

ZHIGANG SU, Prof.¹
 E-mail: srsu@vip.sina.com
 MENGQI QIU, M.Eng.¹
 E-mail: mqqiu_77@sina.com

¹ Sino-European Institute of Aviation Engineering
 Civil Aviation University of China
 No. 2898, Jinbei Road, Dongli District, Tianjin 300300,
 China

Traffic Planning
 Original Scientific Paper
 Submitted: 17 Mar. 2018
 Accepted: 24 June 2019

AIRPORT SURFACE MODELLING AND SIMULATION BASED ON TIMED COLOURED PETRI NET

ABSTRACT

In order to satisfy the requirements of International Civil Aviation Organization (ICAO) for aircraft taxi route planning in Advanced Surface Movement Guidance and Control System (A-SMGCS), an airport surface operation modelling and simulation approach based on timed and coloured Petri net is presented. According to the layout of the airport surface and the features of surface operation units, a static Petri net model of the airport surface is established. On this basis, in line with the requirements on the aircraft taxiing velocity in ICAO DOC 9830, the dynamic Petri net model of the airport surface operation is established by adding the time attribute to the static model. Additionally, the method of defining the capacity of airport operation unit place is proposed and the constraints of the airport surface operation are incorporated using Petri net elements. Unlike other papers in the field, the airport surface Petri net model established in this paper can simulate conflict-free taxiing using a Petri net simulator without relying on other model-independent algorithms. Based on the CPN Tools software, taking Toulouse Airport as an example, the validity of the model has been verified by comparing the model running data with real flight data.

KEY WORDS

Petri net; airport surface model; conflict-free taxiing; taxi route planning;

1. INTRODUCTION

Airport is a complex system with a large movement area. Under various control rules, the occupancy and release of airport surface resources by aircraft change at any time. Considering that there are usually several aircraft taxiing simultaneously at the airport, the modelling and simulation of airport surface operation should not only consider the static factors such as distance and time of taxiing, but it should also avoid conflict between aircraft in operation. Thus, it is difficult to describe and solve all kinds of problems using a simple model.

Airport Collaborative Decision Making (A-CMD) concept proposed by EUROCONTROL aims at improving Air Traffic Flow and Capacity Management (AT-FCM) at airports by reducing delays, improving the

predictability of events and optimising the utilisation of resource, which is divided in six elements: information sharing, milestone approach, variable taxiing time, pre-departure sequencing, adverse conditions, collaborative management of flight updates [1]. Considering microscopically, several taxiing time calculation methods for different kinds of airports are proposed in variable taxiing time part, while putting emphasis on the accurate taxiing time estimation instead of a taxiing route optimization, which is mainly the focus of this paper. Besides, the "First Come First Served" principle is replaced by a more efficient and optimal departure order based on the traffic situation on the apron in pre-departure sequencing part. However, only the departure aircraft are taken into account as the objects in sequencing.

Based on the collaborative decision-making concept, multi-agent principle and game theory are applied to the research of ground operation modelling and optimization. Aircraft (pilots), ground controllers and operation controllers are described as agents by Nakamura et al. and Tang et al. [2, 3]. The interactions of aircraft agent with other agents in different states from landing to take-off are modelled and validated. Game theory is applied in Ribeiro's research, in which aircraft are represented as players in the negotiation process for slot allocation in order to get the stable departure sequencing [4]. The previous work constructs general frameworks of ground operation while detailed implement approaches are ignored; for example, the methods to plan a taxiing route. The overall framework proposed by Ravizza et al. combines the runway sequencing problem and the ground movement problem, aiming for better global solutions [5]. A sequential algorithm is described based on the graph theoretical concepts and can include several restrictions. However, the definition that conflicts are avoided by preventing any two aircraft from using the same edge simultaneously is not accurate. Besides, SIMMOD, which has been developed by the US Federal Aviation Administration (FAA), is a widely used model for detailed and dynamic simulation of aircraft operations [6]. It is a discrete-event simulation based on

a node-link model of airfields. The event schedule is determined stochastically with event queues without checking the interface of aircraft in different path segments in lateral directions. These features can be limitations of SIMMOD for simulating local interference of aircraft precisely.

Petri net is a widely used modelling method in the field of automatic manufacturing, which is usually used in flow control and workshop scheduling. The High Level Petri net is an extension of the basic Petri net proposed by Carl Adam Petri in 1962, and the Coloured Petri net (CPN) is one of the High Level Petri nets [7-9]. It is a graphical language for building models of systems of concurrent and discrete events. The model is allowed to use individual tokens, which carry valid information to decide when to fire transitions, so that tokens may be transferred in places with particular type, called colour set, which is defined by the Standard ML functional programming language. The CPN modelling based on the state instead of event allows the creation of complex models such as airport models. Aircraft arrival and departure flows have been simulated in many documents [10-14]. Among them, Davidrajuh and Lin studied airport planning in northern Norway by taking different airport flight data as the input of Petri net and examining the Petri net operation condition [11]. Skorupski ran Petri net with different input streams to estimate the airport capacity and proposed to increase the capacity by sorting the flights [12]. Kovacs et al. studied the effect of runway availability on airport capacity by comparing Petri net running conditions with different runway closures [13]. Shortle et al. calculated the number of runway conflicts using Petri net in order to estimate the possibility of aircraft collision when landing [14]. Route planning based on Petri net is a new idea. Dezan et al. combined Petri net model with Genetic Algorithm (GA) and applied it to urban traffic control [15-17]. The Petri net model was used as a fitness function and the urban traffic flow was optimized by analysing the number of vehicles in the traffic network at different moments. Huang et al. solved the shortest directed path by using directed graphic and simulator of Petri net, but only static properties were considered [18]. Wang et al. established a Petri net model of airport surface activities with related specifications. The static pre-selected path was used as a feasible solution set to obtain the optimal taxi path [19]. Tang et al. established the Petri net model of airport surface and described the runway capacity and conflicts condition by inequality constraints that the model elements should satisfy. However, the constraints that should be satisfied when two aircraft are approaching head-on and crossing were both described as capacity constraints of corresponding units, which is not accurate [20]. Zhu et al. built a surface operation model by extended timed place Petri net and used GA to solve it, then proposed a control strategy for taxiway conflict [21-23].

In the previous research of airport surface modelling by Petri net, some of them made full use of Petri net inherent features but the airport operation flow model was built without considering the layout of the airport and the surface control rules. The taxi route planning on an airport surface has not been studied. Although some of them did research on airport surface route planning, the applications of Petri net model were only theoretical in nature, rather than effectively running in practice. In addition, although Petri net was used to describe the layout and taxiing rules of airports in theoretical research, the representation of some constraints was not accurate and the method to resolve conflict by Petri net itself was not provided. In this paper, based on the airport surface layout Petri net model, the conflict-free taxiing of aircraft is achieved using Petri net own elements to restrain the aircraft taxiing.

The rest of this paper is organized as follows: In Section 2, the model constraints are discussed. In Section 3, according to the mapping relationship between Petri net's basic elements and airport operating units, combined with the features of Petri net's operation, a static Petri net model of airport surface is established. In Section 4, timed factors are added to the static model and a dynamic Petri net model of airport surface is established while considering the capacity constraints and airport surface operation rules. In Section 5, the validity of the model is verified using Toulouse airport as an example. Section 6 provides a conclusion.

2. MODEL CONSTRAINTS

In the following sections, an airport surface operation model is built. It aims at seeking an optimal solution for taxiing route combination of aircraft on the surface, which is beneficial to the ground movement optimization. Therefore, this paper focuses mainly on the modelling and simulation of free-conflict taxiing of aircraft in respecting the taxiing rules instead of on the estimation of accurate taxiing time.

In the process of model construction, the following constraints should be made:

- 1) do not consider the effect of aircraft types on longitudinal separation and on the selection of taxiing route;
- 2) consider that only one operation direction of runway is available at one period of time;
- 3) do not consider the effect of extreme meteorological condition on taxiing.

3. AIRPORT SURFACE STATIC MODEL

The airport surface is a complex system consisting of aircraft, runways, taxiways, terminals, parking, etc. According to the basic definition of Petri net, the mapping relationship between basic airport operating units and basic Petri net elements can be established.

Token represents the aircraft, which can move in places through transitions.

Place represents runway, taxiway and parking, usually indicated by p . For these places, the colour set is AIRCRAFT. However, different types of operating units should be considered individually with respect to their own characteristics. Taxiways can be divided into straight parts and intersections. For the taxiway area shown in *Figure 1a*, if p_1 is bidirectional, the corresponding Petri net model is shown in *Figure 1b*, where the circle represents place, rectangle represents transition, t_r represents that the aircraft taxiing from p to p_1 and t_1 represents that the aircraft taxiing from p_1 to p . If p_1 is unidirectional from left to right, the corresponding Petri net model is shown in *Figure 1c*. As for runway area, usually the taxiway connections with the ends of the runway are used both for taxiing into and out of the runway so that they are bidirectional, while the other connections with the middle of the runway are unidirectional. Usually, an airport contains several aprons which contain many parking places. The model would be extremely complicated if each parking place was represented by a separate place, thus parking places that meet the following conditions are classified into the same place in order to simplify the model:

- 1) located at the same apron;
- 2) connected with the same taxi-in linking and taxi-out linking.

Additionally, the arrival place and the departure place are added to the model for the aircraft in arrival and departure states.

Transition represents the activities of aircraft between different areas, usually indicated by t . When an aircraft operates on the airport surface, it carries valid information including serial number, flight ID, arrival or departure state and parking place name, as displayed by the small circle shown on the left in *Figure 2a*. At this time, as there is a token in p_1 , t_1 is enabled. When t_1 is fired, a token will be subtracted from the input place of t_1 and it will be determined whether it should be added to the output place based on the statement on the output arc, which means that it is possible to achieve the movement of the token in the places with colour set of AIRCRAFT as shown in *Figure 2a* by matching the relevant information of the aircraft. However, due to the characteristics for firing transitions [24], flight information might be lost due to mismatching of relevant information on the aircraft and on the output arc of transition during the movement as shown in *Figure 2b*.

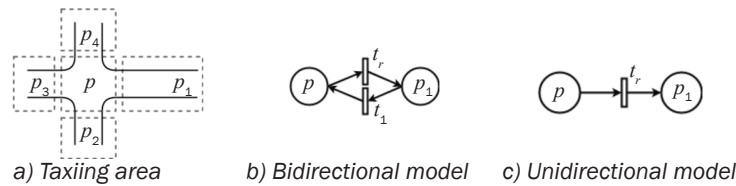


Figure 1 - Petri net model diagram for the taxiing area

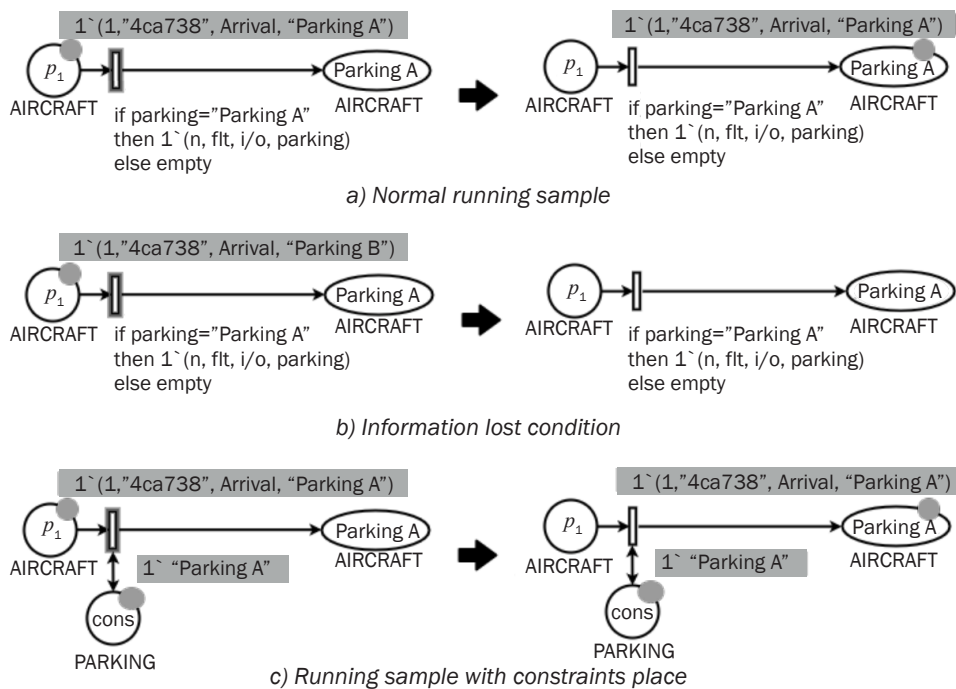


Figure 2 - Petri net running samples

Fortunately, the above problem can be resolved by adding some constraint places (cons) in Figure 3c, to the layout model. As the taxiing direction of aircraft mainly depends on arrival or departure state and parking place in the valid information carried by the tokens, the colour set of constraint places can be divided into three categories: (1) I/O; (2) PARKING; (3) I/O PARKING.

The colour set of constraint place in Figure 2c is PARKING, which means the information of the token in constraint place is the parking name. Only if the information of the token in constraint place matches the information of the token in p_1 , t_1 will be enabled, then the movement of the token shown in Figure 2c can be achieved. When the information of the tokens in these two input places of t_1 is mismatching, the token in p_1 will remain in it instead of disappearing, which will not result in the loss of information. At the moment, if there is any other output transition of p_1 , the token in p_1 will find another possible export according to the above rules. As a result, in the combination with the two methods shown in Figure 2a and Figure 2c it is possible to achieve the directional taxiing of the aircraft token to the target place.

4. AIRPORT SURFACE DYNAMIC MODEL

4.1 Timed dynamic model

A basic dynamic model can be created by adding a time factor to the static model so as to describe the operating state at different moments. Taxiing time is usually decided by taxiing velocity. According to the requirements on aircraft velocity in ICAO DOC 9830, the following velocity values should be available in the range of \pm kts: (1) 0 kts to 80 kts at runway exit; (2) 0 kts to 50 kts on straight runway; (3) 0 kts to 20 kts at the turning area; (4) 0 kts to 10 kts on parking taxiways. As for the aircraft, in good visual conditions, they are allowed to exit the runway at a maximum velocity of $v_r = 50$ kts, to taxi at a maximum velocity of $v_s = 30$ kts on a straight taxiway, to reduce to $v_t = 10$ kts in the turning area and the taxiways that are too short for velocity to accelerate or decelerate to the specified value, and to reduce to $v_0 = 0$ kts waiting for the take-off at the entrance of the runway. In order to ensure the continuity of acceleration and deceleration process, the acceleration of $a = \pm 2m/s^2$ is taken.

According to the change of aircraft velocity in different taxiing areas, the taxiing time τ_p at the corresponding place p could be estimated when given the length of taxiing place l_p . If p is an intersection and the aircraft makes a turn, its taxiing time is $\tau_p = \frac{l_p}{v_t}$. When the aircraft makes straight taxiing at the intersection, its taxiing time is $\tau_p = \frac{l_p}{v_s}$. If p is a straight part of the taxiway and the aircraft makes straight taxi-

ing at the adjacent places of p , its taxiing time is also $\tau_p = \frac{l_p}{v_s}$. When an aircraft makes straight taxiing and makes a turn at the adjacent places of p , respectively, which means there is an acceleration (or deceleration) process in p , its taxiing time can be calculated with Equation 1.

$$\tau_p = \frac{v_s - v_t}{a} + \frac{l_p - \frac{v_s^2 - v_t^2}{2a}}{v_s} \tag{1}$$

When an aircraft makes a turn at the adjacent places of p , which means there are both acceleration and deceleration processes in p , its taxiing time can be calculated with Equation 2.

$$\tau_p = \frac{2 \cdot (v_s - v_t)}{a} - \frac{l_p - \frac{v_s^2 - v_t^2}{a}}{v_s} \tag{2}$$

The acceleration (or deceleration) process is considered as a uniform acceleration (or deceleration) movement. Therefore, the relationships in velocity, distance and time are shown in Equation 3.

$$\begin{cases} v^2 - v_0^2 = 2a \cdot s \\ v = v_0 + a \cdot t \end{cases} \tag{3}$$

If the length of taxiway connection with runway is too short for the aircraft to decelerate to the required velocity, the velocity continuity has the priority.

Timed dynamic model means that the token is marked with a time stamp whose value consists of the following three parts: (1) Initial value of the token time stamp, which represents the time that the token is ready to be used; (2) Time attribute of transition, which represents the time that the transition is ready to be fired; (3) Time attribute of output arc of transition, which represents the time needed to complete the taxiing represented by this transition. As for the aircraft token, the initial time value indicates the time when aircraft enters the airport operation area. As for time attribute of transition, the default value is 0, which indicates that the taxiway is available at any time. Meanwhile, this value can be set as infinity to indicate that the taxiway is unusable. During taxiing, when the aircraft token goes through different transitions, the time attribute of the output arc will be added to the time stamp of the aircraft token. The final value in time stamp of the aircraft token is the time when the aircraft arrives at the target place.

4.2 Capacity constraints

In order to ensure the operational safety of the airport surface, there is a maximum limit of the number of aircraft for each operation unit, which corresponds to the definition of capacity of place in Petri net. The capacity of place p is indicated by $K(p)$.

When there is already an aircraft running on the runway, other aircraft are not permitted to land on or take off from it. Therefore, the runway place p_{rwy} should meet Equation 4.

$$K(p_{rwy}) = 1 \tag{4}$$

Additionally, when there is an aircraft at the final approach and about to land, other aircraft can only wait outside the runway and cannot enter the runway waiting for the take-off or to cross the runway. Runway place p_{rwy} and approach place p_{app} should satisfy Equation 5.

$$K(p_{rwy}) + K(p_{app}) = 1 \tag{5}$$

Intersection place p_{int} should meet Equation 6.

$$K(p_{int}) = 1 \tag{6}$$

As for straight taxiway, longitudinal separation S of aircraft when taxiing should be considered.

According to the *Flight basic rules of the People's Republic of China*, when two or more aircraft are in follow-up taxiing, the latter should not exceed the former one and the latter shall not be less than $S_d=50$ m away from the former. Additionally, considering the length of the aircraft L_a , the minimum spacing between two aircraft is $S_1=S_d+L_a$.

However, according to the requirements for longitudinal separation in ICAO DOC 9830, the following parameters should be considered.

- 1) The detection distance S_0 , which means the distance between two successive aircraft throughout the time that the pilots, controllers and A-SMGCS react.
- 2) The braking distance S_v , which means the distance that the aircraft needs in order to come to a complete stop.
- 3) The safety margin S_s which means the minimum distance needed to be maintained between two aircraft, excluding the exhaust effect of jet.
- 4) Length of the aircraft L_a and exhaust margin L_j , which means the distance needed to avoid the exhaust effect of the jet engine.

Hence, the longitudinal separation is S_2 in Equation 7.

$$S_2 = S_0 + S_v + S_s + S_p \tag{7}$$

where $S_p=L_a+L_j$ and the specific parameters are shown in Figure 3.

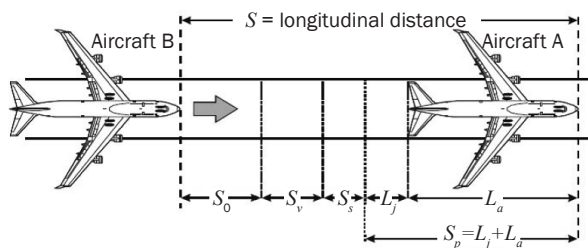


Figure 3 – Parameters for longitudinal distance

For the detection distance, based on the aircraft velocity model and the requirements of A-SMGCS, the following parameter values are given: pilot's reaction time $\tau_{pir}=1$ s, controller's reaction time $\tau_{cor}=1$ s, system's reaction time $\tau_{sys}=2$ s, safety reaction time $\tau_{sar}=1$ s. So the total reaction time is $\tau_{reac}=5$ s. The detection distance S_0 can be calculated according to Equation 8.

$$S_0 = v_s \cdot \tau_{reac} \tag{8}$$

Considering the aircraft velocity reduced from v_s to 0, the braking distance S_v can be calculated from Equation 1. Moreover, it is said in Section 4.5.4 of ICAO DOC 9830 that after the aircraft is stopped, the system should provide a minimum separation (S_s+L_j) of approximately 60 m to 15 m. To ensure safety, the separation is taken as $S_s+L_j=60$ m. Therefore, according to the above parameters, the minimum longitudinal separation between two aircraft S_2 can be calculated by Equation 7.

For safety reasons, the final longitudinal separation S is the maximum value of S_1 and S_2 as shown in Equation 9.

$$S = \max\{S_1, S_2\} \tag{9}$$

After determining the minimum longitudinal separation, the place capacity is defined as follows:

- 1) $\forall p \in P$, if $lp \leq S$, then $K(p) = 1$;
- 2) For any n adjacent places $\{p_1, p_2, \dots, p_n\}$, $n \in \mathbb{N}^*$, if $\sum_{k=1}^n l(p_k) \leq S$, then $\sum_{k=1}^n K(p_k) \leq 1$;
- 3) $\forall p \in P$, $n \in \mathbb{N}^*$, $n \cdot S < lp$, if there exists a place p^* that is an adjacent place of p , and they satisfy $l_p + l_{p^*} \geq (n+1) \cdot S$, then $K(p) = n+1$; otherwise $K(p) = n$.

In CPN Tools, there is no definition of capacity, so capacity constraints need to be achieved by anti-place which is a concept in Petri net. Let p and p' be two places, if the input transition of p equals the output transition of p' and the output transition of p equals the input transition of p' , then p' is the anti-place of p . In the airport operation model, anti-place is a kind of constraint place with the colour set of UNIT where the tokens serve just for counting purposes, without any real meaning. The definition of anti-place and the rules for firing a transition show that the sum of the numbers of a place and its anti-place is a fixed value, which is the capacity of the place as well as the number of tokens in the anti-place at the initial time. As shown in Figure 4, p' is the anti-place of p and the capacity of p is 2.

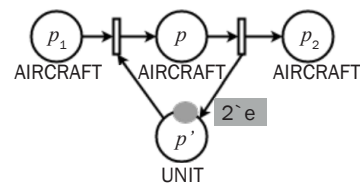


Figure 4 – Anti-place diagram

4.3 Taxiing rules constraints

Except for the capacity constraints of operating units, the relations with other aircraft on the surface should be taken into consideration when taxiing, so as to avoid surface conflicts.

- 1) When there is no aircraft on the runway, the aircraft in the final approach takes priority.
- 2) When an aircraft needs to taxi across the runway, it is necessary to ensure that there is no aircraft on the runway or on the final approach phase.
- 3) When two aircraft approach head-on, if neither of them is at the parking area, the rule of “First In First Out” (FIFO) shall be respected; else, the aircraft at the parking area has the priority to taxi.
- 4) When two aircraft are on a converging course, the one which has the other on its right shall stop taxiing and give way to the other.

All the above rules could be realized by the inhibitor arc which is a concept in the Petri net. If a transition is connected to an inhibitor arc, one of the conditions for the transition to be enabled is that there is no token in the input place to which the inhibitor arc is connected. Accordingly, the models for the above four rules are shown in Figure 5, respectively.

Figure 5a corresponds to Rule 1. When there are aircraft tokens presented both in the *approach* place and the *taxiway* place, as the inhibitor arc connects *approach* with t_1 , t_1 is not enabled, but t_2 is enabled at this time. Therefore, the token in *approach* could move to the runway. After that, there is no token in *approach* and t_1 becomes enabled. All this shows that the approaching aircraft has priority over the aircraft that is to taxi to the runway for the take-off. Figure 5b corresponds to Rule 2. One point that needs particular attention is the connection type. While the places *taxi_in* and *taxi_out* indicate the

taxiways connected to both sides of the runway, *taxi_in* does not directly connect to the runway with the arc in the figure because there is a fundamental difference between using the runway for take-off (or landing) and crossing the runway in real taxiing. Their duration is totally different. Therefore, it is more logical to connect *taxi_in* with *taxi_out*. However, *taxi_in* is connected to *taxi_out* with inhibitor arc for constraints control, which means that only when there is no aircraft in *approach* and on the *runway*, t_3 is enabled and tokens could run from *taxi_in* to *taxi_out*. Figure 5c corresponds to Rule 3. When there are aircraft tokens presented both in parking and p_2 , t_3 could be enabled only after the token in parking has taxied out, which means that the token in parking has priority to that in p_2 . Figure 5d corresponds to Rule 4. According to the same basic principle, the token in the left place has priority to that in the right.

Using the above methods, the potential conflict between aircraft transforms into a waiting time, which achieves the modelling and simulation of conflict-free taxiing on the airport surface.

5. MODEL VERIFICATION

Based on CPN Tools, a case study of modelling and simulation experiments of timed coloured Petri net on Toulouse airport surface have been carried out. At first, the parking places have been simplified according to the method in Section 2. Eighty-four parking places with detailed information are simplified into 18 parking places and the other aprons without detailed information are divided into four places, totalling 22 parking places on the surface. A part of the Toulouse airport surface layout shown in Figure 6a is clearly represented by the airport layout Petri net model as shown in Figure 6b,

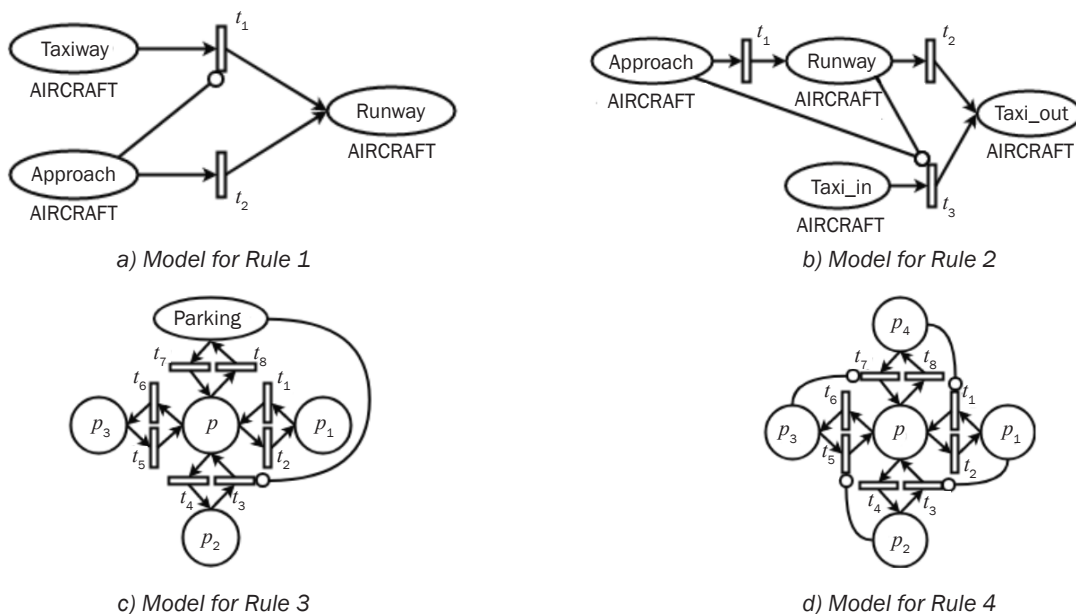
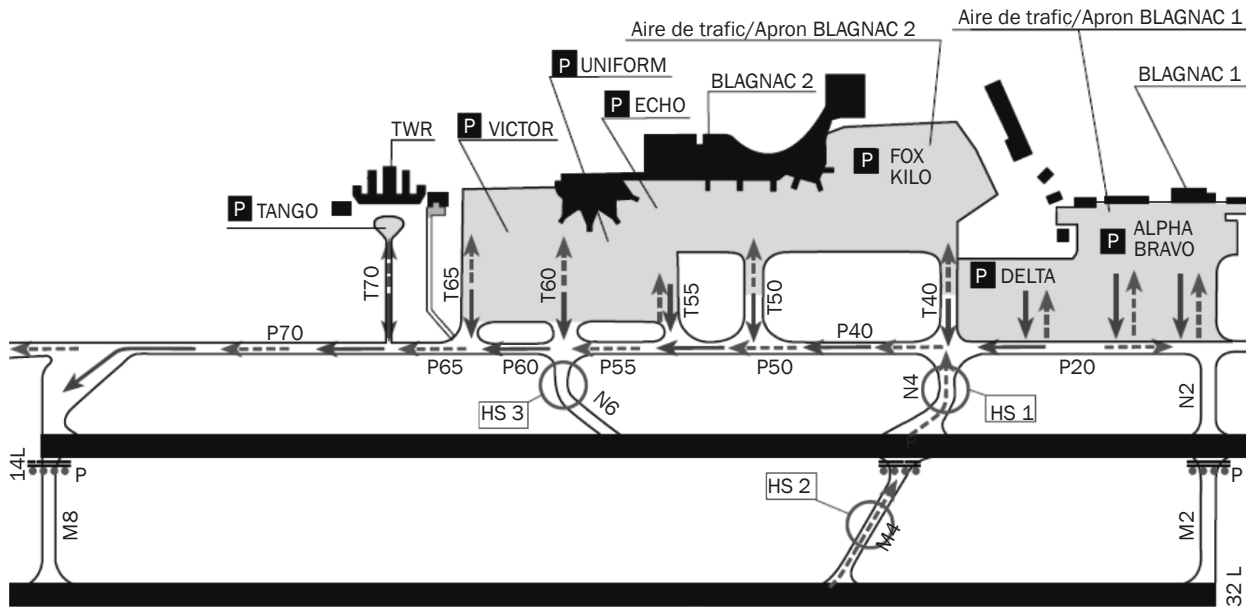
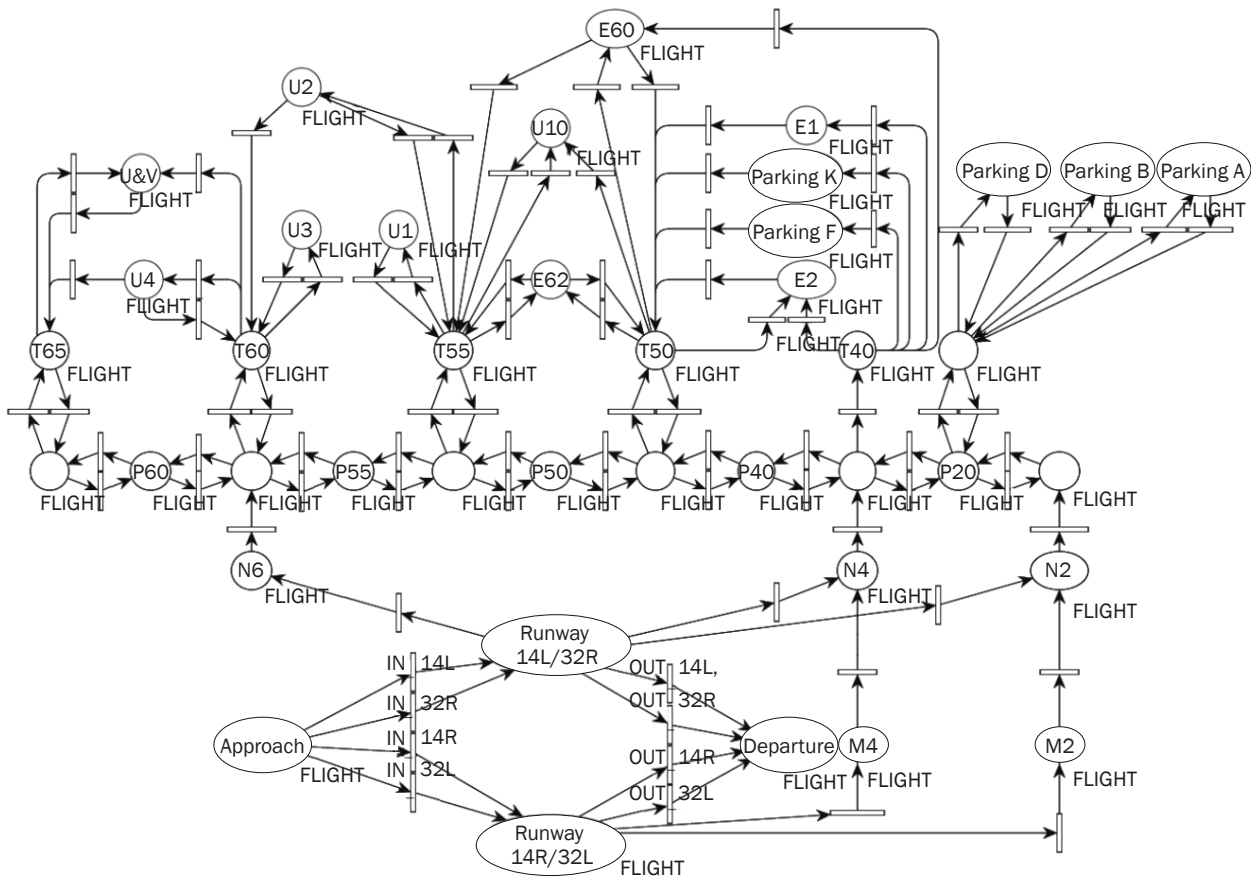


Figure 5 – Petri net models for taxiing rules



a) Blagnac airport map



b) Blagnac airport layout Petri net model

Figure 6 – Petri net model diagram of Blagnac airport surface

where circles represent different areas, rectangles represent the possible activities and arrows represent possible movement directions.

Based on the Blagnac airport surface static Petri net model, with respect to the surface operation constraints described in Section 2 and Section 3, and in-putting relevant aircraft taxiing information, the Petri net simulator in CPN Tools can be used to simulate aircraft taxiing routes. The running process could be saved as a running report in text format so as to analyse the process and result. To verify the effectiveness of the model in taxiing route planning, static verification and dynamic verification of the established Toulouse airport operation model are carried out.

5.1 Static verification

Static verification means the process when the aircraft taxis from one place to another without considering the effect of other aircraft on the surface and the related airport surface operation constraints.

The taxiing information of flights at different time periods of the day from July 7, 2017 to July 11, 2017 was collected based on the information on the flightra-dar24 website platform. In order to make sure of the integrity and usability of each taxiing route data, 30 flights' data are counted and classified according to the taxiing path. The results are shown in Table 1.

In Table 1, the surface state indicates whether other aircraft have an impact on the taxiing route, in which "1" is unaffected and "0" is affected. According to the definition of static verification, when there is another aircraft on the surface, it may result in a waiting time for the target aircraft, while here only one flight is set in the model and the model running time is for an un-impeded route. Therefore, the real flight taxiing time with a surface state of "1" can be used to compare it with the runtime of the Petri net model. The flights' taxiing time of taxiing path numbered 14, 15, 16 and 17 are still much different from the corresponding Petri net runtime. This is due to the fact that the taxiing time of aircraft in the Petri net is calculated based on

Table 1 – Aircraft taxiing timetable

No.	Path	Petri net running time τ_0 [s]	Real flight operation time τ_{real} [s]	Surface state
1	C-T10-P10-P20-P40-P50-P55-P60-P65-P70-N8-M8	312	314	1
2	M2-N2-P10-T10-M	200	265	1
3	N2-P20-T40-K	156	489	0
			439	1
			294	1
			188	1
4	F-T50-P50-P55-N6	206	217	1
5	F-T50-P50-P55-P60-P65-P70-N8	261	251	1
6	E2-T50-P50-P55-P60-P65-P70-N8-M8	202	371	0
			227	1
7	E2-T50-P50-P55-N6	147	166	1
8	M2-N2-P20-T40-E2	230	252	1
9	E62-T55-P55-P60-P65-P70-N8-M8	205	225	1
			205	1
10	U&V-T65-P65-P70-N8	190	286	1
			256	1
			186	1
11	M2-N2-P20-P40-P50-P55-P60-T65-U&V	270	259	1
12	T60-P60-P65-P70-N8-M8	160	153	1
13	N1-P10-P20-T40-E60	278	289	1
14	N2-P20-P40-T50-E62	192	499	1
			285	1
			330	1
15	U2-T55-P50-P40-P20-P10-N1	210	685	0
			572	0
			475	1
16	N1-P10-P20-P40-P50-P55-T60-U4	273	639	0
			525	1
17	N2-P20-P40-E2	206	518	1
			511	1

the velocity model established in Section 2, where the maximum taxiing velocity is 55 kts and the average velocity is 30 kts, while some of the above flights are executed at the local time of 4 a.m. to 5 a.m., when the taxiing speed is slower than during the day. The real maximum taxiing speed is 30 kts and the real average speed is about 16 kts.

Considering the factor of taxiing velocity, the average taxiing velocities of aircraft with numbers 14, 15, 16, 17 and surface state "1" are counted and time τ_s needed to taxi along these paths with the aircraft model velocity $v_s=30$ kts is calculated according to Equation 10. The results are shown in Table 2.

$$\frac{\tau_s}{\tau_{real}} = \frac{v_{real}}{v_s} \tag{10}$$

The minimum values of τ_{real} are chosen for the numbers 1 to 13 and the minimum values of τ_s are chosen for the numbers 14 to 17 for comparison with τ_0 of different numbers. The result is shown in Figure 7.

The trend of two polylines in Figure 7 is the same. Define the relative error as

$$\bar{e} = \frac{1}{n} \sum_{i=1}^n \frac{\tau_i - \tau_0^i}{\tau_0^i} \tag{11}$$

where n is the amount of taxiing routes, τ_i is the operation time of real airport operation for i -th route in Figure 7 and τ_0^i is the operation time of Petri net operation for i -th route in Figure 7. The relative error

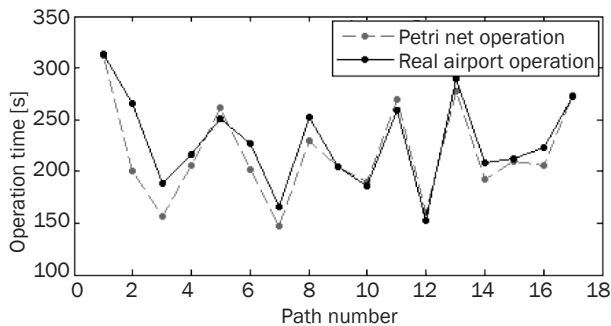


Figure 7 – Comparison of aircraft taxiing time

Table 2 – Comparison of aircraft taxiing time and velocity

No.	Petri net running time τ_0 [s]	Real flight operation time τ_{real} [s]	Surface state	Real flight velocity τ_{real} [kts]	Real flight operation time with velocity of 30 kts τ_s [s]
14	192	499	1	19.6	326
		330	1	19.0	209
		285	1	23.8	226
15	210	685	0	/	/
		572	0	/	/
16	206	475	1	13.4	212
		518	1	18.6	320
17	273	511	1	13.1	223
		639	0	/	/
		525	1	15.5	272

between two polylines in Figure 7 is calculated as $\bar{e} = 5.9485\%$. The error mainly results from two aspects: (1) Aircraft taxiing on the surface is affected by the weather, the conditions of the pavement and some other factors; (2) The taxiing of aircraft is simply considered as the uniform movement and uniform acceleration movement while it is more complicated in the actual situation. Therefore, the relative error is necessary and reasonable.

5.2 Dynamic verification

Dynamic verification means the process that all the aircraft on the surface taxi from their initial places to the target places under the airport surface operation constraints and taking the interplay between all aircraft on the surface into consideration. It mainly considers the interplay between aircraft on the surface. This paper uses 30 minutes as a time window and compares the real total taxiing time of all aircraft on the surface and the Petri net runtime with the same aircraft number and the same start or arrival parking as the real situation within the time window.

Toulouse airport is still taken as an example and flight data of four time windows are selected for the verification. For the real flight data in each time window, the corresponding model data are given by running Petri net 40 times. The results are shown in Figure 8 where the horizontal axis represents the running times and the vertical axis represents the total taxiing time. From the figure, the real total taxiing time is obviously higher than the average of the model running time as the model is an ideal state compared to the real situation and the actual factors as described in the model constraints part cannot be fully considered.

The mean relative error of the total taxiing time for each time window is calculated and the results are shown in Table 3. The mean value for these four time windows is $\bar{e} = -13.533\%$, which is a little larger than the static error. This occurs because the static error is included in the dynamic error and there is a case

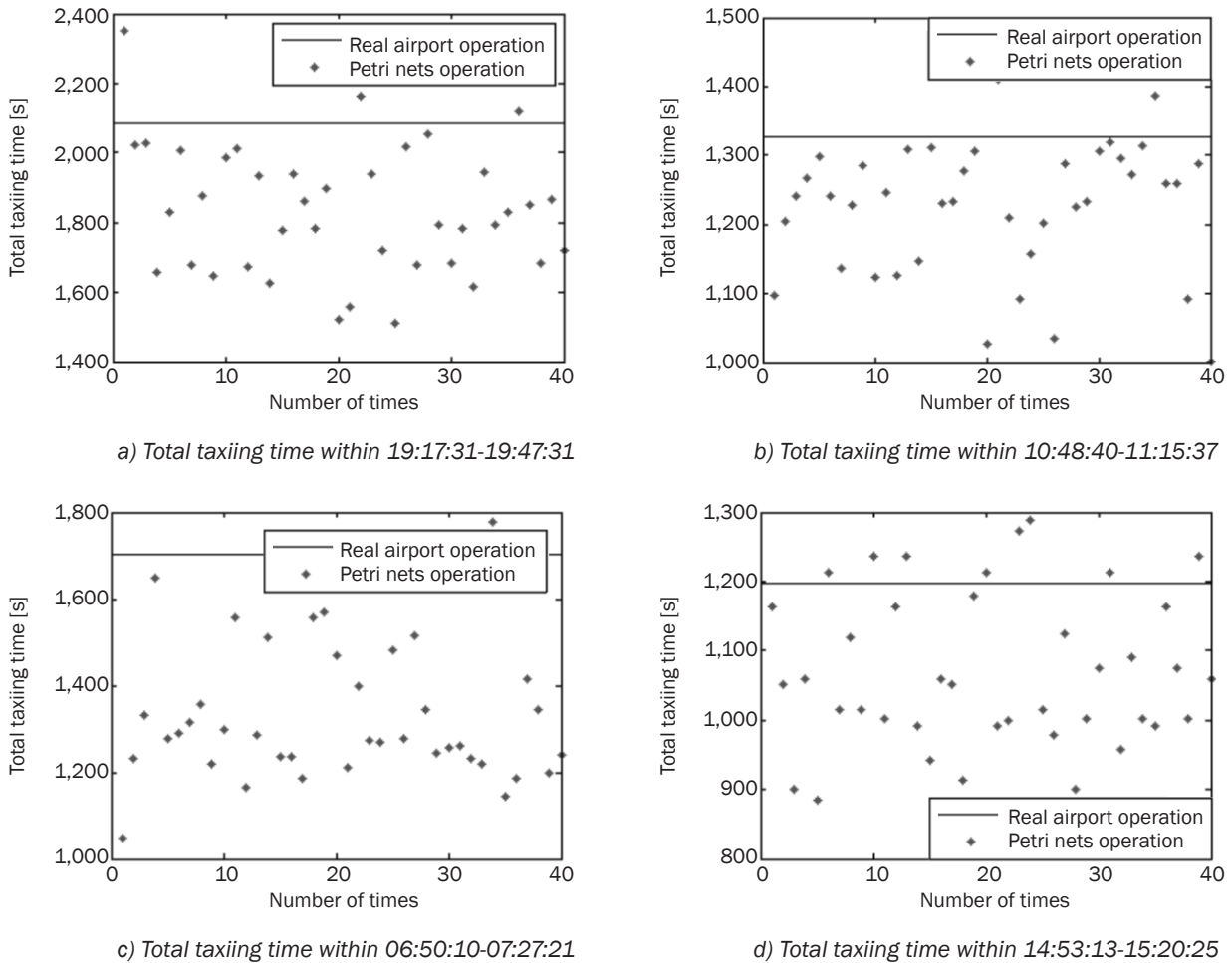


Figure 8 – Comparison of total taxiing time for several aircraft in different time periods

Table 3 – Mean relative error of total taxiing time in different time windows

Time window	19:17:31-19:47:31	10:48:40-11:15:37	06:50:10-07:27:21	14:53:13-15:20:25
Number of aircraft	8	5	6	4
Actual total taxiing time [s]	2,084	1,328	1,701	1,196
Mean relative error of total taxiing time	-11.91%	-7.79%	-22.02%	-10.43%
Maximum time saving percentage	27.5%	23.9%	38.3%	26.0%

of error accumulation because of multiple aircraft on the surface. Considering the static error for each aircraft, the dynamic error is much smaller than the sum of static error for all aircraft on the surface within the corresponding time window.

Besides, the routes of aircraft run by Petri net model might be different each time. Therefore, the discrepancies between actual time and Petri net model time are definite. However, this reminds us that the actual taxiing route might not be optimal. In respecting the taxiing rules modelled by Petri net, the model provides several solutions for conflict-free taxiing. If the optimal solution among these 40 samples which means it cost the minimal taxiing time is selected for taxiing, the time saving could be calculated and the results are shown

in Table 3. The time saving percentage could be 28.5% as an average value, which proves that the airport surface Petri net model built in this paper could provide a more optimal solution for the taxi route planning.

6. CONCLUSION

In this paper, the timed and coloured Petri net is used to model the airport surface operation and the simulation is carried out by CPN Tools. The airport layout is clearly represented by Petri net model which is availably simplified by the division of parking places. The directional movement of aircraft on the surface can be effectively simulated with the constraints for type of places by Petri net. Based on the general

taxiing time estimation for each taxiway and the time stamp concept of Petri net, the taxiing time costs can be calculated along aircraft movement and the airport situation can be seen from the Petri net model at any moment. The method of defining the capacity of the operating unit place is presented and it allows for the shortest longitudinal separation between aircraft to ensure movement safety. Petri net's inherent elements are used to incorporate the operation rule that the aircraft need to obey when taxiing, and finally the simulation of conflict-free taxiing of aircraft is achieved.

The simulation model is evaluated by comparison of simulation results with the field data obtained at Toulouse Blagnac Airport. It proves that the model could not only represent the layout of the airport but also simulate the free-conflict taxiing of aircraft on the airport surface and provide a better solution for the aircraft ground movement. These findings will be useful to achieve more efficient airport operation.

ACKNOWLEDGMENTS

This research has been supported by the Fundamental Research Funds for the Central Universities (Grant No. 3122017111).

苏志刚, 教授
电子邮箱: srsu@vip.sina.com
邱梦琦, 硕士
电子邮箱: mqiu_77@sina.com
中国民航大学中欧航空工程师学院
中国天津市东丽区津北公路2898号

基于赋时着色PETRI网的机场场面运行建模与仿真

摘要

为了满足国际民航组织(ICA0)对先进的场面引导与控制系统(A-SMGCS)中机场航空器路由规划的需求,给出了基于赋时着色Petri网的机场场面运行建模与仿真方法。根据机场场面结构布局和场面运行单元特点,建立了机场场面静态模型。在此基础上,根据ICA0 DOC 9830文件中对航空器场面滑行速度的要求,对静态模型赋予时间属性,建立了机场场面运行动态模型。提出了场面运行单元库所容量的定义方法和利用Petri网元素进行场面管制规则约束的方法。与本领域的其他文章不同,本文所建立的赋时着色Petri网模型可以在不依赖模型外其他算法的情况下,利用Petri网模拟器进行场面航空器无冲突滑行模拟。基于CPN Tools软件,以图卢兹机场场面运行过程为例,验证了模型的有效性。

关键字:

Petri网; 机场场面模型; 无冲突滑行; 滑行路径规划

REFERENCES

- [1] *The Manual - Airport CMD Implementation*. Brussels: Eurocontrol; 2017. Available from: <https://www.eurocontrol.int/sites/default/files/publication/files/airport-cdm-manual-2017.PDF>
- [2] Tang M, Tang X, Zhou L, et al. Research on Airport Air-side Operation Simulation Based on Multi-Agent. *Aeronautical Computing Technique*. 2015;45(4): 51-56.
- [3] Nakamura S, Furuta K, Kanno T. Multi-agent simulation of ground aircraft operations at a large airport. In: *International ICST Conference on Simulation TOOLS and Techniques*; 2010.
- [4] Ribeiro VF, Li W, Milea V, et al. Collaborative Decision Making in Departure Sequencing with an Adapted Rubinstein Protocol. *IEEE Transactions on Systems Man & Cybernetics Systems*. 2016;46(2): 248-259.
- [5] Ravizza S, Atkin JAD, Burke EK. A more realistic approach for airport ground movement optimisation with stand holding. *Journal of Scheduling*. 2014;17(5): 507-520.
- [6] *Simmod Manual - How Simmod Works*. US: Federal Aviation Administration. Available from: http://www.tc.faa.gov/acb300/how_simmod_works.pdf [Accessed 27 October 2018].
- [7] Wu Z. *Introduction to Petri net*. Beijing, China: China Machine Press; 2006.
- [8] Yuan C. *Petri Net Principle*. Beijing, China: Publishing House of Electronics Industry; 1998.
- [9] Tang X, Zhu X. *Petri net theory and its application in civil aviation transportation engineering*. Beijing, China: China Civil Aviation Publishing House; 2014.
- [10] Werther B. Colored Petri Net Based Modeling of Airport Control Processes. In: *Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, 28 Nov - 1 Dec 2006, Sydney, Australia*. IEEE; 2006. p.108-108.
- [11] Davidrajuh R, Lin B. Exploring airport traffic capability using Petri net based model. *Expert Systems with Applications*. 2011;38(9): 10923-10931.
- [12] Skorupski J. Airport traffic simulation using Petri nets. *Communications in Computer & Information Science*. 2013;395: 468-475.
- [13] Kovacs A, Nemeth E, Hangos KM. Modeling and optimization of runway traffic flow using coloured Petri nets. In: *International Conference on Control and Automation, 26-29 June 2005, Budapest, Hungary*. Vol. 2. IEEE; 2005. p. 881-886.
- [14] Shortle JF, Xie R, Chen C, et al. Simulating collision probabilities of landing airplanes at Nontowered airports. *Simulation Journal of the Society for Computer Simulation*. 2004;80(1): 21-31.
- [15] Dezani H, Gomes L, Damiani F, et al. Controlling traffic jams on urban roads modeled in Coloured Petri net using Genetic Algorithm. In: *IECON 2012 - Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society, 25-28 Oct. 2012, Piscataway, N.J., USA*. IEEE; 2012. p. 3043-3048.
- [16] Dezani H, Marranghello N, Damiani F. Genetic algorithm-based traffic lights timing optimization and routes definition using Petri net model of urban traffic flow. *IFAC Proceedings Volumes*. 2014;47(3): 11326-11331.
- [17] Dezani H, Bassi RDS, Marranghello N, et al. Optimizing urban traffic flow using Genetic Algorithm with Petri net analysis as fitness function. *Neurocomputing*. 2014;124(2): 162-167.

- [18] Huang S, Sun T, Lv B. Petri net simulation arithmetic of the shortest directional path in transportation net. *Journal of Nanjing University of Aeronautics & Astronautics*. 2002;34(2): 121-125.
- [19] Wang C, Tang X, An H. Research on the optimization of aircraft dynamic routes planning for A-SMGCS. *Journal of Wuhan University of Technology (Transportation Science & Engineering)*. 2012;36(5): 1069-1073.
- [20] Tang X, Wang Y, Han S. Aircraft dynamic taxiway routes planning for A-SMGCS based on DEDS. *Systems Engineering and Electronics*. 2010;32(12): 2669-2675.
- [21] Zhu X, Tang X, Han S. Conflict prediction and avoidance control for A-SMGCS taxiway. *Journal of Nanjing University of Aeronautics & Astronautics*. 2011;43(4): 504-510.
- [22] Zhu X, Tang X, Han S. Avoidance strategy for head-on conflict on taxiway based on supervisory control theory of DES. *Journal of Southwest Jiaotong University*. 2011;46(4): 664-670.
- [23] Zhu X, Tang X, Han S. Aircraft initial taxiing route planning based on Petri net and Genetic Algorithm. *Journal of Southwest Jiaotong University*. 2013;48(3): 565-573.
- [24] Jensen K, Kristensen LM, Wells L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*. 2007;9(3-4): 213-254.
- [25] ICAO Doc9830. *Advanced Surface Movement Guidance and Control Systems (A-SMGCS) Manual*. Montreal: ICAO; 2004.