Multi-layer Point Merge System for Dynamically Controlling Arrivals on Parallel Runways

Man Liang, Daniel Delahaye, Mohammed Sbihi, Ji Ma

Multi-layer Point Merge System for Dynamically Controlling Arrivals on Parallel Runways

Man LIANG, Daniel DELAHAYE, Mohammed SBIHI, Ji MA
Laboratory in Applied Mathematics, Computer Science and Automatics for Air Transport
Ecole Nationale de l'Aviation Civile
Toulouse, France
Email: man.liang@enac.fr, daniel@recherche.enac.fr, mohammed.sbihi@enac.fr, ji.ma@recherche.eanc.fr

***Abstract*—In order to efficiently and robustly land more aircraft on two parallel runways, search the conflict-free and less-delay reroute for arrival aircraft, we plan to control arrival aircraft with a novel trajectory operation model named Multi-layer Point Merge System (ML-PM). In this paper, firstly the notional example of ML-PM system is introduced, the characteristics of its horizontal and vertical profiles are described. Secondly, according to the availabilities on two parallel runways, the positions of aircraft on sequencing legs, a trajectory reroute model is built. After that, the geometric relations for conflict detection in merging zones are analyzed in details. Furthermore, two tests with runway re-assignment on and off are compared to study the performances of proposed ML-PM system, the numerical results have shown that: 1) both tests could generate conflict-free trajectories, 2) average delay, average landing interval, and makes-pan could be reduced by runway re-assignment. The relative improvement for average delay is 36.36%, for make-span and average landing interval is 1.35% and 1.36% respectively. However, the average flight time, flight distance, fuel consumption and co_2 emission will increase due to runway re-assignment, the relative increases are 13.49%, 1.11%, 13.49% and 13.49%, respectively. After that, a detailed study about the runway assignment is done, the changes of different indicators are analyzed. Finally, the conclusion is made, ML-PM system shows a very positive ability to dynamically control arrival flows to land on parallel runways.**

***Keywords*—Arrival sequencing and merging, multi-layer point merge, runway assignment**

I. Introduction

Terminal Control Area (TMA) is a controlled airspace surrounding one or several airports where there is a high volume of traffic climbing out or descending into airports. Due to its operational complexity, optimization of traffic in dense TMA is identified as one of the most challenging problems in air transportation.

The future air traffic management system will include higher level of automation in the separation assurance. Current researches, such as NextGen project in US and SESAR project in Europe, have developed some decision support tools to help the controllers to handle the routine work, to facilitate the arrival flows into congested airports.

Under the concept of Trajectory Based Operation (TBO), in order to further improve the arrival management at congested airport with parallel runways, a novel trajectory operation model named Multi-layer Point Merge (ML-PM) is proposed recently in [1], [2]. The key point of this concept is to efficiently and robustly land more aircraft by segregating Heavy/Medium/Light aircraft to different flight layers on the sequencing legs of Point Merge (PM). This idea combines the benefits of PM and the automated separation assurance techniques, aims to make the arrival trajectory at congested airport more efficient, orderly, and safe. However, previous work in [1] did not consider the reassignment of runway in its optimization model, therefore a deep study on the integration of ML-PM based autonomous arrival management system with balancing the landing rate of two runways will be necessary and interesting, which will enforce the previous study works. Thus, in this paper, an advanced system with runway reassignment in consideration for dynamically control arrival aircraft to land on parallel runways is studied.

The structure of this paper is organized as below: following a short introduction, the concept of ML-PM based trajectories management system is presented in Section II, a notional example and a trajectory reroute system are described. After that, in Section III the conflict-free tactical reroute model is proposed, the problem formulation, the geometric relations for conflict detection and resolution in merging zone are analyzed in details. In Section IV, several scenarios are studied and the numerical results are discussed. Finally, conclusion and remarks are made.

II. ML-PM Based Dynamical Trajectories Management Concept Overview

A. Parallel Instrument Runway Operation

According to ICAO DOC.9643 Manual on simultaneous operations on parallel or near-parallel instrument runways, there are four modes of operation concepts relating to simultaneous operations on parallel instrument runways:

1) Simultaneous approaches to parallel runways with independent parallel instrument approaches.
2) Simultaneous approaches to parallel runways with dependent parallel instrument approaches.
3) Independent instrument departures from parallel runways.
4) Segregated operations on parallel runways.

In Mode 1), 2) and 4), there may be semi-mixed operations, i.g. one runway is used exclusively for departures (or

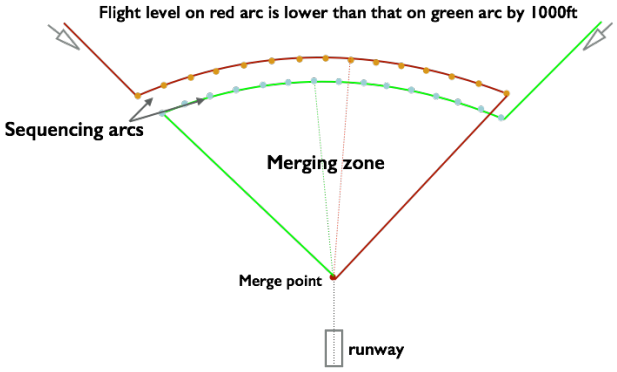


Fig. 1: Basic PM topology for single runway

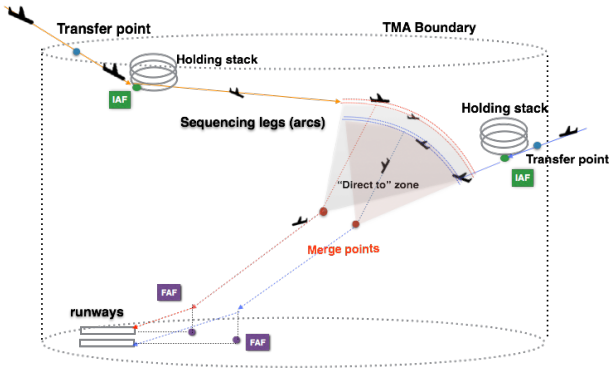


Fig. 2: ML-PM system for parallel runways operation

approach) while the other runway is used for a mixture of approaches and departures, or there may be mixed operations, all modes of operation are possible in each runway. In our research, to simplify the problem, we consider the Mode 1), which means that the dependent separation between two aircraft landing on adjacent runways will not be considered. Some safety related requirements and operation procedures relating to independent parallel instrument approach should be considered in our study. For example, each pair of parallel approaches should have a "high side" and a "low side" to provide vertical separation until aircraft are established inbound on their respective parallel ILS localizer courses, the high-side altitude should be 1000 ft above the low side at least until 10 NM from the threshold.

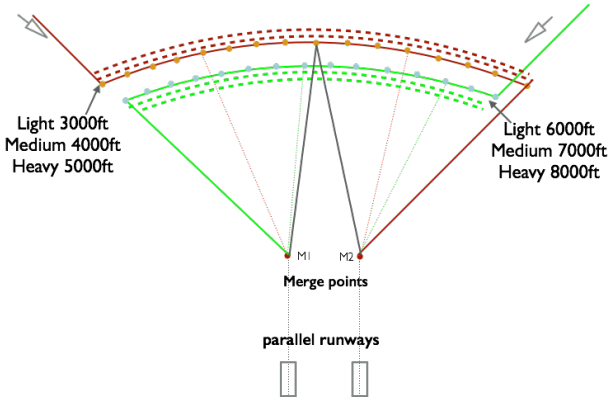


Fig. 3: Horizontal plan of ML-PM system

B. ML-PM System Design

PM is a systematized method for sequencing arrival flows developed by the EUROCONTROL Experimental Centre in 2006. As showed in Fig. 1, a classic topology of PM system in horizontal plan consists of a point, named the merge point, and the pre-defined legs, named the sequencing legs which are equidistant from this point. PM is now one of the ICAO Aviation System Block Upgrades and is referenced as a technique to support continuous descent operations (Doc 9931) [3]. Performance of PM was analyzed both by real time simulations and real operation environments. The results show that less workload for controller and more predictable and efficient trajectory for flight will be gained [4], [5].

Based on the basic PM system, an example of ML-PM system for independent operations on two parallel runways is developed. As showed in Fig.2 and Fig.3, in this novel system, aircraft from different directions arrive to the sequencing legs, they remain in lateral mode. Horizontally, inner and outer sequencing legs have a common center of circle. Vertically, different segregated flight layers on the sequencing legs could be provided for coming aircraft according to their wake turbulence categories. "Heavy" aircraft will choose the higher level, "Medium" aircraft will use the middle level and "Light" aircraft will enter the lower level, all of the three layers have a same projection on horizontal level. After aircraft enter the sequencing leg, they fly at a constant and pre-defined speed, and when at some moment where there is no conflict or clear of weather, they will perform a "Direct to" turn towards the merge point, during the merging process, they perform a continuous descent. The airborne separation between aircraft is maintained automatically by a conflict detection and resolution algorithm.

The advantage of this ML-PM system is that arrival aircraft can easily change the runway to land due to some reasons, such as unavailability on initial landing runway, weather, UAV, military control, etc., the challenges are: first, the distances from one point on sequencing legs to different merge points are not equidistant, second, there maybe some intersections between trajectories in the merging zone due to the change of landing runway. Therefore, the conflict detection and resolution in this ML-PM system with mixed operation on parallel runways is more complex than the ML-PM system with separated operation designed in reference [1].

C. Tactical Reroute System Design

Terminal air traffic control system is the most complex and dynamic system in ATM. In order to well adapt its system requirements, we choose to solve this 24 hours' optimization problem by application of Receding Horizontal Control (RHC) technique. As illustrated in Fig.4, the overall time horizon of 24 hours is firstly divided into several smaller time horizons called sliding time windows, then at each time step we

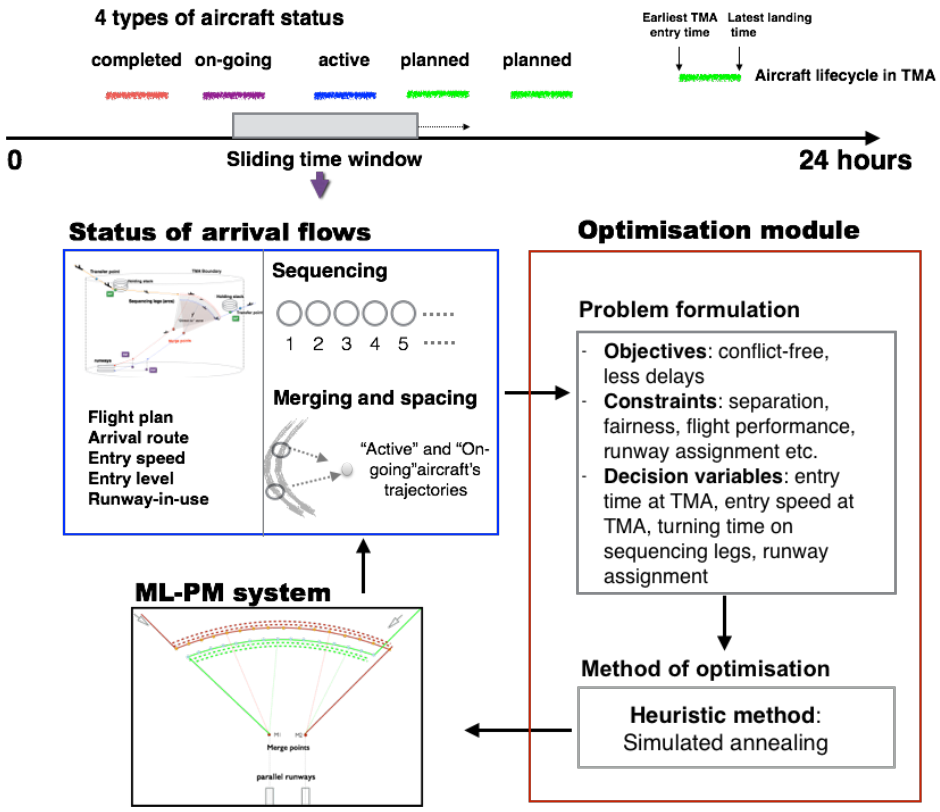


Fig. 4: Architecture of tactical reroute system

determine a plan of action in this time window and apply it.

According to the relative relationship between the aircraft life cycle and the active time window, we classify the status of aircraft into 4 types: Completed, On-going, Active and Planned. *Completed* means that the aircraft already landed. *On-going* means aircraft is already in TMA, but has not landed yet. Some part of their trajectory is fixed, while some part of their trajectory could be changeable. *Active* means the whole trajectories of aircraft could be changed. *Planned* is the rest part of aircraft who do not belong to any types mentioned before. Those aircraft with *Active* and *On-going* status will be selected to enter the optimization module inside the current time window.

In the current window, the architecture of tactical reroute system consists of several parts: status of arrival flows, ML-PM system, and optimization module. The optimization module is described in two parts: problem formulation and optimization method. In the part of problem formulation, decision variables include: change the entry speed of aircraft, change the entry time of aircraft, control the turning time on the sequencing legs, and assign the landing runway for each aircraft. Objective of our optimization problem is to generate the conflict-free and less-delay trajectories for all arrival aircraft. Constraints include separation, runway assignment, fairness and flight performance. Runway assignment is to decide which runway aircraft are planned to land. Fairness and flight performance is to ensure an operation-acceptable landing sequence. In the part of optimization method, it is applied by a kind of heuristic algorithms named Simulated Annealing algorithm, more detailed information is in reference [1], here we keep some of its parts.

III. Trajectory-based Tactical Reroute Mathematical Model

A. Problem Formulation

Based on the South-landing operation at Beijing Capital International Airport (BCIA), a route network is designed, see Fig. 5. Flights Enter from West, waypoints KM, JB and BOBAK, follow a set of routes denoted by $G_W = (V_W, E_W)$, flights enter from East, waypoints GITUM, DOGAR, and VYK, follow another set of routes denoted by $G_E = (V_E, E_E)$. We denote $G = G_W \cup G_E = G(V, E)$. V is the set of way-points w_i ($i \in N$), E is the set of arcs u_i ($i \in N$) connecting two way-points by a straight line or convex arc.

Assume that there is a set of aircraft $F = \{1, 2, ..., n\}$, and for each aircraft $i \in F$, the following data are also given:

- e_i Entry point in TMA,
- t_i^e Estimated Time of Arrival (ETA) for aircraft i at the entry point,
- v_i^e Initial speed of aircraft i at the entry point,

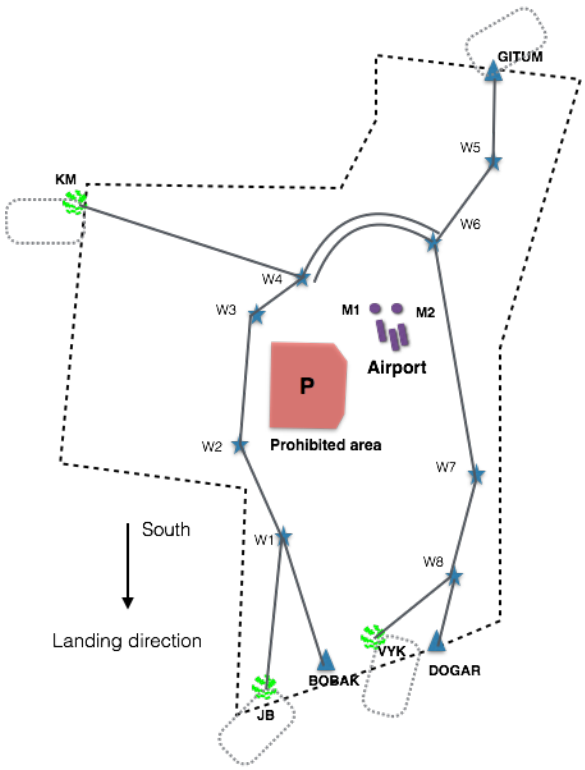


Fig. 5: Route network for mathematical model

- r_i^e Initial landing runway for aircraft i, for aircraft coming from KM, JB and BOBAK, $r_i^e = 1$, for aircraft coming from VYK, DOGAR and GITUM, $r_i^e = 0$,
- ETA_i^L Estimated Time of Landing for aircraft i,
- cat_i Wake turbulence category of aircraft i.

1) Variables: decision variables in this optimization problem include t_i, v_i, t_i^T, r_i. Here, for each aircraft i,

- t_i is the actual entry time at TMA,
- v_i is the actual entry speed at TMA,
- t_i^T is the actual turning time on the sequencing legs,
- r_i is the actual landing runway.

From an operational point of view, all these variables are adjusted in a discrete way to adapt the real world. We adjust t_i by a number of slots denoted j, typically each slot $\Delta t = 5s$,

$$t_i = t_i^e + j\Delta t, j \in \mathbf{Z} \tag{1}$$

we adjust v_i by a discrete way,

$$v_i = v_i^e(1+g), g = 0, \pm 1\%, \pm 2\%, ..., \pm n\% \tag{2}$$

similarly, we adjust t_i^T as formula below,

$$t_i^T = t_{imin}^T + h(t_{imax}^T - t_{imin}^T), h = 0, 1\%, 2\%, ..., 100\% \tag{3}$$

where, t_{imin}^T is the earliest turning time for aircraft i, and t_{imax}^T is the latest turning time, note that t_i^T depends on the allowable turning arc u_i in the G. Finally, for the runway reassignment, r_i is defined as below:

$$r_i = \begin{cases} 1 & \text{if } i \text{ expected to merge at } M_1 \\ 0 & \text{if } i \text{ expected to merge at } M_2 \end{cases} \tag{4}$$

here, M1 connects to runway 18R-36L, M2 connects to runway 01-19.

2) Constraints: some operation constraints should be considered in this optimization problem , they are vital to fairness and safety of aircraft.

Firstly, t_i could vary in a reasonable range. If aircraft arrive too early before t_i^e , they need to fly at a higher speed before entry TMA, which induce a high fuel consumption, if aircraft arrive too late after t_i^e, they will produce propagation of delay for the destination airport and passages etc. [6]–[8]. In this paper, the earliest time of arrival is limited to 3 minutes before ETA, the latest arrival time is 10 minutes after ETA, thus t_i should comply with the formula below:

$$t_i^e - 3\text{min} \leq t_i \leq t_i^e + 10\text{min} \tag{5}$$

Secondly, speed change is limited by the performance of commercial aircraft in descent profile. In TMA aircraft typically fly below 10000ft, their airspeed can not be bigger than 250 kts due to bird ingestion damage, and at the same time it cannot be less than the minimum clean configuration speed due to low speed stall, thus v_i should be subject to:

$$v_i^e(1-15\%) \leq v_i \leq v_i^e(1+15\%) \tag{6}$$

$$v_i^e(1-15\%) \geq \begin{cases} 230\text{kts for Heavy aircraft} \\ 220\text{kts for Medium aircraft} \end{cases} \tag{7}$$

Thirdly, t_i^T should comply with:

$$t_{imin}^T \leq t_i^T \leq t_{imax}^T \tag{8}$$

Last but not least, radar separation and wake turbulence separation minimum should be considered. ICAO-regulated Wake turbulence minimum is a distance-based separation under radar control environment, together with approach radar separation between two successive aircraft with the same category of wake turbulence, in total the required aircraft minimum separation in TMA, denoted by $s_{i,j}^{min}$ is listed in Table. I. In order to adapt to a time-based metering system, which is convenient for conflict detection and metering the flows, we assume a reference velocity for each category of aircraft, then this distance-based separation could be transferred to a time-based separation. Referring to the performance of commercial aircraft with Medium and Heavy category, taking into account the speed on final approach of successive arrivals, the distance-based minimum is converted to the minimum time-based wake turbulence separation at FAF, showed in Tab.II, references could be found in [9] as well. Based on this time-based separation minimum, we will discuss how to comply with these constraints in the next part: conflict detection.

3) Objective: we group all decision variables for a flight i as a vector named $\vec{y_i}$, and $\vec{y_i} = (t_i, v_i, t_i^T, r_i)$, then the objective function is defined as below:

$$z = Min \quad \sum_{i=1}^{n} C_i(\vec{y_i}) + \frac{\alpha}{n}\sum_{i=1}^{n}(t_i^L - ETA_i^L)^2 \tag{9}$$

here, n is the number of flights, t_i^L is the actual landing time of flight i on the runway, it depends on t_i and t_i^T. C_i is

TABLE I: Distance-based Aircraft Minimum Separation (unit: Nm)

Preceeding \ Trailing	Heavy	Medium	Light
Heavy	4	5	6
Medium	3	3	5
Light	3	3	3

TABLE II: Time-based Equivalent Minimum Separation (unit:Second)

Preceeding \ Trailing	Heavy	Medium	Light
Heavy	82	118	150
Medium	60	64	94
Light	60	64	68

the conflict indicator, it depends on $\vec{y_i}$, and α is a weighting parameter.

Minimizing such an objective function enables to eliminate the conflicts and reduce the delay. The value of α in this paper is 0.001, which gives a priority to conflict resolution for safety reason.

B. Conflict detection

In order to calculate C_i, we have to make conflict detection between each pair of aircraft. According to ICAO regulations, two aircraft are considered to be in a conflict if their horizontal separation is less than the minimum aircraft separation in Tab. II or if their vertical separation is less than 1000 ft.

In ML-PM system, vertical separation is partially assured by vertical profile design, specially on the entry points of sequencing legs, Final Approach Fixes (FAF), etc.. The designed transferring altitudes are listed in Tab. III. These altitudes are designed in consideration of executing a continue descent in BCIA TMA.

TABLE III: Altitude at Significant Waypoints

Name of way-points	Designed altitudes (units:meters)
KM	4500
JB	4200
BOBAK	4200
VYK	5100
DOGARD	4200
GITUM	3600
W4	2400 for Medium, 2700 for Heavy
W6	2900 for Medium 3000 for Heavy
M1	1500
M2	1200

If aircraft could not be assured by vertical separation, then a horizontal separation must be assured. In the ML-PM system, in order to well detect horizontal conflicts, we separate them into three types: *link*, *node*, and *merge*. A *link* conflict refers to the catch-up conflict and overtake conflict between two successive aircraft flying on the same arc, a *node* conflict refers to the conflict between two aircraft on different arcs converging to a common way-point, a *merge* conflict refers to the conflict between aircraft approaching to M1 or M2.

Link conflict is detected by verification of the time difference at the beginning and the end of arc. Assuming that proceeding aircraft i and trailing aircraft j will entry into the same arc $u \in E$ defined by way-points w_a and w_b, we define $\Delta t_{i,j}^{\overline{u}}$ as the entry time difference and $\Delta t_{i,j}^{\underline{u}}$ as the exit time difference, they should be subject to the constraints:

$$\Delta t_{i,j}^{\overline{u}} := t_j^{w_a} - t_i^{w_a} \geq s_{i,j}^{min} \quad | \, j, i \in F \tag{10}$$

$$\Delta t_{i,j}^{\underline{u}} := t_j^{w_b} - t_i^{w_b} \geq s_{i,j}^{min} \quad | \, j, i \in F \tag{11}$$

$$\Delta t_{i,j}^{\overline{u}} \times \Delta t_{i,j}^{\underline{u}} \geq 0 \tag{12}$$

Node conflict is detected by verification of the time difference on passing the common way-point. Assuming that proceeding aircraft i and following aircraft j will pass a common way point $w_c \in V$, then aircraft safety must be assured by the following constraints:

$$\Delta t_{i,j}^{w_c} := t_j^{w_c} - t_i^{w_c} \geq s_{i,j}^{min} \quad | \, j, i \in F \tag{13}$$

Merge conflicts are more complex, because the distances between each point on the sequencing legs and the M1 or M2 are different. Taking into account of the air traffic control reality, as well as the complexity of trajectories in contingency, we build a method for simplifying this merge conflict detection problem. First, a merging procedure is pre-defined. As shown in Fig. 6, two merging zones are designed, named zone 1 and zone 2. Aircraft from West coming from way-points KM, JB, and BOBAK, if it wants to change its initial landing runway to runway 01-19, it has to pass the midpoint mp_2 firstly, then it can take the action "Turn" in the zone 2; conversely, aircraft from East coming from way-points GITUM, DOGAR, and VYK, if it changes to land on runway 18R-36L, it has to pass the midpoint mp_1 firstly, then it can take the action "Turn" in the zone 1. Second, in each merging zone, we transfer the geographic route network of this system into a virtual time-based network to detect the merge conflicts. All the links are time segments in Fig. 6, so $b + \vec{d_s}$ and c are not fixed, because t_i^T is different. The node "Turning point" is to assure the separation between two successive aircraft leaving the sequencing legs, making sure that a lateral separation is always kept between each pair of aircraft. Finally, to ensure the safety in merge zones, on the turning point, t_i^T and t_j^T comply with the constraints:

$$\Delta t_{i,j}^T := t_j^T - t_i^T \geq s_{i,j}^{min}(r_i \times r_j + (1 - r_i)(1 - r_j)) \tag{14}$$

then, from turning point to merge point, we assume that there are two aircraft i and j arriving on different sequencing legs, i flies on outer sequencing leg, j flies on inner sequencing leg.

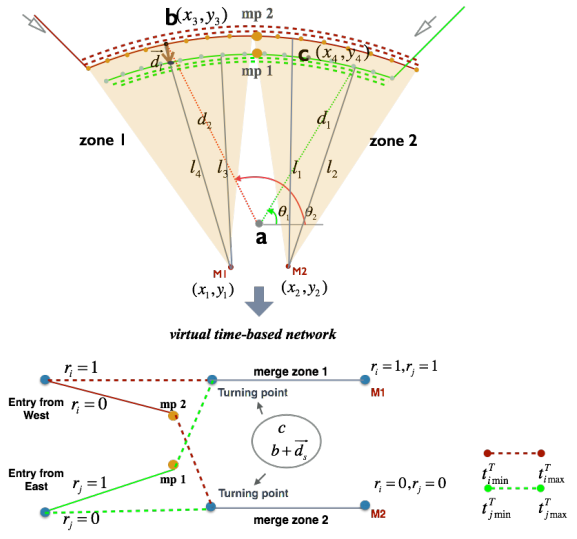


Fig. 6: Conflict detection in merging zone

Given the coordinate (x_a, y_a) of centre point a, the coordinates $(x_3, y_3), (x_4, y_4)$ could be calculated by:

$$\theta_2 = -\frac{v_i(t_i^T - t_{imin}^T)}{d_2} + \theta_{imin}(r_i) \quad (15)$$

$$\theta_1 = \frac{v_j(t_j^T - t_{jmin}^T)}{d_1} + \theta_{jmin}(r_j) \quad (16)$$

$$x_3 = d_2 \times \cos\theta_2 + x_a \quad (17)$$

$$y_3 = d_2 \times \sin\theta_2 + y_a \quad (18)$$

$$x_4 = d_1 \times \cos\theta_1 + x_a \quad (19)$$

$$y_4 = d_1 \times \sin\theta_1 + y_a \quad (20)$$

where, θ_{imin} corresponds to t_{imin}^T, and θ_{jmin} corresponds to t_{jmin}^T. Then, θ_{imin} and θ_{jmin} depend on r_i and r_j respectively, if aircraft do not need to change the runway, then its earliest turning time is the entry of sequencing leg, its latest turning time is the midpoint of sequencing leg; if it needs to change the runway, then its earliest turning time is the midpoint of it's sequencing leg, its latest turning time is the end of sequencing leg.

Consequently, l_1, l_2, l_3, l_4 could be calculated based on $(x_3, y_3), (x_4, y_4)$, the conflict detection between turning point to merge point in each merging zone could be applied by *Link* conflict and *Node* conflict methods.

IV. Scenarios and Results

A. Input data

1) Flight data: due to lack of real radar data, we prepare the virtual input data for the scenario simulation. As shown in Tab. IV, there are 418 flights planned to land at BCIA in 18 hours, of which 77.99% are Medium, 22.01% are Heavy. 13.40% traffic come from KM, 12.20% from JB, 11.96% from BOBAK, 20.33% from VYK, 19.86% from DOGAR, and 22.25% from GITUM. In total, 62.44% flights will land on runway 01-19, 37.56% flights will land on runway 18L-36R. The distribution of flights is based on the real operation situations at BCIA.

2) Routes data: there are 26 nodes (waypoints), 24 arcs (or links), and 12 possible routes to present the route network of BCIA in Fig.5.

$V = \{w_1, w_2, w_3, w_4, ..., w_{26}\}$
$E = \{u_1, u_2, u_3, ..., u_{24}\}$
$R_1 = \{u_1, u_{13}, u_{15}, u_{19}, u_{20}\}$
$R_2 = \{u_2, u_3, u_4, u_5, u_{13}, u_{15}, u_{19}, u_{20}\}$
$R_3 = \{u_6, u_3, u_4, u_5, u_{13}, u_{15}, u_{19}, u_{20}\}$
$R_4 = \{u_7, u_8, u_9, u_{22}, u_{24}, u_{15}, u_{19}, u_{20}\}$
$R_5 = \{u_{10}, u_8, u_9, u_{22}, u_{24}, u_{15}, u_{19}, u_{20}\}$
$R_6 = \{u_{11}, u_{12}, u_{22}, u_{24}, u_{15}, u_{19}, u_{20}\}$
$R_7 = \{u_1, u_{21}, u_{23}, u_{16}, u_{17}, u_{18}\}$
$R_8 = \{u_2, u_3, u_4, u_5, u_{21}, u_{23}, u_{16}, u_{17}, u_{18}\}$
$R_9 = \{u_6, u_3, u_4, u_5, u_{21}, u_{23}, u_{16}, u_{17}, u_{18}\}$
$R_{10} = \{u_7, u_8, u_9, u_{14}, u_{16}, u_{17}, u_{18}\}$
$R_{11} = \{u_{10}, u_8, u_9, u_{14}, u_{16}, u_{17}, u_{18}\}$
$R_{12} = \{u_{11}, u_{12}, u_{14}, u_{16}, u_{17}, u_{18}\}$

Here, $R_1, ..., R_6$ lead the aircraft to land on runway 18L-36R, $R_7, .., R_{12}$ lead the aircraft to land on runway 01-19. Aircraft from u_1, u_2, and u_6 will initially land on runway 18L-36R, with $r_i^e = 1$, aircraft from u_7, u_{10} and u_{11} will initially land on runway 01-19, with $r_i^e = 0$.

Nomenclature

a	Centre of the sequencing legs	b	A point on the outer sequencing leg
c	A point on the inner sequencing leg	M1	Merge point for runway left
M2	Merge point for runway right	d_1	Radial of inner sequencing leg
d_2	Radius of outer sequencing leg	l_1	Distance between b and M2
l_2	Distance between c and M2	l_3	Distance between c and M1
l_4	Distance between b and M1	θ_1	Turning angle for a/c on inner sequencing leg
θ_2	Turning angle for a/c on outer sequencing leg	mp_2	midpoint of the outer sequencing legs
mp_1	midpoint of the inner sequencing legs	$\vec{d_s}$	vector from b to one point of inner sequencing leg and its direction is toward the merge point

TABLE IV: Flight Data

No.route	Entry point	Total flights	H. A/C	M. A/C
1	KM	56	21	35
2	JB	51	14	37
3	BOBAK	50	15	35
4	VYK	85	12	73
5	DOGAR	83	10	73
6	GITUM	93	20	73

3) Speed design: in order to simplify the study problem, the speed change profile is pre-defined. As shown in Fig. 7, aircraft maintain the entry speed from the entry point of TMA to a point which is 10 nm from the entry of PM, then they linearly change to 220 kts (knots) for the Medium or 230 kts for the Heavy at the beginning of the sequencing legs, after that, keep their speeds until "Turn" action, then they continuously reduce speed to 150 kts (knots) for the Medium or 180 kts for the Heavy on reaching M1 or M2, finally reach 140 kts both on FAF and maintain this speed until threshold of runway.

B. Flight efficiency performance index

Based on the flight performance data, we suppose that 25 kg/min is the common fuel consumption, denoted by μ_a, for all aircraft operating at BCIA in the simulation, then the amount of co_2 produced by each aircraft i, denoted by $E_i^{co_2}$, could be calculated consequently by the formulas:

$$J_i = \int_{t_i}^{t_i^L} \mu_a dt = \mu_a(t_i^L - t_i), \quad i \in F \tag{21}$$

$$E_i^{co_2} = 3.16 \times J_i = 3.16 \times \mu_a(t_i^L - t_i), \quad i \in F \tag{22}$$

through this rough assessment, the potential benefits could be estimated.

C. Numerical results

The experience-based parameter settings in SA are shown in Tab. V, detailed information could be referenced in [1]. We firstly design two scenarios, Test 1 is without runway change, Test 2 is with runway change. The overall performances with different elements, such as total number of conflicts, average delay, make-span, total flight time, etc., are presented and compared in Tab. VI.

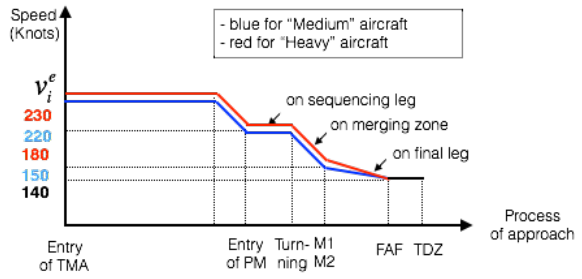


Fig. 7: Speed profile

TABLE V: Experience-based Parameters Configurations

Sliding Window	
Duration of window	3600 seconds
Window shifting interval	1800 seconds
Simulated Annealing Algorithm	
Initial Temperature for heating	0.01
Heating rate	1.1
Number of transition for heating or cooling	300
Cooling rate	0.97
Cooling stopping criterion	$T < 0.0001 \times T_{init}$

We can find that both cases have a conflict-free results, and the average delay, average landing interval, and make-span (the time segment between the first landing aircraft and the last landing aircraft) could decrease through runway re-assignment. From Test 1 to Test 2, the relative improvement for average delay is 36.36%, for make-span and average landing interval is 1.35% and 1.36% respectively. However, the average flight time, flight distance, fuel consumption and co_2 emission increase due to runway re-assignment, the relative increases are 13.49%, 1.11%, 13.49% and 13.49%, respectively. In Test 1, in order to generate a conflict-free trajectories, the adjustment on the t_i^e is much bigger than on Test 2, because without runway re-assignment, aircraft landing on the runway 01-19 is too much charged, in order to find conflict-free results, aircraft have to extend their entry time, which produces more delays, while with runway re-assignment, the maximum adjustment on the t_i^e could be reduced to only 180 seconds. Besides, with runway re-assignment function, we could balance or control the number of landing aircraft on two parallel runways, which is very useful to comply with the requirements from airport. Here, from Test 1 to Test 2, 49% aircraft change their initial landing runway, of which 81 aircraft change from 18L-36R to 01-19, 133 aircraft from 01-19 change to 18L-36R, as a result, 52 more aircraft land on runway 18L-36R in Test 2 compared with Test 1.

Furthermore, we look more precisely at each decision variables in Test 2, see Tab. VII. It is found that after optimization, 45% aircraft will increase their initial entry speed, 48% aircraft will decrease it, 7% aircraft will maintain it; 49% aircraft will prolong the initial entry time at TMA, 48% aircraft will advance it, 3% aircraft will not change it. Regarding t_i^T, 22% aircraft will choose to make an early turn, 28% aircraft will make a moderate turn, 25% aircraft will make a slightly late turn, 25% aircraft will make a very late turn.

On the basis of case 2 with runway reassignment in account, we continue to design several scenarios to study the differences

TABLE VI: Numerical Results Comparison 1

Elements	**Test 1**	**Test 2**	**Relative Changes**
number of unsolved conflicts	0	0	0.00%
average delay (min)	11	7	-36.36%
make-span (s)	76831	75793	-1.35%
average landing interval (s)	368	363	-1.36%
average flight time in TMA (min)	26	29	13.49%
average flight distance in TMA (km)	203	205	1.11%
average fuel consumption (kg)	639	725	13.49%
average CO2 emission (kg)	2019	2291	13.49%
maximum time change on t_i^e (s)	800	180	-77.50%
maximum speed change on v_i^e (%)	15	15	0.00%
number of a/c landing on 18L-36R	157	209	33.12%
number of a/c landing on 01-19	261	209	-19.92%

TABLE VII: Distribution of Decision Variables in Test 2

Changes on initial value \ Variables	Entry speed	Entry time
positive	45%	49%
negative	48%	48%
null	7%	3%

Per. of $[t^T_{imin}, t^T_{imax}]$ \ Variables	Turning time
0-25%	22%
25%-50%	28%
50%-75%	25%
75%-100%	25%

by controlling the landing rate on two different runways. Here, a new control parameter β is designed to control the probability of choosing runway 18R-36L as a landing runway, and with its value 0.2, 0.5, 0.8 and 1.0, four simulations are made, their results are shown in Tab. VII. In Test 6 all the aircraft land on runway 18R-36L, our algorithm could not generate conflict-free trajectories for all the aircraft, runway 18R-36L capacity is reaching its limitation, therefore some of the aircraft should deviate to land on runway 01-19. Similarly in Test 3, runway 01-19 reaches its capacity, and some of aircraft should deviate to land on runway 18R-36L. Test 4 and Test 5 well balance Test 3 and Test 6, conflict-free trajectories are both generated in these two cases. This study shows that within a range of β, it is possible to generate conflict-free trajectories for all arrival flights. Besides, according to the acceptance on different runways, the decision makers could choose the value of β as they prefer. Therefore, more scenarios are taken, the relative elements such as number of aircraft on runway, make-span, average delay and average landing interval are shown in Fig. 8. It is found that within a range [0.3, 0.9] for β, our algorithm could always find zero conflict results for all flights, and around $\beta = 0.4$, the near-optimal result would probably found, which corresponds to the a situation where two parallel runways accept almost same number of aircraft to land.

TABLE VIII: Numerical Results Comparison 2

Elements	**Test 3**	**Test 4**	**Test 5**	**Test 6**
value of β	0.2	0.5	0.8	1.0
No. of unsolved conflicts	90	0	0	212
make-span (s)	66205	65633	66264	65461
average delay (min)	8.95	8.72	8.69	9.0
average landing interval (s)	317	314	317	313
number of A/C on 18R-36L	155	221	265	418
number of A/C on 01-19	263	197	153	0

V. Conclusion and Remark

In this paper, three points should be highlighted: 1) topology of ML-PM system for operation on parallel runways with runway re-assignment in consideration is built in this paper. A trajectory-based reroute model is built to dynamically merge the arrivals to land at BCIA with two parallel runways. 2) the geometric relations for conflict detection in merging zone with non-equidistant merging distances between sequencing legs and merge point simultaneous operations are analysed, a new method for this kind of conflict detection is proposed. 3) ML-PM system has demonstrated the ability to automatically and dynamically control the arrival flights to land on parallel runways. Numerical results have shown that, this system has great benefit on automatic runway re-assignment to adapt the requirement of airport.

In the end, TMA environment is very complex, more work on the recent algorithm to increase robust-ability and efficiency of the ML-PM system is still required.

References

[1] M. Liang, D. Delahaye, and P. Maréchal, "A Framework of Point Merge-based Autonomous System for Optimizing Aircraft Scheduling in Busy TMA," in *5th SESAR Innovation Days*, Bologna, Italy, Dec. 2015. [Online]. Available: https://hal-enac.archives-ouvertes.fr/hal-01240314

[2] M. Liang, "An agent-based approach to automated merge 4d arrival trajectories in busy terminal maneuvering area," *Procedia Engineering*, vol. 99, pp. 233–243, 2015.

[3] E. Hoffman, "Point merge: Improving and harmonising arrival operations with existing technology," 2011.

[4] D. Ivanescu, C. Shaw, C. Tamvaclis, and T. Kettunen, "Models of air traffic merging techniques: evaluating performance of point merge," in *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, 2009.

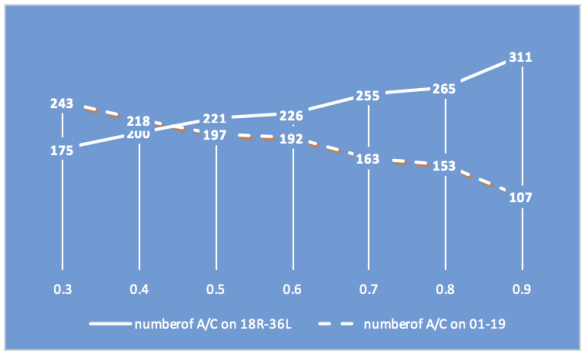
[5] L. Boursier, B. Favennec, E. Hoffman, A. Trzmiel, F. Vergne, and K. Zeghal, "Merging arrival flows without heading instructions," in *7th USA/Europe Air Traffic Management R&D Seminar*, 2007.

[6] H. Balakrishnan and B. Chandran, "Scheduling aircraft landings under constrained position shifting," in *AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO*, 2006.

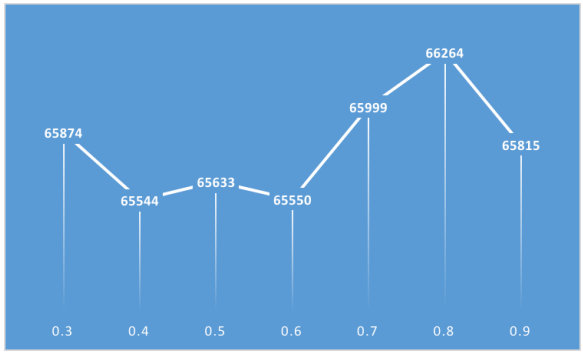
[7] G. C. Carr, H. Erzberger, and F. Neuman, "Fast-time study of airline-influenced arrival sequencing and scheduling," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 3, pp. 526–531, 2000.

[8] H. Lee and H. Balakrishnan, "A study of tradeoffs in scheduling terminal-area operations," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2081–2095, 2008.

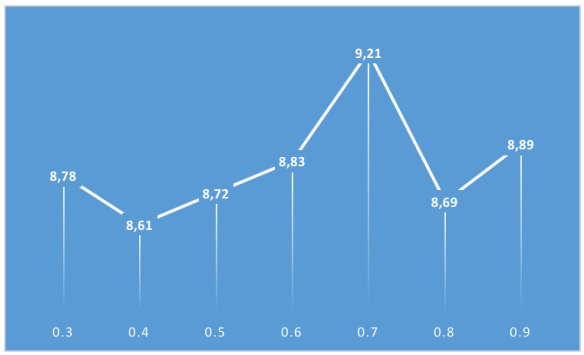
[9] T. Nikoleris, H. Erzberger, R. A. Paielli, and Y.-C. Chu, "Autonomous system for air traffic control in terminal airspace," in *proceedings of 14th AIAA Aviation Technology, Integration, and Operations Conference (ATIO 2014), Atlanta, GA, USA*, 2014.



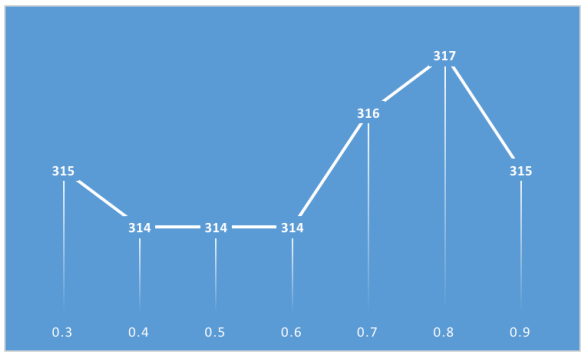
(a) Number of A/C on Runways



(b) Make-span (unit:s)



(c) Average Delay (unit:min)



(d) Average Landing Interval (unit:s)

Fig. 8: Numerical Results Comparison with Different β