HAL
archives-ouvertes.fr

A Framework of Point Merge-based Autonomous System for Optimizing Aircraft Scheduling in Busy TMA

Man Liang, Daniel Delahaye, Pierre Maréchal

▶ To cite this version:

Man Liang, Daniel Delahaye, Pierre Maréchal. A Framework of Point Merge-based Autonomous System for Optimizing Aircraft Scheduling in Busy TMA. 5th SESAR Innovation Days, Dec 2015, Bologna, Italy. <hal-01240314v2>

HAL Id: hal-01240314
https://hal-enac.archives-ouvertes.fr/hal-01240314v2

Submitted on 11 Dec 2015

A Framework of Point Merge-based Autonomous System for Optimizing Aircraft Scheduling in Busy TMA

LIANG Man
MAIAA Laboratory
Ecole Nationale de l'Aviation Civile
Toulouse, France
Email: man.liang@enac.fr

Daniel DELAHAYE
MAIAA Laboratory
Ecole Nationale de l'Aviation Civile
Toulouse, France
Email: daniel.delahaye@enac.fr

Pierre MARECHAL
Mathematical Institute of Toulouse
Université Paul Sabatier
Toulouse, France
Email: pr.marechal@gmail.com

Abstract—In this article we present recent work towards the development of an autonomous system with point merge (PM) that performs sequencing, merging and spacing for arrival aircraft in the busy terminal area. This autonomous arrival management system aims to safely solve the major arrival flight scheduling problems currently handled by human controllers. With PM, it has the potential to handle higher traffic demands without more workload on controllers, consequently increasing capacity and reducing delay. The main objective of this paper is to introduce the framework of this autonomous system with PM. Based on analysis of classic PM route structure, a novel PM-based route network is firstly designed for Beijing Capital International Airport. Vertically, this PM system consists of multi-layers on the sequencing legs for different categories of aircraft with Heavy and Medium, horizontally, it is shaped as a lazy "8". Then, a multiple-objectives function is discussed for this aircraft scheduling problem, operational constraints and conflict detection and resolution are analysed in detail, a modelling strategy with sliding time window and simulated annealing algorithm is proposed for solving this real-time dynamic problem. Experimental results verify our algorithm is well adapting the high-density traffic optimisation, and finally a conclusion is made and future work is pointed out.

Keywords—Air traffic management, Autonomous system, Arrival sequencing and merging, Point Merge

I. INTRODUCTION

Sustained air traffic demand growth is a major contributor to economic expansion but has led to congestion and significant delay at the busiest airports. According to the statistics report of year 2014 published by CAAC (Civil Aviation Administration of China) in May 2015, for the three busiest Chinese airports–Beijing, Guangzhou, Shanghai Pudong–, the on-time performances are only 69.69%, 68.82% and 56.25% respectively in 2014. Facing the severity of this problem, an urgent need is addressed by the CAAC: increased capacity, minimising environment impact, while maintaining or improving safety.

In busy airports, increasing its capacity means in a fixed time range, e.g. one hour, it could accept more aircraft to land on the runway. Because the makespan (duration of the whole sequencing) of all arrival aircraft depends on the wake turbulence separation, an optimal landing sequence will achieve a higher runway throughput. Due to the operational constraints from controller's workload in terminal control area (TMA), it is not easy to shift the position of aircraft in the landing sequence. Nowadays, First Come First Served (FCFS) and radar vector baseline methods are still widely used in today's operations for managing the arrival traffic. This traditional method has a good fairness between arrival aircraft and is also easily handled by controllers, while under high density traffic demand condition, it normally results in heavy delays.

Today's situation with radar vectoring makes a heavy controller workload, a great deal of radio communication, diminution of pilot situational awareness, difficulty in predicting and improving vertical profiles and large dispersion at low altitudes. Therefore, automation in air traffic operations are likely to be needed to handle the denser and more diverse mix of air traffic in busy airports.

As a transition from today's radar vector baseline operation to a fully automated terminal management system, our research aims to make a large portion of routine works of controller, mainly sequencing, merging and spacing, more autonomous. With the application of new emerging technologies in Communication, Navigation and Surveillance (CNS), and also point merge route network concept, we manage to make the trajectories of arrival aircraft more orderly and efficient, finally enabling significantly increased throughput at the busiest airports.

II. POINT MERGE

A. Classic PM System

Point Merge (PM) is designed to work in high traffic loads without radar vectoring. As showed in Fig.1, It is based on a specific P-RNAV route structure, consisting of a point, named the *merge point*, and pre-defined legs, named the *sequencing legs*, equidistant from this point. The sequencing is achieved with a *direct-to* instruction to the merge point at the appropriate time. The legs are only used to delay aircraft when necessary, similar to path stretching method; the length
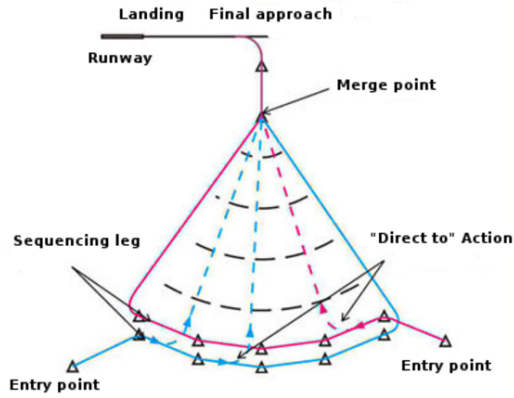
Fig. 1.   Classical Topology of PM System

of legs reflects the required delay absorption capacity. [1], [2].

Referring to the operational experiences described by Euro-control, PM could provide benefits in terms of safety, environment (in approach sectors) and capacity (in terminal sectors), even with high traffic loads. Depending on the operational and environmental constraints, and on the design choice made, some expected benefits may be gotten as below:

1) simplification of controller tasks, reduction of communications and workload;
2) better pilot situational awareness;
3) more orderly flows of traffic with a better view of arrival sequences;
4) improved containment of flown trajectories after the merge point;
5) better trajectory prediction, allowing for improved flight efficiency;
6) standardisation of operations and better airspace management.

Standardisation of operations and simplification of controller tasks are two key benefits of PM system for successfully adapting to an autonomous system for sequencing, merging and spacing arrival aircraft. Moreover, sequencing legs provide a good way to easily change the position of aircraft in the sequence. Later, we will design a PM-based route network for studying the autonomous arrival management in TMA of busy airport, such as Beijing Capital International Airport (BCIA).

B. Design of PM-based Route Network for BCIA

We choose BCIA as a study case. BCIA ranks the second busiest airport in the world, there are three parallel runways: 18R-36L, 18L-36R, and 19-01. Independent parallel departures are used in all of these three runways, and runway 18R-36L and 19-01 are used with independent instrument parallel approaches. In Beijing TMA, arrival flows come mainly from the South, where there are 4 points of entry: JB, BOBAK, VYK and DOGAR, on the North, there are 2 points of entry: KM and GITUM. In order to efficiently merge the
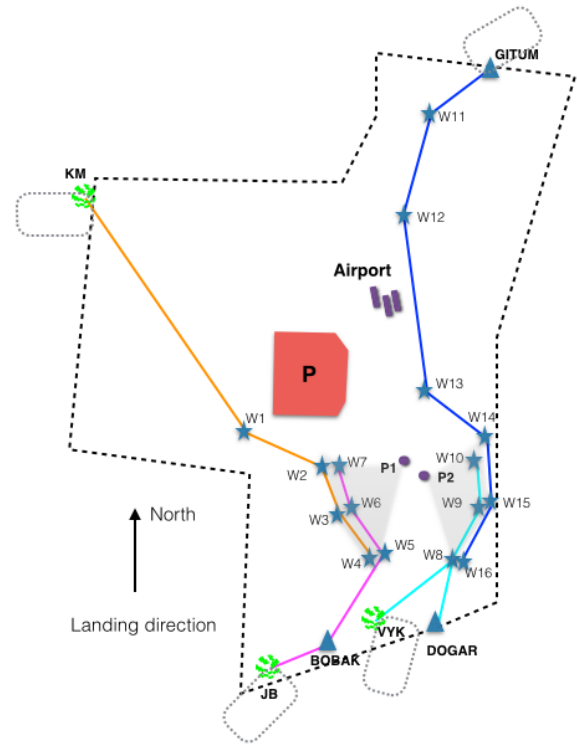

Fig. 2.   PM-based Route Network for BCIA

arrival traffic, aircraft from different departure airports will follow different arrival procedures to land on the right runway. As illustrated in Fig.2, based on the RNAV procedures in the published standard instrument arrival chart (STAR) for RWY 36L/36R/01, a new PM-based arrival routes network is designed. The arrival trajectory is separated into three phrases, the first phrase is from the entry points of Beijing TMA to the entry point of PM system, the second phrase is from entry point of PM system to the merge point of PM system, the third phrase is from merge point to the runway. In details, aircraft from the north follow the route GITUM-W11- W12-W13-W14 to arrive at merging zone of P2, aircraft from the west follow the route KM-W1-W2 to arrive at merging zone of P1, aircraft from south are separated into two groups, one group of aircraft follow JB-BOBAK-W5 to the merging zone of P1, the other group of aircraft follow VYK-W8 and DOGAR-W8 to the merging zone of P2. Aircraft on sequencing legs W2-W3-W4 or W5-W6-W7 will merge to P1 initially, then land on the runway 18R-36L. Aircraft on sequencing legs W8-W9-W10 or W13-W14-W15 will merge to P2 initially, then land on the runway 19-01. Aircraft on P1 and P2 have 300 meters vertical separation. If the traffic demand is higher than the airspace capacity, then four standard holding procedures designed on entry waypoints KM, JB, GITUM, VYK will be used for absorbing this part of overloaded traffic. From a plan view, we can find that the PM-based route network for BCIA is similar to lazy "8" shape, because of limited airspace available on the south.

In order to significantly make arrival traffic robust, a multi-layers concept on sequencing legs is also designed for separating different category of aircraft. Based on the statistics of traffic at Beijing International airport from 2000 to 2014, normally there were no "light" aircraft arriving, hence we could design two layers for each sequencing leg, in consideration of the noise and fuel consumption, the upper layer is for "Heavy" and the lower layer is for "Medium" aircraft, they have 300 meters vertical separation, while both layers will have an unique projection on the horizontal plane. In total, the part of sequencing legs in PM system will be like in Fig.3 . Outer sequencing leg W2-W3-W4 or W14-W15-W16 have minimum 2 NM lateral separation from inner sequencing leg W5-W6-W7 or W8-W9-W10. The lower layer on each inner sequencing leg is 300 meters higher than the upper layer of the outer sequencing legs. For example, on leg W2-W3-W4 there are two available layers, one is 1500 meters, the other is 1800 meters, while on leg W5-W6-W7, one is 2100 meters and the other is 2400 meters. Aircraft with "Heavy" on W2-W3-W4 will maintain on level 1800meters, while on W5-W6-W7 will maintain 2400 meters.

With this kind of sequencing leg structure design, same category of aircraft will fly along the same sequencing leg layer, if they keep the same speed, then the conflict detection and resolution process on the sequencing legs will be simplified, so as to provide a good way for realising an autonomous arrival management.

III. MATHEMATICAL MODEL FOR AIRCRAFT SCHEDULING PROBLEM

A. Given Data and Assumptions

Assume that there are a set of aircraft, $\mathcal{F} = \{1, 2, 3, \ldots, n\}$, each aircraft i ($i \in \mathcal{F}$) has the following predetermined information:

- E_i, initial entry waypoint at TMA;
- t_i^e, arriving time at E_i;
- v_i^e, speed at E_i;
- fl_i, level at E_i;
- W_i, wake turbulence category of aircraft i;

Then, a set of routes, $\mathcal{R} = \{r_g \mid g \in N\}$, are also defined for aircraft to fly. One route is composed of several segments, each segment is defined by two waypoints, each aircraft will follow exactly one route. As showed in Fig.2, there are in total five routes:
$r_1 = \{KM, W1, W2, W3, W4\}$,
$r_2 = \{JB, BOBAK, W5, W6, W7\}$,
$r_3 = \{GITUM, W11, W12, W13, W14, W15, W16\}$,
$r_4 = \{VYK, W8, W9, W10\}$,
$r_5 = \{DOGAR, W8, W9, W10\}$.

After that, we have to make some assumptions to simplify our study problem. Firstly, because the data of wind grid is not available up to now, so in this research case, we will not account the wind effect, then aircraft airspeed is equal to ground speed; secondly, the initial entry time of TMA of arrival aircraft refers to the planned time, hence there may be

conflict between two successive aircraft; thirdly, there are no path change for aircraft to resolve the conflict before it enters the PM system; Fourthly, aircraft on the sequencing leg will maintain its level.

B. Objective Function

A lot of research works have been done to study the problem of scheduling aircraft in TMA. [3] studied the scheduling problem of maximising runway throughput under CPS (Constrained Position Shifting). [4] investigated a sequencing algorithm to take account of airline priorities. While, at the same time, some researchers attempted to find the trade-offs between different stakeholders' interests. In [5], it is indicated that: firstly, the significant improvement in the average delay could be achieved through re-sequencing under CPS. Secondly, in most case the re-sequencing using CPS improves both the makespan and the average delay when compared to the FCFS solution. Thirdly, on average maximising the throughput only resulted in modest increases in the fuel costs. Based on the analysis above, under the high density operation environment, we choose to simultaneously optimise three objectives:

1) minimising the average landing time interval between aircraft, hence maximising landing rate, maximising the capacity.
2) minimising the average delay.
3) minimising the average conflict with each aircraft.

The multi-objectives function is defined as below:

$$z = Min\{\alpha T + (1 - \alpha)D + C\} \tag{1}$$

$$T = \frac{m}{n} Max\{S^1, ..., S^m\} \tag{2}$$

$$D = \frac{1}{n} \sum_{i=1}^{n} (t_i^L - ETA_i^L) \tag{3}$$

$$C = \frac{1}{n} \sum_{i=1}^{n} c_i \tag{4}$$

- T is the average landing interval,
- D is the average delay,
- C is the average duration of conflict,
- n is the number of flights,
- m is the number of parallel runways,
- t_i^L is the actual arrival time of flight i on the runway,
- ETA_i^L is the estimated time of arrival of flight i on the runway,
- S^m is the makespan of the landing aircraft for runway m,
- c_i is the duration of conflicts with aircraft i, finally its value should be zero after the conflict resolution process,
- α is the control parameter, if we more focus on delay, its value could be smaller, if we more focus on throughput, its value could be bigger. This control parameter could be dynamically changed, according to the severity of delay on the airport, so as to control the flow of arrival flights, here we set its value with 0.5.
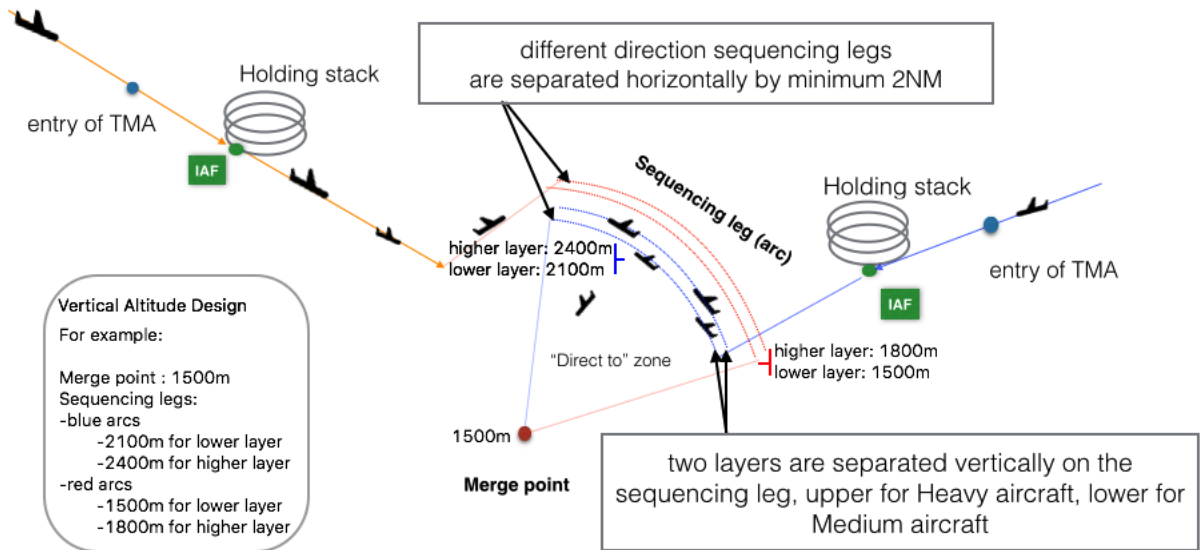
Fig. 3.   Vertical and Horizontal Spacing between Different Sequencing Legs

C. Constraints

1) Maximum number of position shift in the sequence: The maximum number of position shifts allowed (CPS) is denoted by k. It has been noted that the reasonable values of k for CPS might be 1, 2, or 3 [6]. In almost all current ATC automation systems, very limited overtaking is allowed, normally only 1 position shift, while due to the nature of our M-PM route structure, a more relaxed position shift could be allowed, a slightly bigger value of k may not impose too much workload to controller, so here we could choose $k \leq 3$.

2) Arrival time windows: Estimated Time of Arrival (ETA) of flight at the entry point of TMA can vary, let us denote this range as δt_i^e. Normally the earliest time of arrival is usually limited to 1 minute before the ETA because of the resultant fuel expenditure, but if we consider mostly average delay decrease for arrival flights, then 3 minutes allowed time advance is feasible [5]. The latest arrival time is determined either by fuel limitations or by the maximum delay that a flight can incur, in our case, we choose the maximum delay of 10 minutes for the constraint of latest arrival time.

3) Minimum aircraft separation: ICAO regulates the minimum spacing between landing aircraft to avoid the danger of wake turbulence. It is a distance-based separation under radar control environment. We also have to consider approach radar separation between two successive aircraft with the same category of wake turbulence. Given two successive aircraft i and j, the required horizontal aircraft minimum separation $s_{i,j}$ in TMA is listed in Table.I, here in our case, we consider only Heavy and Medium two categories. Denote the horizontal distance between aircraft i and j as $d_{i,j}$, then $d_{i,j}$ normally must be always bigger than $s_{i,j}$. However, in our study, we have to consider delay and capacity as well, maybe in some cases it is very difficult to find the conflict-free resoltion, so

TABLE I
DISTANCE-BASED AIRCRAFT MINIMUM SEPARATION (UNIT: NM)

Leading ╲ Trailing	Heavy	Medium
Heavy	4	5
Medium	3	3

in order to maximum reduce the conflicts, we choose to put a bigger weight for the conflict in our multi-objective function, so as to relax this operational constraints.

4) Maximum turning time on sequencing leg: The aircraft on the sequencing legs have to turn before the end of sequencing leg. Denoting the maximum maintaining time on the sequencing leg as t_{imax}^T, the turning time on the sequencing leg as t_i^T.

In sum, the operational constraints can be described as below:

$$s.t \begin{cases} d_{i,j} \geq s_{i,j} & \forall i,j \in \mathcal{F} \\ k \leq 3 & \forall k \in Z \\ \delta t_i^e \in [-3mins, 10mins] & \forall i \in \mathcal{F} \\ t_i^T \leq t_{imax}^T & \forall i \in \mathcal{F} \end{cases} \quad (5)$$

D. Conflict Detection

For safety consideration, all the aircraft in this autonomous system should be well lateral separated, except they are on the sequencing legs where there are already sufficient spacing either on vertical or on horizontal. In our case, three kinds of conflicts are taken into account: out of merge zone, link conflict, node conflict, showed in Fig.4, and in the merge system, turn conflict, showed in Fig.5.
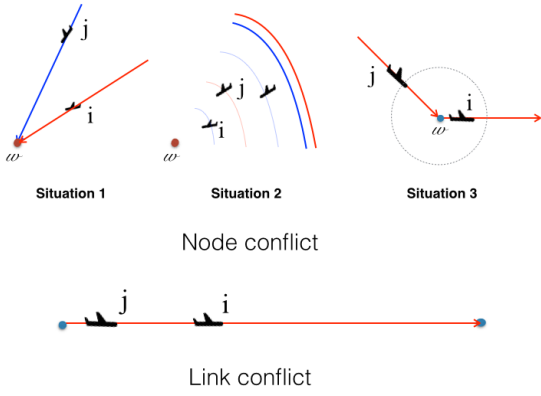
Fig. 4. Conflict Detection out of Merge Zone: Link and Node

Leading \ Trailing	Heavy	Medium
Heavy	80	100
Medium	60	60

Fig. 5. Conflict Detection in Merge Zone: Turning time

1) Link Conflict: If two successive aircraft follow the same route, their position in the sequence will not change before PM system. As a route segment is composed of two waypoints, so for a given route segment, we verify twice whether a link conflict occurs at the entry and at the exit of route segment, $d_{i,j} \geq s_{i,j} \quad \forall i,j \in \mathcal{F}$. The speed will remain constant during this process , so if no conflict is detected either in entry nor in exit, then there is no conflict in the whole route segment. Otherwise, conflict resolution will be applied. In PM system, same category of aircraft with the same entry point will fly on the same sequencing leg, because they keep the same speed, so it is unnecessary to detect the link conflict.

2) Node Conflict: There are three situations relating to the node conflict. Situation 1: a common exit waypoint is shared by two route segments, such as waypoint W8 in Fig. 2. Situation 2: aircraft leave the sequencing leg and fly toward the merge point. Situation 3: two successive aircraft on the same route, aircraft i is just passed the waypoint w, while aircraft j is ready to pass the same waypoint w. Node conflict is different from link conflict. In situation 1 and 2, as two aircraft approach the common point, their spacing is reducing, and before they reach the common point, at specific time, $d_{i,j}$ may be already less than the $s_{i,j}$. In situation 3, with the change of fly direction between two connecting route segments, $d_{i,j}$ may be less than $s_{i,j}$ according to triangle rule. Therefore, in order to efficiently detect the merging conflict, we transfer the distance-based conflict detection to time-based conflict detection. Referring to the performance of commercial aircraft with Medium and Heavy category (normally 150kt on final approach phrase), and wake turbulence will dissipate more quickly under a wind condition, hence a 180 knots is chosen as a reference speed, then the minimum time-based wake turbulence separation $s_{i,j}^T$ for detecting the merging conflict could be calculated, the results are showed in Table.II. Time-based separation plays another important role, that is to calculate the objective function.

Assumpt that the passing times at the common point w by aircraft i (leading) and j (trailing) are denoted t_i^w and t_j^w, then
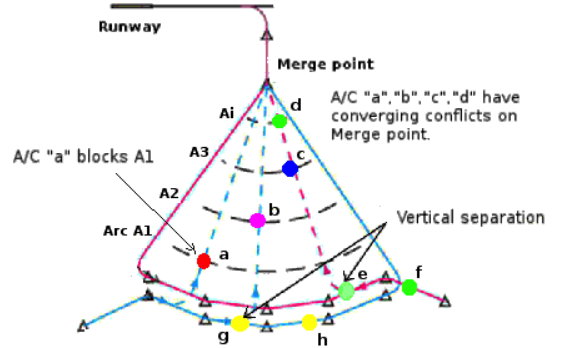
for node conflict, the conflict detection function will be:

$$t_i^w \leq t_j^w - s_{i,j}^T, \quad \forall i,j \in \mathcal{F}. \tag{6}$$

3) Turn Conflict: As illustrated in Fig.5, a turn conflict should be detected for aircraft turning to the same merge point. All the aircraft in the merging zone should be kept laterally separated, which means once aircraft a is flying to arc $A1$, it will block other aircraft turning to arc $A1$. When a is continuing to arc $A2$, then arc $A1$ is released for other aircraft. We will control the turning time of each aircraft in order to avoid the conflict between two successive aircraft approach to the same merging point. Assumption that the turning times to point p by aircrft i (leading) and j (trailing) are denoted t_i^T and t_j^T, then for turn conflict, the conflict detection function will be:

$$t_i^T \leq t_j^T - s_{i,j}^T, \quad \forall i,j \in \mathcal{F}. \tag{7}$$

E. Decision Variables

In our autonomous system, for solving the conflicts and controlling the rate of arrival flows, we consider two possible manoeuvres: modify the entry time t_i^e at TMA or adjust the speed; for optimising the landing sequencing, we consider to control the turning time t_i^T of aircraft on the sequencing leg.

1) Modification of Entry Time at TMA: From an operational point of view, the arriving time change and speed change are usually discrete, moreover controller radar display usually updates every 5 seconds, so the discretisation of changes on entry time at TMA and speed will be more practical in the real world. Given $\Delta t = 5s$ as time discretised interval, for each aircraft $i \in \mathcal{F}$, it exists $\delta t_i^e \in [-3mins, 10mins]$, then the space χ of decision variable t_i^e change is equal to $\delta t_i^e / \Delta t$. We define the number of slot as $j \in \chi$, the new entry time of

aircraft i at the TMA as t'^e_i, then:

$$t'^e_i = t^e_i + j * \Delta t \quad | \; j \in Z \text{ and } j \in \chi. \tag{8}$$

2) Adjustment of Speed: In TMA, controller usually reduces the speed of aircraft by a discretised value, and for Medium and Heavy aircraft, their speed below 10000ft is normally less than 250knots, what is more, considering that significant acceleration during approach process is not a reasonable way for fuel saving, therefore the range of the speed change is very limited.

For each aircraft $i \in \mathcal{F}$, we define the percentage of change as $g \in \psi$, and $\psi \in [-20\%, 5\%]$, the new speed of aircraft i as v'_i, then:

$$v'_i = v_i * (1 + g) \quad | \; g \in \psi. \tag{9}$$

3) Ajustment of Turing Time on the Sequencing Leg: Change the t^T_i is limited by the length of the sequencing leg. Let us define the new turning time as t'^T_i. The earliest turning time is the entry time of PM system, denoted as t^T_{imin}, the latest turning time is maximum turing time t^T_{imax}, then the change range of t^T_i is between $[t^T_{imin}, t^T_{imax}]$. Assumption that the percentage of change from t^T_{imin} is h, and then its change state space is defined as $\Phi \in [0\%, 100\%]$, then it exits:

$$t'^T_i = t^T_{imin} + h * (t^T_{imax} - t^T_{imin}) \quad | \; h \in \Phi. \tag{10}$$

In total, we have three parameters for indirectly changing the state of aircraft, they are j, g, h, named control parameters.

IV. RESOLUTION APPROACH FOR AUTONOMOUS ARRIVAL MANAGEMENT

The autonomous arrival management system should match the requirement of dynamic situation. Receding Horizon Control (RHC) is a N-step-ahead on-line optimisation strategy, it could reduce the dynamic aircraft scheduling problem into a sequence of static sub-problems based on a sliding time window [7]. [8] and [9] introduced the concept of RHC into Genetic Algorithm and Ant Colony Algorithm to solve the problem of arrival scheduling and sequencing at a busy hub airport, which proved that RHC could well realise a real-time implementations in a dynamic environment of air traffic control. Here, we will apply the RHC strategy into the autonomous system.

A. Sliding Time Window Approach

As illustrated in Fig.6, the overall time horizon of 24 hours is firstly divided into smaller time horizons of prediction with different start times. The difference between two consecutive start times is named roll period. According to the relative relationship between the aircraft life cycle and the Sliding time window in the whole timeline, we classify the status of aircraft into 4 types: Completed, On-going, Active and Planned. *Completed* means that the aircraft's trajectory is already fixed. *On-going* means some part of the aircraft trajectory is still on the Sliding time window, some part of their trajectory is not changeable while some part of their trajectory

are changeable. *Active* means the aircraft trajectories could be changed. *Planned* is the rest part of aircraft who are not belong to any types mentioned before. After that, those aircraft with *Active* and *On-going* status will be selected to enter the optimisation procedures inside the sliding time window.

In the sub-optimisation procedures, it consists of two key components: one is to sequence the aircraft, the other is to merge the arrival flow without conflict. According to the objectives of this scheduling problem, we could change the speed of aircraft, its entry time in the TMA, or it's turning time on the sequencing leg. With respect of operational reality and producing less impact on the adjacent sectors, if possible, we would prefer to search the solution from turning change decision variable firstly, then the speed change, finally the entry time change, therefore a strategy to select the decision variables is designed for modification of these three decision variables, it is described as below:

- define two parameters $p_{turn}, p_{speed} \in [0, 1]$,
- random produce a value for $p \in [0, 1]$, which is used for controlling the selection of decision variable,
- if p less than p_{turn}, adjust the turn time, if p less than p_{speed}, change the entry speed, else change the entry time.

B. Simulated Annealing Algorithm

The sub-optimisation process in the sliding time window could be resolved by application of Simulated Annealing (SA) algorithm. SA is a meta-heuristic inspired by the annealing process in metallurgy. It consists in bringing the system from a disordered random state to a global-minimum energy state, involving heating process and cooling process. SA is will known for its ability to trap out of the local minimum by allowing random neighbourhood changes, moreover it can be easily adapted to different kinds of problems with continue or discrete space states.

In order to successfully apply the SA algorithm to our problem, we must specify the relative parameters. Fig. 7 presents the simulated annealing heuristic process for our problem. It starts from a warm up algorithm to gain an initial temperature T_0 and state s_0, and then follows a cooling down algorithm to a T_{min}. In the process, the call $neighbour(s)$ should generate a chosen neighbour of a given state s; the call $random(0, 1)$ should pick and return a value in the range $[0, 1]$. The annealing schedule is defined by the call $T(k)* = (1 - k)$, k is the cooling rate, which should yield the temperature to use. function $P(E(s), E(s_{new}), T)$ is the acceptance probability from state s to new state s_{new}. Choosing neighbours is very important for finding the good solution as quickly as possible. Because de-conflict process is most important objective in the program, so we will prefer to find the neighbour who has more conflicts, then because "Heavy" aircraft play more impact on the delay and capacity, so we will also prefer to choose "Heavy" aircraft as the neighbour. Finally we design a parameter named *perfo* for each aircraft, which balances the number of conflict and wake turbulence, it is a value between $[0, 1]$, after that the neighbourhood function will use it to find a good neighbour, the Pseudo code is showed in Fig.8.
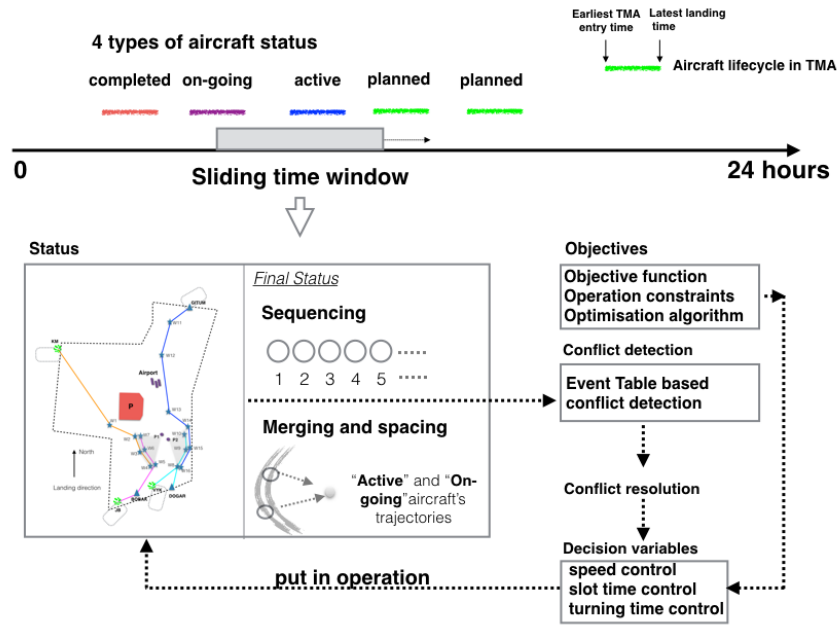
Fig. 6. Modelling Method with Sliding Time Window

Algorithm 1: Simulated Annealing

1: Set initial temperature T0 = warm up algorithm();
2: Let s = s_0
3: Initiate the control parameters: j, g, h
4: Make the objective function result as the energy function E(),
   calculate initial energy E(s_0), set initial E(s_{best})=E(s_0).
5: While T > T_{min} *the stopping criterion
6:     Pick a neighbour s, s_{new} ← neighbour(s)
       change the control parameters: j, g, h
                                   * neighbour selecting
7:     Calculate the E(s_{new})
8:     Calculate the acceptance probability P(E(s), E(s_{new}), T),
       If P(E(s), E(s_{new}), T) ≥ random(0,1)
           move to the new state: s ← s_{new}
9:     end If
10:    If E(s_{new})<E(s_{best}),
           change to the best state: s_{best} ← s_{new}
11:    end If
12:T ← T*(1-k)   *k: the cooling rate
13:Output: the final state s_{best}

Fig. 7. Pseudo Code of Simulated Annealing Algorithm

Algorithm 2: Neighbourhood function

Input: *Perfo* of each aircraft in the active window
Output: index of neighbour

1: for iStart < i < jEnd do
2:    sumPerfo ← calculate the sum of the *Perfo(i)* of aircraft i
3:    if (sumPerfo>0)
       set a target P(t)← random (0,1) *sumPerfo
4:       do calculate the sum of *perfo(i)* from iStart and give value to P(i)
         P(i)← sum of *Perfo(i)*, (iStart < i < jEnd)
       until P(i)<P(t)
5:         find out the index of neighbour i and return this value
6: else
7:    random set the index of neighbour i and return this value
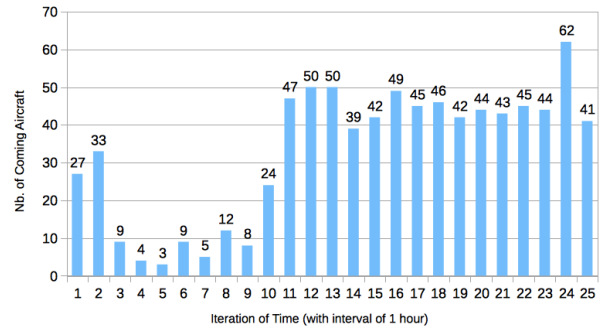
Fig. 8. Pseudo Code of Neighborhood Function

Fig. 9. Traffic Flow at BCIA on Nov. 6th 2015

V. EXPERIMENTS AND RESULTS

We choose the real data of BCIA on Nov.6th 2015 as scenario simulation input. There are 823 flights to land at BCIA in 24 hours, of which 78.5% are "Medium", 21.5% are "Heavy", remark that 12.27% traffic come from KM, 17.98% from JB, 47.63% from VYK, 11.18% from DOGAR, and 10.94% from GITUM. Because VYK connects to A461 and A593, which are the two busiest air routes in China, the traffic from VYK is much heavier than other entry points. In a whole view, the traffic coming from south is more than three times from the North. The distribution of hourly traffic flow entering the TMA of BCIA is presented in Fig.9, the average number of aircraft per hour is around 48, and the peak period on this day is in the 24th iteration with 62 flights, the high-density period is from the 11th to the 25th iteration, and the low-density period is from the 3th to the 9th iteration, the normal-density period is the 1th, 2th and 10th iteration.

Sliding Window

Duration of window	5400 seconds
Window shifting interval	1800 seconds
Uncertainty of t_i^e	± 120 senconds
p_{turn}	0.4
p_{speed}	0.6

Simulated Annealing Algorithm

Initial Temperature for heating	0.01
Heating rate	1.1
Number of transition for heating or cooling	200
Cooling rate	0.95
Cooling stopping criterion	$T < 0.0001 * T_{init}$

Speed on Sequencing Legs and Final approach

"Heavy" Aircraft	230 knots, 180 knots
"Medium" Aircraft	220 knots, 150 knots

TABLE IV
EXPERIMENTAL RESULT COMPARISONS

	HighDensity	NormalDensity	LowDensity
Nb. of aircraft	642	70	65
Duration of traffic(h)	15	2	7
Initial Conflict	61691.98	3879	67.36
Single Objective Optimisation-Conflict			
Conflict	36.7	0	0
Average Delay (s)	47.63	22.78	139.6
Ave. Land Interval (s)	155.27	246.89	813.36
Multi-Objective Optimisation-Conflict,Delay,Capacity			
Conflict	98.49	1.15	0
Average Delay (s)	28.26	20.05	95.05
Ave. Land Interval (s)	152.60	236.92	804.3
Elapse Time (s)	400s	23s	8s

Experimental tests are carried out in this section to verify the PM-RHC-SA algorithm. Three cases are studied, they are: low density situation, normal density situation and high density situation. Some user-defined parameters in this algorithm are listed in Tab.III. Tests are implemented in Eclipse IDE for Java Developers (Version: Luna Service Release 2) on a MacBook Air running a CPU 1,4 GHz Intel Core i5 with a memory of 4 GB 1600 MHz DDR3. The results are illustrated in Tab.IV.

The results show that under normal and high density situation, it is more difficult to find out the good decision variable to produce a conflict-free trajectories. Comparing with only consideration of conflict in the objective function, the results of multi-objective optimisation show that the average delay and the average landing interval are improved simultaneously. However, they are improved in compromising on conflicts. Under high-density situation, the landing interval could not be improved very much, that is because the use of runway already researches the maximum. Besides, after optimisation, most of the conflicts are well solved, but in same cases they can not be reduced to zero, the reason should be the probability

parameter setting of the decision variables, sliding window parameters setting etc..

VI. CONCLUSION AND PERSPECTIVES

In this paper, a concept of PM-based autonomous system for optimising aircraft scheduling in busy TMA is proposed. Four highlighting points should be listed: first, a novel PM route system was designed for BCIA, it has multi-layer sequencing legs on vertical and lazy "8" shape on plan view. Second, the objective function balances different stakeholders' interests, specially delay, throughput and safety. Third, time-based separation is applied to detect conflict, which simplifies the calculation. Fourth, a modelling approach with sliding time window and simulated annealing algorithm is proposed to solve the problem. Scenarios are studied based on the real data of BCIA, the results show that this PM-RHC-SA algorithm has great potential to handle the high-density situation.

In the next step of our research, we will continue to improve the performance of our algorithm, to study the control parameters sensitivities, the speed change strategy and the vertical descent profile in the merging zone.

REFERENCES

[1] EUROCONTROL, *Point Merge Integration of Arrival Flows Enabling Extensive RNAV Application and CDA Operational Services and Environment Definition*, Version2.0. Eurocontrol Experimental Center, 2010.
[2] B. Favennec, E. Hoffman, A. Trzmiel, F. Vergne and K. Zeghal, *The Point Merge Arrival Flow Integration Technique: Towards More Complex Environments and Advanced Continuous Descent*, 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO). South Carolina, September 21-23, 2009.
[3] H. Balakrishnan, B. Chandran, *Scheduling aircraft landings under constrained position shifting*. AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, August, 2006.
[4] Gregory C. Carr, Heinz Erzberger, Frank Neuman, *Fast-time study of airline-influenced arrival sequencing and scheduling*. Journal of Guidance Control and Dynamics, vol.23, (no.3), 2000, p.526-531.
[5] H. Lee, H. Balakrishnan, *A study of tradeoffs in scheduling terminal-area operations*. Proceedings of the IEEE, vol.96, (no.12), 2008, pp. 2081-2095.
[6] R. G. Dear, *The dynamic scheduling of aircraft in the near terminal area*. Cambridge, Mass.: Flight Transportation Laboratory, Massachusetts Institute of Technology, 1976.
[7] HU Xiao-Bing, CHEN Wen-Hua, *Receding horizon control for aircraft arrival sequencing and scheduling*. Intelligent Transportation Systems, IEEE Transactions on, 2005, vol. 6, no 2, p. 189-197.
[8] ZHAN Zhi-Hui, ZHANG Jun, LI Yun, et al, An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem. Intelligent Transportation Systems, IEEE Transactions on, 2010, vol. 11, no 2, p. 399-412.
[9] HU Xiao-Bing, CHEN Wen-Hua, Genetic algorithm based on receding horizon control for arrival sequencing and scheduling. Engineering Applications of Artificial Intelligence. 2005, vol. 18, no 5, p. 633-642.