B > E > D.

### 3.4. STEP-3: Solution generation with constraint checking

In Section 3.2 and Section 3.3, the time slot and assignment sequence have been decided. Next step is to allocate time slot to each flight according to the sequence.

As shown in Fig. 6, seeking one available time slot for each flight has 3 parts to check: whether time slot free, capacity limitation satisfied, and separation requirement satisfied. The pseudo code of searching process is given in Fig. 7.

We take the upper black color flight A in Fig. 6 as one example, to explain the solving process of slot allocation. The flight A's ETO is 10:22, it has one ATFM measure MIT with the time separation of 15 minutes. Flight A and the blue colored flights have to satisfy the constraints, while other assigned flights with black color do not need. Because in MIT, only flights belong to the same traffic flow have the separation constraints.

Step 1: Based on the ETO (10:22), firstly, we check the time slot 10:25. There is no free slot. Move to next time slot.

Step 2: Checking time slot 10:30, there is one free slot. Checking the separation constraints. The time separation between flight A and right blue flight is 10 minutes (10:40-10:30), that is smaller than 15 minutes constraint. Move to next time slot.

Step 3: Checking time slot 10:35, separation is not satisfied. Checking time slot 10:40, there is no free time slot. Checking time slot 10:45,

**Table 4**  
Flights information of illustrative example.

Flight	Status	ATFM Measure Level	STO
A	Exempt Flight	2-2, 2-4	10:00, 11:00
B	Taxing Flight	2-2, 2-3	09:30, 10:00
C	-	1-3, 2-4	12:30, 13:40
D	-	2-2	09:30
E	-	2-2	09:20

10:50, separation is not satisfied.

Step 4: Checking time slot 10:55. There is one free time slot, and separation is satisfied. Therefore, the final slot assignment for flight A is 10:55.

### 3.5. Enhanced STEP-2: Meta-heuristic slot allocation algorithm

In Sections 3.2, 3.3 and 3.4, we can get one solution by using one rule-based approach. Next, we try to optimize the solution by using meta-heuristic algorithm.

It is obvious that the assignment sequence of flights is the key point for producing one solution. If one sequence is achieved, based on method in Section 3.2 and Section 3.4, one final solution can be generated.

There have been a great number of heuristic algorithms and meta-heuristic algorithms developed in recent decades, such as generic algorithm (GA), simulated annealing (SA), particle swarm optimization (PSO). For these algorithms, suitable parameters are needed to achieve satisfied solutions, but how to tune the parameters is one critical job. Then, one probability model-based algorithm Estimation of Distribution Algorithm (EDA) (Tian, Hao, & Gen, 2019) was developed. The key idea of EDA is to build an explicit probabilistic model for the distribution of promising solutions found so far, and use the constructed model to guide further search behavior.

Therefore, we take meta-heuristic algorithm EDA as a searching engine, to search and train the optimal assignment sequence, leading to final solution with high performance. As shown in Fig. 8, by using the heuristic method in Sections 3.2 and 3.3, the sequence generated is taken as the initial input for EDA. Hybrid with solution generation and evaluation method in Section 3.4, the meta-heuristic slot allocation algorithm based on EDA aims to improve the optimality of solution and convergence performance.

The key issues of using EDA are how to use probability model to express the distribution of decision variables for one specific application, and how to construct and generate new solutions based on probability model. By adopting EDA, several key aspects are illustrated as following:

#### (1) Objectives and Mathematical Model

Objectives are used to evaluate whether the generated solution is good or not. The possible objectives of ground delay problem could be, for example, minimizing the total delay time or averaged delay time, or minimizing the total cost of waiting time, or maximizing the measures of fairness. In real world problem, one solution should consider more than one single objective (Androutsopoulos, Manousakis, & Madas, 2020). Delay is one important criterion for both airlines company and air traffic manager. Fairness is important for airlines. Without fairness evaluation, some flights could have large delay while other flights have no delay at all. The criterion of fairness is to guarantee flight delays should be as average as possible. As a result, in this work, we take cost of delay and fairness as bi-objective, based on the decision variables of flights being assigned to slots in Eq. (1).

#### - Indices

$i$	Index of flights, $i = 1, \dots, N$
$j$	Index of slots, $j = 1, \dots, M$

#### - Parameters

$N$	Total number of flights
$M$	Total number of slots
$c_i$	Delay cost of flight $i$
$\omega_D$	Weighting of total delay cost objective
$\omega_F$	Weighting of total fairness objective
UC	Unit cost of time delay
$\omega_i^D$	Weighting of delay cost of flight $i$
$\omega_i^F$	Weighting of fairness of flight $i$
$STO_i$	Scheduled time over of flight $i$
$F(X_{ij})$	Fairness measure of slot $j$ assigned to flight $i$

(continued on next page)

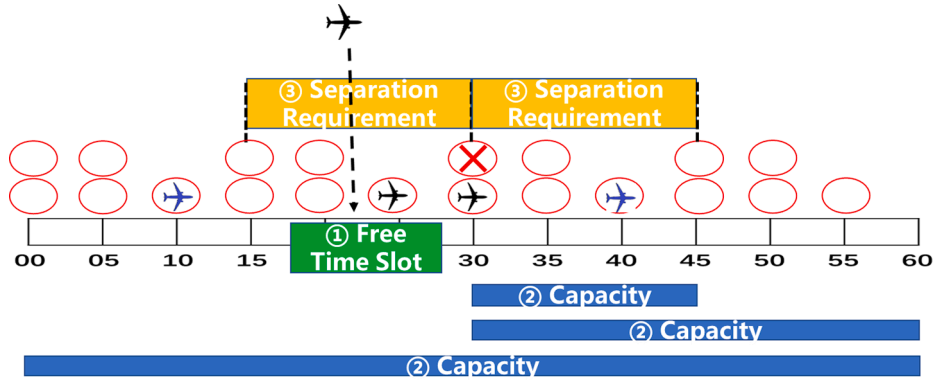


Fig. 6. Constraint checking based on capacity and separation.

### Solution generation with constraint checking

begin

for  $i := 1$  to  $\max\text{FlightRankingNo}$  do

Step 1 find all ATFM measures related and rank them according to the highest outer level,  $X_{i1}, X_{i2}, \dots, X_{ij}, \dots$ ;

for  $j := 1$  to  $\max\text{MeasuresNo}_i$  do

Step 2 Update the ETO for each measure:  $t_{i1}, t_{i2}, \dots, t_{ij}, \dots$ ;

Step 3 Based on the slots already divided, find the time slot  $s_{ij}^k$  include time  $t_{ij}$ ;

Step 4 Check whether time slot  $s_{ij}^k$  has free slot;

Step 5 If satisfied, go to Step 6, else  $k++$  and go to Step 4;

Step 6 Check whether time slot  $s_{ij}^k$  satisfies capacity constraints, including 15-minutes-capacity, 30-minutes-capacity, and 60-minutes-capacity;

Step 7 If all satisfied, go to Step 8, else  $k++$  and go to Step 4;

Step 8 If measure  $X_{ij}$  has no separation requirements for flight  $i$ , go to Step 11; else go to Step 9;

Step 9 Check whether time slot  $s_{ij}^k$  satisfies separation constraints;

Step 10 If all satisfied, go to Step 11, else  $k++$  and go to Step 4;

Step 11 Set the time slot  $s_{ij}^k$  for flight  $i$ , if  $(j>1 \text{ and } s_{ij}^k \neq t_{ij}), j=1$ , else  $j++$ , go to Step 2;

end

end

Fig. 7. Pseudo code of solution generation with constraint checking.

(continued)

$t_j$	Starting time of slot $j$
$c_i^*$	Standard delay cost of flight $i$ , based on the solution made by priority rule
$d(x_{ij})$	Deviation to standard cost, for slot $j$ assigned to flight $i$

$$c_i = \sum_{j=1}^M \omega_i^D \times UC_i \times (t_j - STO_i) \times x_{ij} \quad (3)$$

$$d_i = \max(0, c_i - c_i^*) \quad (4)$$

$$F(X_{ij}) = \sum_{i=1}^N \omega_i^F \times d_i \quad (5)$$

- Decision Variable

$$x_{ij} = \begin{cases} 1, & \text{if flight } i \text{ being assigned to slot } j \\ 0, & \text{else} \end{cases} \quad (1) \quad \sum_{j=1}^M x_{ij} = 1 \quad (6)$$

- Objective

$$\text{Min } \omega_D \times \sum_{i=1}^N c_i + \omega_F \times F(X_{ij}) \quad (2) \quad \sum_{i=1}^N x_{ij} \leq 1 \quad (7)$$

- Subject to

$$(t_j - STO_i) \times x_{ij} \geq 0 \quad (8)$$

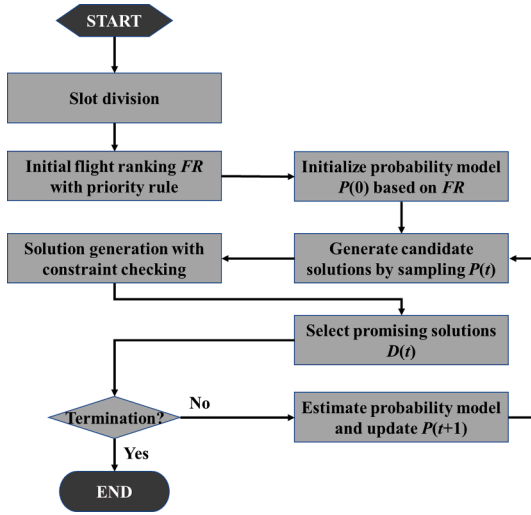


Fig. 8. Flowchart of enhanced meta-heuristic algorithm.

where in Eq. (2), it shows the bi-objective that including minimizing delay cost and fairness measure. Eq. (3) is to calculate the delay cost with the slot and STO. In Eqs. (4) and (5), they are used to calculate the deviation, and total fairness measure value of all flights. The weights in Eqs. (2), (3) and (5) represent the importance of different flights. Without any prior knowledge, usually we can set both weights the same as 1. Eq. (6) shows that one and only one slot time assigned for one flight, and Eq. (7) ensures that for one slot time, there are at most one flight being assigned for it. Eq. (8) represents the constraint of schedule time, for each flight, slot time should be greater or equal to schedule time.

## (2) Probability model

In order to solve a specific problem, one task is to build probability model for that. In slot allocation problems, the target is to decide the slot assignment for each flight. One intuitive way is each factor in probability matrix represents the probability of flight choosing slot. However, there are some disadvantages:

(a) This kind of probability model would generate some illegal solutions; (b) with our method of slot division, in one time interval, there could be several time slots. These slots are same, but it's difficult to accurately model in the matrix and sampling new candidates; (c) two flights may seek the same time slot, one get the time slot, another flight has to find another slot, so the interaction between these flights are obviously. By using some structure, such as tree or Bayesian network, it is one possible way to find the relationship among them, but the calculation time becomes higher and very difficult to decide which kind of structure is suitable for slot allocation problems in ATFM. How to further improve the searching ability will be discussed in future or concurrent work.

Therefore, in this work, the probability model is used to represent and train the assignment sequence of flights. Acting as the sequence optimizer, it is easily to embed into the original solving framework, which is shown in Fig. 8.

Compared with the conventional approaches based on EDA, the decision variables do not need to fully represented by EDA coding. The complete solution includes sequencing and slot allocation. With the help of constraint checking and solution generation in Section 3.4, the probability model of EDA is simplified to assignment sequence only. As a result, we design one matrix to represent the probability of assignment sequence for each flight in Eq. (9).

$$P(t) = \begin{bmatrix} p_{11} & \dots & p_{1N} \\ \dots & p_{ij} & \dots \\ p_{N1} & \dots & p_{NN} \end{bmatrix} \quad (9)$$

where  $p_{ij}$  represents the probability value of flight  $i$  with ranking sequence  $j$ .

Usually, we randomly make the initial probability matrix in 1st generation, or choose every  $p_{ij}$  with the same value  $1/N$ . However, it's not a good initial solution, because we already have one "good enough" solution. In Section 3.3, we have already generate one rule-based flight sequence. Therefore, we generate initial probability matrix based on that sequence.

In the solution given by the rule-based method, the assignment sequence of flight  $i$  is  $j$ . We try to make the probability of choosing near ranking  $j$ th position is higher than others. One way is to generate the probability following the normal distribution, but there is no previous research that has proved this assumption. To simplify this problem, we set  $p_{i,j-1} = p_{i,j} = p_{i,j+1} = 20\%$ , and other positions are randomly generated with the constraint of total probability should be 40%.

## (3) Solution sampling

The assignment sequence could be viewed as independent decision variable. Therefore, when we generate next generation population, for ranking sequence  $j$ , only the information of column  $j$  are needed to be calculated (see Fig. 9).

We use Monto Carol Sampling method to produce new candidate solutions. For generating one individual: (a) Randomly choose sequence  $j$ th, according to the probability of each flight in column  $j$ , randomly select one flight  $i$  following the probability. (b) Remove the flight  $i$  and sequence  $j$ , randomly move to another sequence, choosing one flight, until all the sequence has one flight. (c) During the process of selecting flights, if one flight has been already removed, a new flight will be selected according to the probability, until the new selected flight satisfied.

## 4. Experiment

To examine the practical viability and efficiency of our proposed algorithm, we designed numerical studies on efficacy, stability, searching performance of solutions.

### 4.1. Slot division method comparison

In Section 3.2, we designed one slot division method based on Minimal-Time-Interval. In this section, we compare our method with Equally Division, which is one popular way in ATFM.

We select the flights within 2 hours (18:00–20:00) in Beijing Capital International Airport (ZBAA). We assume that the arrival capacity

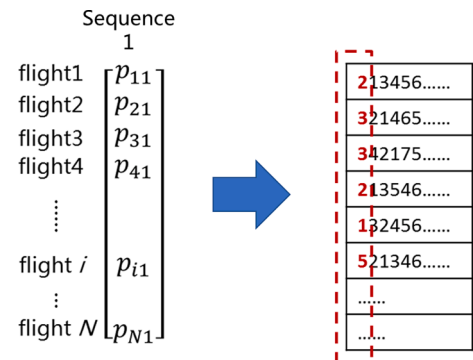


Fig. 9. Probability matrix and candidate solutions.

decrease to 20 per hour. The amount of flights expected arrive at ZBAA during 18:00–20:00 is about 40 per hour. In order to cover different scenarios, we delete parts of arrival flights equally for each time period, and compare two methods based on 15, 20, 25, 30, 35 arrival flights per hour.

From Table 5, it shows that when the amount of expected arrival flights increased, the amount of delayed flights and averaged delay time both increased, also, the percentage of delay flights increased. For the same expected arrival flights per hour, solution made by Minimal-Time-Interval method is better than Equally Division method, because Minimal-Time-Interval method providing flexibility that there can be more than one time slot during the same time interval. Meanwhile, the period of one time slot is longer.

For example, there are 2 flights having the same expected arrival time. By using Minimal-Time-Interval method, no delay occurred. By using Equally Division method, one flight must delay to next time slot with 3 minutes delay.

#### 4.2. Trade-off between fairness and delay cost

In Section 3.3, we proposed one framework of multi-dimension flight priority. In Table 3, it is shown that, if two flights have the same ATFM measures level (both outer and inner level), and the same flight information and state, we have to compare them by STO.

In previous algorithms and existing ATFM systems, one popular way is taking Schedule Off-Block Time (SOBT) as the standard to compare. In this section, we conduct a case study between STO and SOBT.

Take Shanghai Pudong International Airport (ZSPD) for example, 09:00–12:00 are selected as the time horizon for case study.

In Table 6, it shows the comparison results. For the solution based on ranking by STO, amount of delayed flights is larger than by SOBT, but the total delay, max delay and averaged delay are smaller than by SOBT. It is obvious that the solution based on STO is better in fairness.

Fig. 10(a) and (b) represents the results of ranking by STO, and Fig. 10(c) and (d) shows the results by SOBT, and Fig. 10(e) is the results of comparison of by STO and SOBT on averaged delay.

From Fig. 10, we can get the conclusion that, with the method ranking by STO, the averaged delay of flights is smaller, and the distribution is better in fairness, no matter for long distance or short distance.

Fairness is one important objective in slot allocation problems. That's why we finally choose ranking by STO in our approach. It is a satisfied solution as initial input for meta-heuristic optimization algorithm, and to improve the searching performance.

#### 4.3. Optimality and calculation performance

In Section 3.4, we propose EDA being embedded into our method, as one meta-heuristic algorithm to search and optimize the assignment sequence of flights. In this section, we take our previous work MMEDA (Tian et al., 2019) as target meta-heuristic algorithm. It is fair to compare, because MMEDA also takes EDA as searching engine, and

**Table 5**  
Comparison on slot division method.

Arrival Flights (in 2 Hours)	Minimal-Time-Interval		Equally Division	
	Amount of Delayed Flights*	Averaged Delay Time **	Amount of Delayed Flights	Averaged Delay Time
15 × 2	0	0	3	0.3
20 × 2	4	0.34	7	0.53
25 × 2	15	2.2	29	2.82
30 × 2	32	4.25	43	4.55
35 × 2	51	6.71	59	6.94

\*: total flights are during 2 hours; \*\*: unit of delay time is minute.

enhanced with Markov network structure. The convergence speed and optimality are compared with initial solution by using heuristic rule-based method in Section 3.3 and with initial population random generated.

Specifically, in order to fairly compare with other approaches, we choose the same parameter value to reduce the disturbance effect of weight variables. All weights of flight delay and fairness are the same as 1.

In Fig. 11, it shows the convergence performance. With the initial solution made by rule-based method, from the first generation, the optimality is better and convergence speed is fast. In about 250th generation, it achieved the optimal solution. Without rule-based method, it achieved optimal solution in about 450th generation.

Table 7 shows the comparison on calculation time. Our proposal requires extra time to generate initial solution once and final solution for each individual by using heuristic method. MMEDA requires extra time to learn the structure and parameters of Markov network, and sampling new candidate solutions based on probability model and network model. In sum, our proposal has smaller calculation time than MMEDA.

#### 4.4. Stability and applicability

In Section 3.2, we design outer level for each ATFM measure, in order to mitigate the impact of new-coming ATFM measures. One measure with lower outer level cannot break the CTOT of flights belonging to higher outer level. In this section, we evaluate the stability of CTOT with the help of bi-level framework including outer and inner level.

We design a series of ATFM measures with sequence. All these measures occurred one by one, and finally we calculate how many times of slot hop, under the conditions of with and without bi-level framework.

There are totally 5 ATFM measures with 5 different outer levels. Four experiments with different conditions are designed. All experiments have the same ATFM measures occurrence sequence, from A to E.

In Table 8, it shows totally 4 scenarios with different outer and inner levels for ATFM measures A-E.

From Table 9, it shows the comparison results of four scenarios. In the second scenario, the outer levels of five measures are getting higher and higher, it is equivalent to no outer level or having the same outer level in scenario 1, because every new measure occurred, the CTOT has to update if needed. In the third scenario, the outer levels of five measures are getting lower and lower. When new measures occurred, the flights' CTOTs do not need to update, because they have already impacted by higher outer level measures, the constraints in new-coming measures are omitted. In the fourth scenario, when the measure occurs with higher outer level, related flights' CTOTs have to change, and CTOTs remain when lower outer level happened.

As shown in Table 8, the scenario 2 and 3 are extreme situations, and scenario 1 can be viewed as one special case of scenario 4. As a result, scenario 4 is a common situation having new-coming ATFM measures with randomly outer levels. We take the averaged CTOT difference and the deviation to represent the stability performance, and conduct experiments based on scenario 4.

In order to explain the benefits of stability in more detail, the conventional method is also required to be compared. In this experiment, similar to Section 4.3, the conventional method is designed based on MMEDA without bi-level or initial solution, which is to mitigate the differences of searching algorithms, making the comparisons fairer.

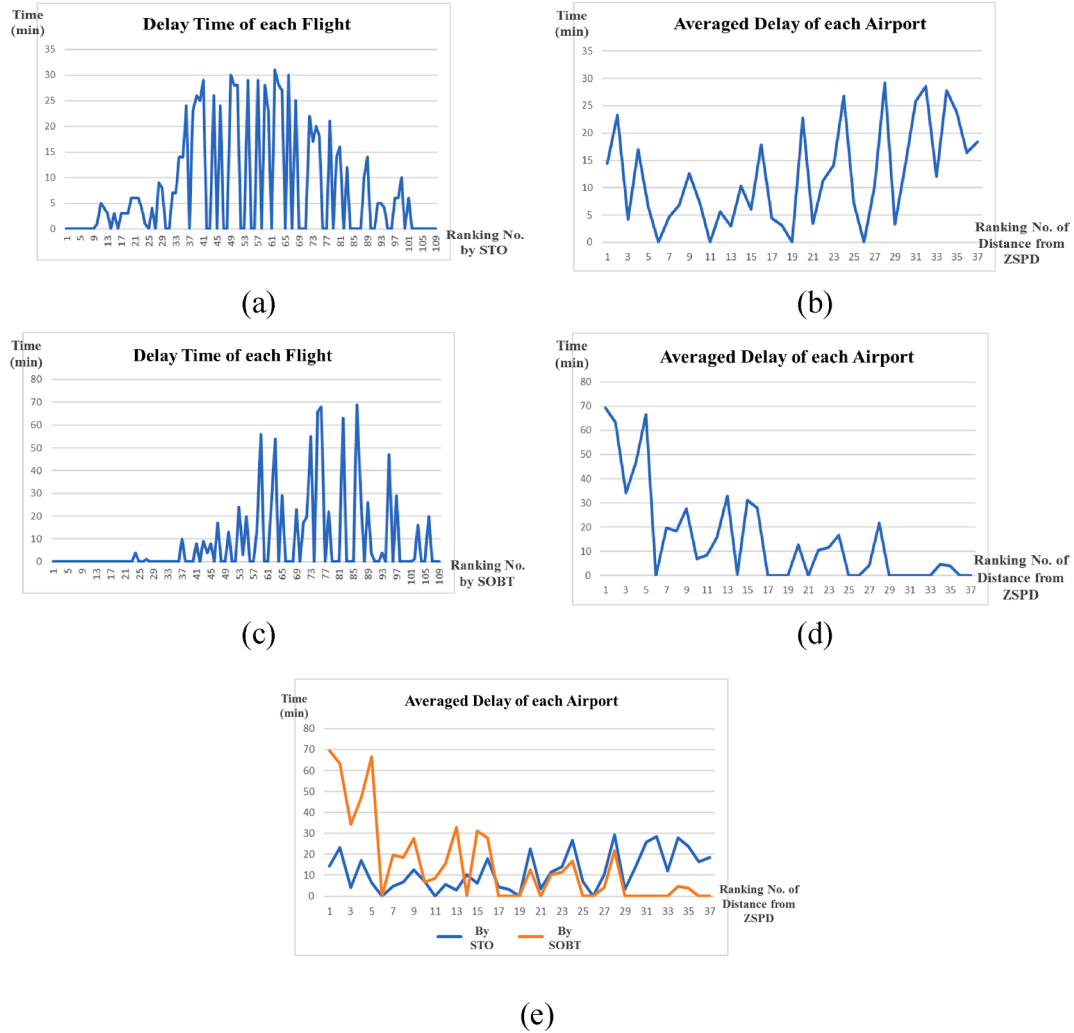
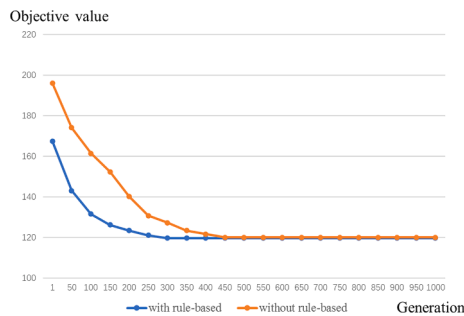
In Table 10, it shows the comparison results of averaged CTOT difference and standard deviation, given by our proposal, our proposal without bi-level pattern, our proposal without rule-based initial solution, and conventional method based on MMEDA. Bi-level pattern is used to distinguish the importance of different flights, and rule-based initial solution provide a stable flight sequence for further evolution, so they are two main aspects to enhance the stability of solutions. From the results in Table 10, the averaged CTOT differences of 4 methods are



**Table 6**

Comparison on ranking by STO and by SOBT.

	Total Amount of Flights	Amount of Delayed Flights	Total Delay Time (Hour)	Max Delay Time (Min)	Averaged Delay Time (Min)
By STO	109	56	13.87	30.93	14.87
By SOBT	109	38	14.63	69.45	23.09

**Fig. 10.** Results of ranking by STO and SOBT.**Fig. 11.** Comparison on convergence performance.

almost the same, but the deviations of method without bi-level and conventional method are smaller. However, when we only count the flights with higher priority (with outer level 1, 2 or 3), the averaged value and standard deviation with bi-level structure is better than other

**Table 7**

Comparison on calculation time (1000 generations).

Calculation time (s)	Our proposal	MMEDA
200 flights	18.4	22.3
1000 flights	87.2	101.5
4000 flights	203.9	245.1

**Table 8**

Experiment description for demonstrating stability.

Scenario	Measure A	Measure B	Measure C	Measure D	Measure E
1	3-4	3-4	3-4	3-4	3-4
2	5-4	4-4	3-4	2-4	1-4
3	1-4	2-4	3-4	4-4	5-4
4	3-4	2-4	4-4	1-4	5-4

**Table 9**  
Comparison on number of times of slot hop.

Scenario	Number of times of slot hop
1	67
2	67
3	0
4	39

**Table 10**  
Comparison on averaged CTOT difference and deviation.

	Averaged CTOT difference A* (min)	Standard deviation A	Averaged CTOT difference B**	Standard deviation B
Our proposal	29.4	18.9	14.5	6.3
Without bi-level	29.2	10.7	28.3	10.4
Without rule-based	29.3	17.8	15.1	6.9
Conventional method	30.1	11.5	29.8	11.3

\*A: for all flights; \*\*B: for higher priority flights with outer level 1, 2, 3.

methods.

In sum, based on the comparisons on slot hop and CTOT difference, it is demonstrated that our proposal having better applicability in daily aviation operation, because it can decrease the number of times of slot hop and protect the higher priority flights getting lower delay.

## 5. Conclusion

Slot allocation is one core issue of research domain in ATFM, and very important for the operational performance in aviation business. For the condition of multiple ATFM measures occurred simultaneously or continuously, there are too many conflicts and disruptions due to coupling measures and flights using the same temp-spatial resources. In order to generate one effective and stable solution while keeping the original pattern of existing ATFM measures, we propose an enhanced *meta*-heuristic algorithm with rule-based heuristic method to solve slot allocation problems uniformly. Firstly, a 3 steps rule-based heuristics method is developed, including new designed slot division method based on Minimal-Time-Interval, flight ranking method based on bi-level ATFM measures, and solution generation based on multiple constraints checking. Next, in order to improve the optimality and flexibility of solutions, Estimation of Distribution Algorithm (EDA) is adopted to optimize the assignment sequence based on probability model. Finally, the experiments are conducted to demonstrate that our approach has high performance on calculation efficiency and can produce robust solutions for re-allocation under the condition of multiple ATFM measures.

In our future work, two research directions could be discussed. First direction is to continue conducting research based on the framework in this work. For example, some knowledge-based *meta*-heuristic algorithms as Markov network-based EDA could be a possible way to enhance the searching ability for complex problems. Second direction is to learn the priority model with, for example, algorithm of random decision forest and convolutional neural networks, that could be adopted to train and generate structure and weighting of priority model.

## CRedit authorship contribution statement

**Jing Tian:** Methodology, Writing – original draft. **Xinchang Hao:**

Methodology, Software. **Jibo Huang:** Data curation. **Jinglei Huang:** Formal analysis. **Mitsuo Gen:** Supervision, Writing – review & editing.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was supported by National Key Research and Development Program of China (2018YFE0208700) and Special Research Project of Chinese Civil Aircraft (MJ-2020-S-03).

## References

- Andreatta, G., & Romanin-Jacur, G. (1987). Aircraft flow management under congestion. *Transportation Science*, 21(4).
- Androutsopoulos, K. N., Manousakis, E. G., & Madas, M. A. (2020). Modeling and solving a bi-objective airport slot scheduling problem. *European Journal of Operational Research*, 284(1), 135–151.
- Ball, M. O., Chen, C.-Y., Hoffman, R., & Vossen, T. (2001). Collaborative decision making in air traffic management: Current and future research directions. *New Concepts and Methods in Air Traffic Management*, 17–30.
- Corolli, L., Lulli, G., & Ntamo, L. (2014). The time slot allocation problem under uncertain capacity. *Transportation Research Part C Emerging Technologies*, 46(46), 16–29.
- Fei, W. (2010). *Research on Slot Allocation Models and Algorithms in Ground Holding Policy*. Doctoral Thesis. Nanjing University of Aeronautics and Astronautics.
- Gao, H. (2014). Modeling and simulation of flow management with multi-restrictions. *Command Information System and Technology*, 5(5).
- Huang, J. (2019). Flight sequencing model for multi-airport terminal area based on delay allocation. *Command Information System and Technology*, 10(1), 37–42.
- Katsigiannis, F. A., & Zografos, K. G. (2021). Optimising airport slot allocation considering flight-scheduling flexibility and total airport capacity constraints. *Transportation Research Part B: Methodological*, 146, 50–87.
- Liu, J., Li, K., Yin, M., Zhu, X., & Han, K. E. (2017). Optimizing key parameters of ground delay program with uncertain airport capacity. *Journal of Advanced Transportation*, 2017, 1–9.
- Milner, J. (1995). *Dynamic slot allocation with airline participation*. Massachusetts Institute of Technology.
- Mukherjee, A., Hansen, M., & Grabbe, S. (2009). Ground delay program planning under uncertainty in airport capacity. *AIAA Guidance, Navigation, and Control Conference*.
- Mukherjee, A., Hansen, M., & Grabbe, S. (2012). Ground delay program planning under uncertainty in airport capacity. *Transportation Planning and Technology*, 35(6), 611–628.
- Pellegrini, P., Bolic, T., Castelli, L., & Pesenti, R. (2017). SOSTA: An effective model for the Simultaneous Optimisation of airport Slot Allocation. *Transportation Research Part E: Logistics and Transportation Review*, 99, 34–53.
- Ribeiro, N. A., Jacquillat, A., & Antunes, A. P. (2019). A large-scale neighborhood search approach to airport slot allocation. *Transportation Science*, 53(6), 1772–1797.
- Ribeiro, N. A., Jacquillat, A., Antunes, A. P., Odoni, A. R., & Pita, J. P. (2018). An optimization approach for airport slot allocation under IATA guidelines. *Transportation Research Part B: Methodological*, 112, 132–156.
- Sankararaman, R. (2004). Impact assessment of dynamic slot exchange in air traffic management. *Diss.*
- Tian, J., Hao, X., & Gen, M. (2019). A hybrid multi-objective EDA for robust resource constraint project scheduling with uncertainty. *Computers & Industrial Engineering*, 130, 317–326.
- Vossen, T. W. M., & Ball, M. O. (2006). Slot trading opportunities in collaborative ground delay programs. *Transportation Science*, 40(1), 29–43.
- Vossen, T., & Ball, M. (2006). Optimization and mediated bartering models for ground delay programs. *Naval Research Logistics*, 53(1), 75–90.
- Vranas, P. B. M., Bertsimas, D., & Odoni, A. R. (1994). Dynamic ground-holding policies for a network of airports. *Transportation Science*, 28(4), 275–291.
- Wang, D., & Zhao, Q. (2020). A simultaneous optimization model for airport network slot allocation under uncertain capacity. *Sustainability*, 12(14), 5512.
- Zhou, Z. (2006). Slots allocation in CDM GDP. *Journal of Beijing University of Aeronautics and Astronautics*, 32(9).