Air Traffic Structuration based on Linear Dynamical Systems

Paveen Juntama, Sameer Alam
ATMRI
Nanyang Technological University
Singapore

Supatcha Chaimatanan
SOAR Research Lab
GISTDA
Chonburi, Thailand

Daniel Delahaye
OPTIM Research Lab
ENAC
Toulouse, France

***Abstract*—This paper presents a novel air traffic structuration approach to maintain flows of air traffic and to adapt traffic situations, which can reduce the mental workload of air traffic controllers. We reformulate the optimization problem by reorganizing the aircraft trajectories in space (e.g. aircraft rerouting) or time dimension (e.g. rescheduling time of departure, flow crossing, time based metering) or both in some areas where the system identifies a high level of disorder in the traffic structure. To structure the traffic, an air traffic complexity metric based on linear dynamical systems is used for this optimization problem. To minimize the impact of traffic structure, we propose an adaptive metaheuristic approach with the integration of reinforcement learning for our resolution algorithm. The resolution algorithm is applied for short-term (flow crossing, time-based metering, and traffic encounters) trajectory planning applications and national scale planning under time uncertainty in French airspace. For short-term scenarios, our approach can restructure the traffic which allows controllers to take less effort for managing traffic situations. Our solution also improves the traffic structure with approximately 50 % reduction of air traffic complexity at national scale. Our research findings introduce further steps toward taking other trajectory structuration techniques into account and developing new search strategies to our resolution algorithm.**

***Keywords*—traffic structuration, air traffic complexity, linear dynamical systems, adaptive metaheuristic, reinforcement learning**

I. Introduction

The air transportation system has been facing an ever-increasing demand for air travel since the emergence of commercial aviation. For instance, the average European traffic was 30 427 flights per day with a maximum traffic demand of more than 37 000 daily flights [1]. The rapid increase in air traffic demand is causing serious airspace congestion due to limited capacity. Optimizing capacity usage is an innovative strategy within capacity management processes in current ATM system.

The capacity of current ATM system is limited by the maximum workload that air traffic controllers (ATCs) can use their ability to provide safe and efficient traffic flows through airspace sectors. A major source factor of ATC workload is *air traffic complexity*. The air traffic complexity is often defined as difficulty of monitoring and managing air traffic situations. Due to complexity in traffic pattern, ATCs may refuse some aircraft entering their sector less than usual if incoming aircraft will increase the level of difficulty for ATCs to maintain the appropriate separation of aircraft and identify potential conflicts.

In European airspace, Network Managers Operation Center (NMOC), previously known as the Central Flow Management Unit (CFMU) is the operational unit of Eurocontrol that performs strategic tasks (flight planning and slot allocation) aimed at adapting traffic demand to capacity before the day of operation and tactical intervention for which ATCs are mainly responsible.

In the context of research, SESAR introduces the Advanced Demand and Capacity Balancing (DCB) concept to identify hotspots through an assessment of traffic complexity and ATC workload for more accurately predicting DCB imbalances and traffic demand. Due to different traffic structures in European airspace, standardization of a complexity metric and DCB resolution remains challenges in this concept. Several complexity metrics have been reviewed in [2]. The complexity assessment on a long-term horizon can identify congested areas and support strategic flight planning, whilst complexity metrics for a mid/short horizon can help identify situations that are significant for distributed conflict resolutions.

This paper assesses the significance of traffic structuration problem in Trajectory-Based Operations (TBO). We propose the modification of departure time and en-route trajectory to the aircraft for minimizing the impact of traffic structure in airspace. An air traffic complexity metric based on linear dynamical system is introduced to quantify the impact of air traffic structure. Moreover, the new adaptive metaheuristic based on reinforcement learning is used to solve this problem. Empirical studies with short-term and long-term traffic scenarios show that the proposed methods are benefit to restructure traffic in terms of algorithm and quality of the traffic organization. This paper is organized as follows: Section II presents the previous related works. Section III gives the mathematical formulation of traffic structuration problem. Our resolution algorithm is presented in Section IV. Section V reports the experimental results. Finally, Section VI concludes our research work in this paper.

II. Previous related works

A. Complexity metrics

The Dynamic Density (DD) suggested by Laudeman et al. from NASA [3], is the first assessment of air traffic complexity considering both number of aircraft and traffic structures. Combining traffic features produces a single positive real number reflecting the level of complexity. In studies of Sridhar et al. [4], the DD is also used to determine a predictive model beforehand up to a given time horizon. The features used in the DD are not enough to explain airspace complexity. This motivation has driven the development of new approaches to complexity measures that are independent from traffic characteristics, such as the fractal dimension [5], the input-output approach [6], and the intrinsic complexity [7].

Intrinsic complexity metrics firstly introduced by Delahaye et al. [7], can better quantifies the airspace congestion than a simple number of aircraft. Aircraft position and speed vector are used to compute such metrics. The research have investigated two classes of indicators. The first class uses geometrical properties to build a metric. The second one formalizes a representation of air traffic as a dynamical system. This class can distinguish different air traffic situations: translation, convergence, divergence and rotation. Few studies have been performed to investigate the use of intrinsic complexity metrics for automated ATM system.

B. *Aircraft 4D trajectory optimization*

Relevant to optimization problems in TBO environment, most previous researches aims to resolve conflicts between aircraft trajectories. Deterministic and metaheuristic algorithms have been widely developped to solve these problems. Durand et al. [8] proposed two trajectory separation methods by modifying headings and flight levels. En-route conflicts between trajectories are solved by genetic algorithms (GA). Dougui et al. [9] suggested a Light Propagation Algorithm (LPA) based on various light refractions. Using a Branch-and-Bound (B&B) algorithm, certain potential conflicts are solved. Chaimatanan et al. [10] proposed a strategic trajectory planning methodology to minimize 4D flight interactions. A hybrid simulated annealing was proposed to generate interaction-free trajectories in European airspace.

Breil et al. [11] applied the convergence indicator to measure airspace complexity and built a temporary route network for reducing the traffic complexity at the tactical level. Juntama et al. [12] proposed allocation of new departure times and flight levels to minimize traffic complexity based on König metric at strategic level. In this work, the distributed metaheuristic optimization is applied for the resolution approach but no learning mechanism incorporated in this framework.

In this paper, we has been used a mathematical formulation to structure aircraft trajectories. We focus on a new objective function which aims at structuring the traffic in the airspace by minimizing air traffic complexity. We apply the complexity metric based on linear dynamical systems [7] to measure the impact of air traffic structure in a TBO environment. We also propose a new adaptive metaheuristic approach using reinforcement learning for the resolution algorithm. The heuristic selection and reinforcement learning mechanisms applied in this approach, are based on the previous work in [13].

III. Mathematical Model

The problem in this work is to determine the optimized 4D trajectories where aircraft can fly in the airspace with less impact of traffic structure. In this section, we start by reformulating the optimization problem. This problem enables the two following opportunities to structure the aircraft trajectories: alternative departure time and en-route trajectory subject to the limited departure time shift and en-route extension length constraints. Finally, air traffic complexity metric based on linear dynamical system is introduced for the objective function.

A. *Decision variables*

The problem instance is given by:

- $\mathcal{X}$: Set of initial sampled 4D trajectories,
- N: Number of aircraft,
- t_s: Trajectory sampling time,
- M: Maximum number of waypoints of each flight,
- δ_a: Maximum allowed advance departure time,
- δ_d: Maximum allowed delay departure time,
- d_i: The maximum allowed route length extension coefficient of each flight i, $0 \leq d_i \leq 1$
- L_i: Length of the initial en-route segment for each flight

1) Alternative departure time: The first option is to advance or delay time of departure with departure time shifts δ_i of each flight i. As given in initial flightplans, each flight i has an initial departure time t_i. If the flight is selected to perform this option, the new departure time will be expressed as follows:

$$\hat{t}_i = t_i + \delta_i \tag{1}$$

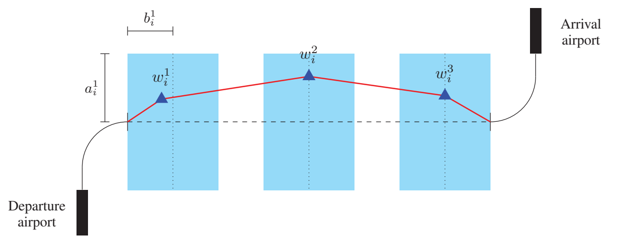


Figure 1. Initial (dashed line) and alternative (red line) trajectory with 3 additional waypoints

2) Alternative en-route trajectory: The second option is to reconstruct an alternative trajectory by setting a set of waypoints defined by

$$w_i = \{w_i^m | w_i^m = (w_{ix'}^m, w_{iy'}^m)\}_{m=1}^M \tag{2}$$

The set of waypoints allows us to reshape the initial trajectory as depicted in Fig. 1. More details regarding with this method to modify the trajectory are presented in [10].

B. *Problem constraints*

Throughout this problem, the constraints will be defined as follows:

1) Maximum allowed departure time shifts: We can manage the overall flight delay by giving the maximum allowed departure time shifts to the solution. The departure time shifts δ_i has the unit of slot. The range of departure time slots can be expressed as follows:

$$\delta_i \in \{-\delta_a, -(\delta_a - 1), \cdots, 0, \cdots, \delta_d - 1, \delta_d\} \tag{3}$$

2) Limited route length extension: To limit the route length extension, the alternative en-route profile of flight i shall satisfy:

$$L_i(w_i) \leq (1 + d_i) \cdot L_i \tag{4}$$

where $L_i(w_i)$ is the new length of the alternative en-route profile determined by the set of waypoints w_i.

C. Objective function

This subsection introduces the modeling and analysis of the air traffic complexity metric and then clarifies the approach to develop the objective function with such metric.

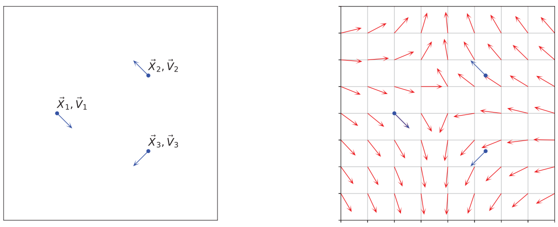
The approach is to model at each instant of time a set of trajectories using a linear dynamic system with the following general equation:

$$\dot{\vec{X}} = \mathbf{A} \cdot \vec{X} + \vec{B} \tag{5}$$

where $\vec{X}$ represents the state vector of the system.

$$\vec{X} = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{6}$$

This equation associates a speed vector $\dot{\vec{X}}$ with each point in the state space $\vec{X}$. The vector $\vec{B}$ represents the static behavior of the system. The matrix $\mathbf{A}$ is the linear relation between the speed vector $\dot{\vec{X}}$. The eigenvalues of the matrix $\mathbf{A}$ represent the evolution of the system. Hence, we use these eigenvalues to determine the complexity metric.



(a) Observation vectors (b) Vector field

Figure 2. Measurement or observation vectors and vector fields derived by the linear dynamical model

Let N be the number of aircraft presented in a sector at a given time. For each aircraft, we consider the two observation vectors (see Fig. 2a): a position measurement:

$$\vec{X_i} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \tag{7}$$

and a speed measurement:

$$\vec{V_i} = \begin{bmatrix} vx_i & vy_i & vz_i \end{bmatrix}^T \tag{8}$$

To obtain the matrix $\mathbf{A}$, we determine the dynamic model which is best fitted to observations. As shown in Fig. 2b, the dynamic model can be illustrated by the vector field in the airspace. The vector field is derived from linear equation ($\dot{\vec{X}} = A.\vec{X} + \vec{B}$) which is best fitted to the given observations.

Our problem is to find the matrix $\mathbf{A}$ and vector $\vec{B}$, which can minimize the error between observations and the dynamical model. This minimization problem can be expressed as follows:

$$\min_{\mathbf{A},\vec{B}} \sqrt{\sum_{i=1}^{i=N} \left\| \vec{V_i} - \left(A.\vec{X_i} + \vec{B} \right) \right\|^2} \tag{9}$$

To obtain the matrix $\mathbf{A}$ and $\vec{B}$ from this minimization problem, the Least Mean Square minimization (LMS) method can be used as detailed in [7].

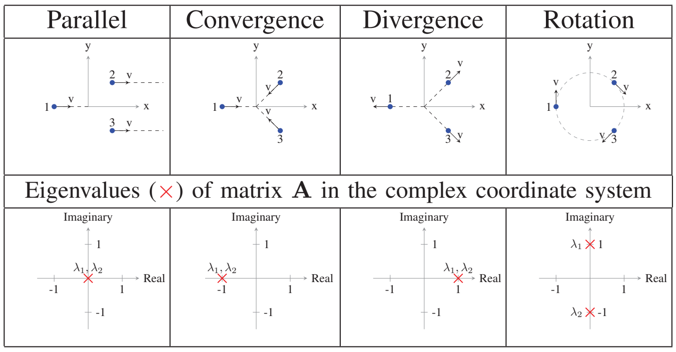
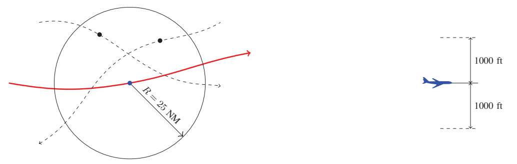


Figure 3. Eigenvalue locations for four different traffic situations

Fig. 3 shows the locations of the eigenvalues of matrix $\mathbf{A}$ for four outstanding traffic organizations: parallel, convergence, divergence and rotation. With the first situation, the eigenvalues are null because the aircraft are flying in parallel, representing a translation: distances between aircraft remain unchanged with time. In contrast with the second situation, the eigenvalues are real negative; the system evolves in a contraction mode and the four aircraft are converging: the norms of the relative distances between aircraft decrease with time. The third situation represents an expansion evolution whose eigenvalues are real positive and the aircraft are diverging: the relative distances increase with time. The rotation situation is associated with full imaginary eigenvalues because the aircraft stay at the same distance for all time from each other in a curl moving.

In this paper, we use complexity metric to measure the traffic structure. To obtain the complexity metric Ψ_{ik} for the aircraft i at time k, the process begins with identifying the traffic situation around an aircraft i at time k with the horizontal and vertical search spaces as represented in Fig. 4. The horizontal search space is centered around the *reference aircraft*. In RVSM (Reduced Vertical Separation Minima) airspace, en-route aircraft vertical separation is 1000 ft. Therefore, the vertical search space is created to find the neighboring aircraft whose altitudes are in the risk of loss separation to the reference aircraft. The observation vectors of reference and neighboring aircraft found within the horizontal and vertical search spaces are included in determining the complexity metric.



(a) Horizontal search space (b) Vertical search space

Figure 4. Search space for structure validation

The extension of the time uncertainty, which can be caused by passenger delay, airport and network operations, is taken into account. We apply the maximum time error t_ϵ to all aircraft within the search space. This error is the predicted arrival time of aircraft at a given position under uncertainty lying in the interval t_ϵ where $t - t_\epsilon/2 \leq t \leq t + t_\epsilon/2$ and t is the actual arrival time of the aircraft. Fig. 5 shows time uncertainty to all aircraft within the search space. All observation vectors presented in this time interval will be considered for the metric.

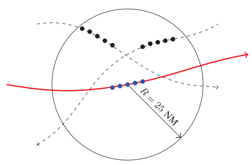


Figure 5. Horizontal search with extension of time uncertainty

After determining the matrix $\mathbf{A}$ from eq. (9), we can extract complex eigenvalues from this matrix. For aircraft i at time k, we have $\Lambda_{ik} = \{\lambda_1^{(ik)}, \lambda_2^{(ik)}, \ldots, \lambda_L^{(ik)}\}$ the set of complex eigenvalues from the matrix $\mathbf{A}$ where $\lambda_l^{(ik)} = a_l^{(ik)} + j b_l^{(ik)}$, we can produce the complexity metric as follows:

$$\Psi_{ik} = \sum_{l \in L} |a_l^{(ik)}|, \quad \text{for } L = \{l | a_l^{(ik)} < 0\} \tag{10}$$

Then, we calculate the total complexity Ψ_i of the aircraft i from the following equation:

$$\Psi_i = \sum_{k=1}^{N_i} \Psi_{ik} \tag{11}$$

N_i is a number of 4D coordinates of aircraft i.

Therefore, the aggregated complexity Ψ determined from all aircraft in airspace can be determined as follows:

$$\Psi = \sum_{i=1}^{N} \Psi_i = \sum_{i=1}^{N} \sum_{k=1}^{N_i} \Psi_{ik} \tag{12}$$

where N is a number of aircraft.

IV. Resolution Algorithm

The resolution algorithm relies on a hyper-heuristic approach. A hyper-heuristic, was first generalized in 2003, is an automated methodology for selecting or generating heuristics to solve hard computational search problems [14]. In this paper, we propose the adaptive metaheuristic using reinforcement learning (AMRL) for air traffic structuration problem. This algorithm enhances the traditional metaheuristic optimization with the incorporation of reinforcement learning, heuristic selection method and several low-level heuristics. This section introduces the AMRL algorithm and its three major components: low-level heuristics, heuristic selection and reinforcement learning.

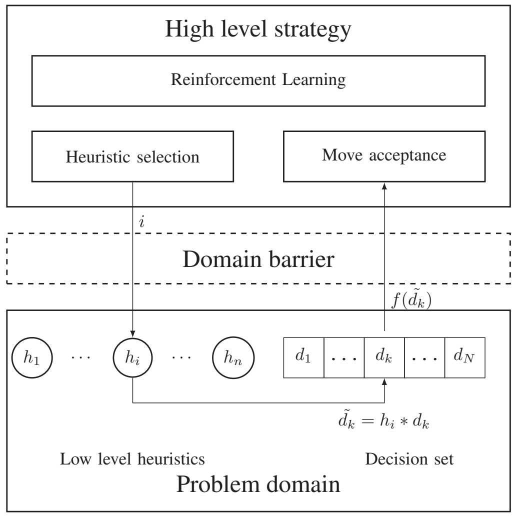


Figure 6. Adaptive metaheuristic framework with three major components: (i) low-level heuristics, (ii) heuristic selection, (iii) reinforcement learning and (iv) move acceptance and (iv)

A. *Adaptive metaheuristic using reinforcement learning*

The AMRL framework is represented in Fig. 6. Suppose that the current decision is d_k in the system, the system performs the *heuristic selection* method for selecting the heuristic operator h_i from one of *low-level heuristics*. The system applies such selected operator h_i to generate a new decision $\tilde{d}_k$. The system then performs the *move acceptance method* to accept or reject the new decision $\tilde{d}_k$. The *reinforcement learning* observes the current state, the selected operator and the evolution of decisions to determine rewards. Values in the state-action matrix will be updated with rewards to favor the selection of the heuristic operators in the same states.

B. *Low-level heuristics*

The specialization of AMRL for a particular optimization problem requires the definition of diversification and intensification operators. In this paper, generation and mutation operators perform the diversification task. Some local descent heuristics are used for intensification. To solve the traffic structuration problem, four diversification operators and four intensification operators are proposed.

1) Intensification operators: The four intensification operators are based on different neighborhood structures: *5-advance*, *5-delay*, *1-move*, *2-move*. These neighborhood structures are used in a local descent procedure that consists in performing a sequence of moves toward a local optimum decision.

- *5-advance* (h_1): a local search which consists in randomly advancing departure time not more than 5 minutes.
- *5-delay* (h_2): a local search which consists in randomly delaying departure time not more than 5 minutes.
- *1-move* (h_3): a local search which finds a neighborhood decision by moving a single waypoint within a given bound.
- *2-move* (h_4): a local search which finds a neighborhood decision by moving one or more waypoints within a given bound.

2) Diversification operators: The set of diversification operators is composed of two generation and two mutation procedures. These operators are used for the diversification process during optimization.

- *20-shift* (h_5): a generation operator which randomly advance or delay departure time 20 minutes from the current time.
- *new-route* (h_6): a generation operator which randomly create one or more waypoints under problem constraints.
- *flip-opposite* (h_7): a mutation operator which consists in vertically flipping current waypoints opposite to the direct route segment in a symmetrical manner.
- *remove-or-insert* (h_8): a mutation operator which consists in randomly removing or inserting a single waypoint in such a way that problem constraints are satisfied.

C. Heuristic selection

As illustrated from Table I, we construct the transition matrix in form of (state, action) where the actions corresponds to intensification and diversification operators. To select the heuristic operator, we perform the roulette-wheel selection principle from each weight associated with each operator for the current state. At initial state, each weight value w_{ij} in the transition matrix is initialized with either μ or 0. The modification of each weight value is performed by reinforcement learning.

TABLE I. INITIALIZATION OF WEIGHT VALUES IN STATE-ACTION MATRIX

	Intensification operators				Diversification operators			
State	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
s_0	μ	μ	μ	μ				
s_1	0	μ	μ	μ				
s_2	μ	0	μ	μ				
s_3	μ	μ	0	μ				
s_4	μ	μ	μ	0				
s_5					μ	μ	μ	μ

The collection of states allows us to perform the Diversification-Intensification cycle (D-I cycle). The cycle starts from selecting a generation operator among the set of diversification operators. Each state is determined based on the type of operator previously applied. Since there are four intensification operators applied in the traffic structuration problem, six states will be represented as follows:

- s_0: a diversification operator has been previously applied.
- s_1: intensification operator h_1 has been previously applied.
- s_2: intensification operator h_2 has been previously applied.
- s_3: intensification operator h_3 has been previously applied.
- s_4: intensification operator h_4 has been previously applied.
- s_5: two successive intensification operators have been previously applied without modifying the current decision.

D. Reinforcement learning (RL)

The reinforcement learning approach [15] selects actions that maximize expected rewards generated by decisions. The agent learns how to choose actions through trial-and-error interactions with a dynamic environment to observe the signals or rewards returned from previous states. The agent may take a long sequence of actions to have a delayed rewards, receiving insignificant reinforcement, then finally arrive at a state with high reinforcement.

At each state-to-state transition in AMRL, the system stores an experience containing the current state s with the selected operator h and the gain g which is the different cost between the new decision and the previous one. Until the end of D-I cycle, RL learns the collection of experiences and then determines the reward σ to update the weight values in the state-action matrix. Only the weight values whose state-operator pairs can improve the decision, will be updated with this formula:

$$w_{i,j} := w_{i,j} + \sigma \tag{13}$$

where $w_{i,j}$ is the weight value of the event whose action is taken by selecting the operator h_i due to the state s_j.

E. Moving acceptance

Move acceptance decides whether to accept or reject a new decision at each step during the search process. The iterations continue until a termination criterion is met.

In our framework, we apply the *metropolis* based criterion to perform our move acceptance mechanism. In case of minimization, the newly produced decision is accepted if it is better and if it is worse, it will be accepted with a probability $\exp\left(\frac{f(d_i) - f(\tilde{d}_i)}{T}\right)$, where $f(d_i)$ and $f(\tilde{d}_i)$ denotes the cost value of the current decision and the newly generated decision respectively, and T denotes the temperature. When T is high/low, the chance of accepting a deteriorated move is high/low.

The AMRL algorithm for traffic structuration problem is detailed in Algorithm 1. Based on the metropolis mechanism, the algorithm is composed of data pre-processing, heating loop and cooling loop. The algorithm starts with the start of the data structures. Then, the heating process is started and the output from this process provides us with the initial temperature T_0 that is required to start the cooling loop. The initial temperature is set with the initial acceptance rate $\tau = 0.8$. The cooling loop of the algorithm is executed as long as the stop criterion is not reached. During an iteration of the cooling loop, the algorithm selects an operator, apply it and update the decision. The end of each transition corresponds to the tasks assigned to the RL. RL action and sharing the transition matrix only take place at the end of a D-I cycle. If some new decisions are improved from their current decisions, The system will reinforce the related state-operator pairs in this D-I cycle by using eq. (13).

Algorithm 1 Adaptive Metaheuristic with RL (AMRL)

$T_0 \leftarrow$ heat_up(τ)
$D \leftarrow$ init_decisions()
$H \leftarrow$ init_heuristic_operators()
$W \leftarrow$ init_transition_matrix()
$E \leftarrow$ init_experiences()
$T := T_0$
repeat ▷ Start cooling loop
  for $i = 1 \rightarrow N$ **do**
    if $f(d_i) > \gamma_T$ **then**
      $s \leftarrow$ compute_state(E)
      $h \leftarrow$ select_operator(W, s)
      $\tilde{d}_i \leftarrow$ apply_operator(h, d_i)
      $g \leftarrow f(d_i) - f(\tilde{d}_i)$
      update_history(E, s, h, g)
      metropolis_acceptance($\tilde{d}_i, d_i$)
      if end of D-I cycle **then**
        individual_learning(W, E)
      end if
    end if
  end for
  $T \leftarrow \alpha \cdot T$
until $(T = T_f) \vee (\Psi = 0)$
return $\widetilde{D}$

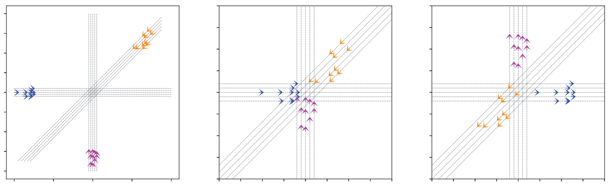
V. Results

By assessing the performance of our model and algorithm, we propose five different scenarios for our experiment. The first three scenarios are related to short-term traffic structuration. The latter two scenarios are the strategic trajectory planning application, which takes place six months to seven days before the day of operation. We perform this experiment with Java on Ubuntu system with Quad-core 2.7 GHz processor and 16 GB memory.

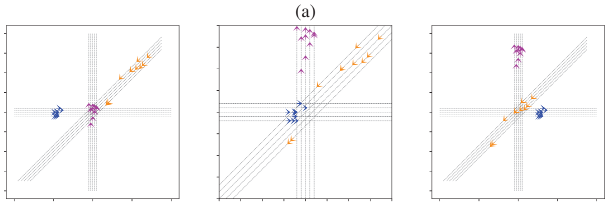
A. Flow crossing

The first scenario is to control an interaction of three traffic flows. Each parallel ten aircraft from W-E, S-N and NE-SW are flying with the same speed and altitude to the same area as depicted in Fig. 7a. Each aircraft enter the sector at different times. As shown in Fig. 7c, the proposed complexity metric can identify the level of disorder of the initial trajectories from $t_s = 100$ to $t_s = 150$. Aggregated complexity for this initial traffic is $\Psi = 50.48$.

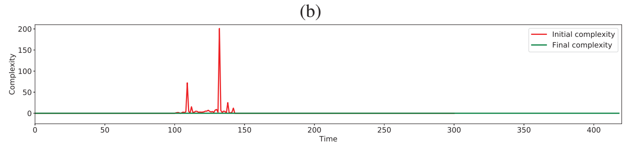
We customize the AMRL algorithm to restructure the traffic in the first scenario by rescheduling the flight time of each aircraft before arriving in the sector. The heuristic operators *5-delay* and *20-shift* are activated to generate the neighborhood decision for this scenario. After running our algorithm with 717 iterations, we can mitigate the risk of collision in this traffic situation. As represented in Fig. 7b, the final aggregated complexity is $\Psi = 0.118$. Each flight time is adjusted before entering the sector to avoid complexity.



(a)

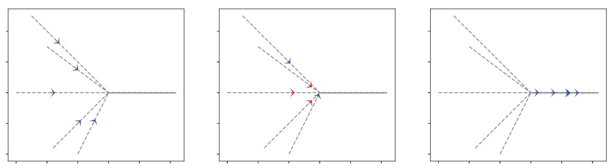


(b)

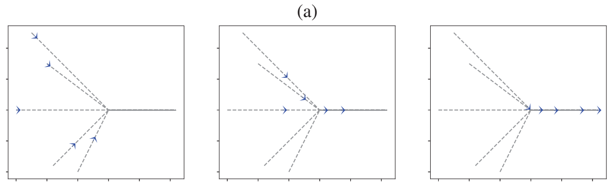


(c)

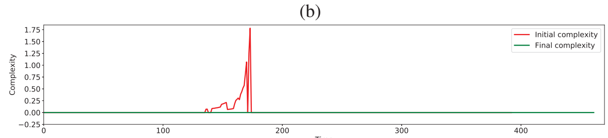
Figure 7. Traffic situation of (a) initial traffic and (b) restructured traffic and (c) Initial and final complexity metrics as a function of time during the operation of flow crossing



(a)



(b)



(c)

Figure 8. Traffic situation of (a) initial traffic and (b) restructured traffic and (c) Initial and final complexity metrics as a function of time during the operation of time based metering

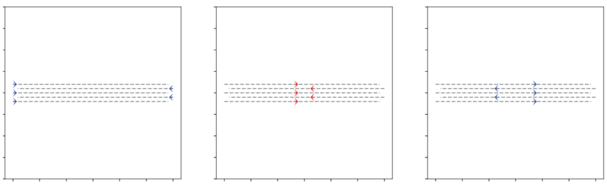
B. Time based metering traffic

The second experiment is related to the management of two traffic flows flying to the same crossing point. *Time-based* *metering* [16] is a method used by controllers to reduce air traffic complexity and to increase predictability to ATC system. Flights are delivered to a specific point at a specific time. This manner of structuring proposes a new short-term schedule of aircraft entering a sector. The new schedule will be easier to perceive and to manage by controllers. In this scenario, five aircraft are flying to the same metering point with the same speed and altitude but different time for entering the sector. These aircraft will fly on the same path after crossing over the metering point. For initial trajectories in Fig. 8a, the complexity tends to be higher than usual when three aircraft converge to the metering point at approximately the same time. The aggregated complexity of these initial trajectories is $\Psi = 10.45$.

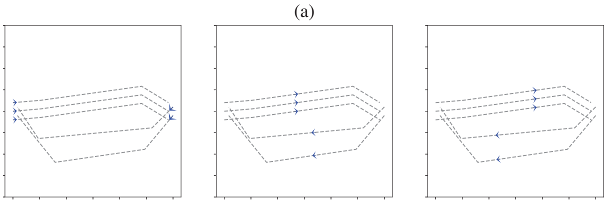
Likely in the first scenario, we reschedule the flight time of each aircraft before arriving in the sector. The resolution algorithm activates the heuristic operators used in previous scenario. Finally, we can mitigate the congested traffic as depicted in Fig. 8b. The final complexity is $\Psi = 2.63 \cdot 10^{-2}$ after 32 iterations. Each flight time is adjusted for entering the sector to avoid complexity.

C. *Traffic encounters*

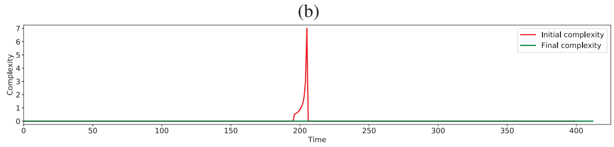
The third scenario is an attempt to manage two flows of traffic encounters with the same speed and attitude (see Fig. 9a). To manage this situation, the controllers should create a temporary route for each aircraft. As depicted in Fig. 9c, high complexity values are identified in theăarea where aircraft are in the risk of collision. The aggregated complexity for these initial trajectories is $\Psi = 18.28$.



(a)



(b)



(c)

Figure 9. Traffic situation of (a) initial traffic, (b) restructured traffic and (c) Initial and final complexity metrics as a function of time during the operation of collision avoidance

We propose the customized AMRL algorithm to restructure the traffic. To propose the temporary route to each aircraft, we apply space-based heuristic operators: *1-move*, *2-move*, *new-route*, *flip-opposite*, *remove-or-insert*. As depicted in Fig. 9b, the AMRL algorithm proposes parallel route structure to minimize complexity in the sector. After running our algorithm with 576 iterations, the aggregated complexity for final trajectories is $\Psi = 1.21 \cdot 10^{-2}$.

D. *Real traffic without uncertainty*

The fourth scenario represents a full day traffic in French airspace with 8836 aircraft trajectories as depicted in Fig. 10. The parameters defined for the optimization problem are given in Table II. All heuristic operators proposed in this paper are applied for this problem. Before starting, initial trajectories represent an overall complexity of $\Psi = 188456$ and the complexity map for these trajectories is represented in Fig. 11a. The final complexity and the computation time to solve the problem are reported in Table III. Restructuring the traffic can improve the overall complexity by 49.6 %, compared to the initial trajectories. The complexity map related to final trajectories is shown in Fig. 11b.

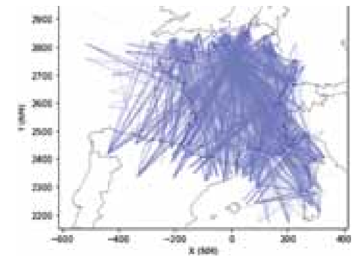
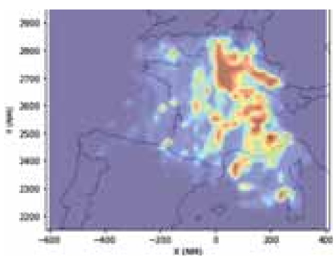


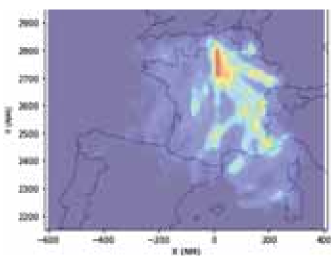
Figure 10. Initial trajectories of a full-day traffic in French airspace

TABLE II. USER PARAMETERS FOR REAL TRAFFIC SCENARIO

Parameters	**Value**
Starting temperature control parameter χ	0.8
Complexity threshold value γ_T	$0.95 \cdot \gamma_{\max}$
Geometric cooling rate α	0.99
Initial weight values of state-operator μ	1.0
Reward value σ	1.0
Maximum changes of departure time δ_a, δ_d	60 minutes
Maximum number of waypoints M	3
Maximum route length extension L	0.15 (15 %)



(a)



(b)

Figure 11. Complexity map of (a) initial trajectories and (b) final trajectories of a full-day traffic in French airspace

TABLE III. NUMERICAL RESULTS FOR RESTRUCTURING A FULL DAY TRAFFIC IN FRENCH AIRSPACE (10 RUNS FOR AVERAGE COMPUTATION)

Numerical result	**Value**
number of iterations	3295
total complexity	94994.1
avg. complexity	10.80
avg. departure time changes	4.35 minutes
avg. route length extensions	0.12 %
avg. computation time	3.72 minutes

E. Real traffic with uncertainty

The proposed robust traffic structuration methodology is tested by accounting time uncertainty. The structure validation model related to this scenario is explained in Section III. The traffic used in this experiment is similar to the previous scenario. The parameters of the AMRL algorithm are the same as those proposed in Table II. The simulation is performed considering time uncertainty interval with t_ϵ = 1, 2 and 3 minutes. The initial and final complexity between trajectories are reported in Table IV.

TABLE IV. NUMERICAL RESULTS FOR RESTRUCTURING A FULL DAY OF TRAFFIC CONSIDERING TIME UNCERTAINTIES OF 1, 2 AND 3 MINUTES

t_ϵ (minutes)	initial Ψ	final Ψ	computation time (minutes)	no. of iterations
1	$1.00 \cdot 10^6$	59 740.5	25.8	3446
2	$7.38 \cdot 10^6$	56 411.4	72.1	3783
3	$9.80 \cdot 10^6$	55 666.3	117.8	4225

The proposed traffic structuration can mitigate the complexity in air traffic by 94.0 %, 99.2 % and 99.4 % for given time uncertainties of t_ϵ = 1, 2 and 3 minutes respectively. The computation time increases when we consider a higher level of time uncertainty.

VI. CONCLUSION

In this paper, we introduced a methodology to address the traffic structuration problem in the framework of trajectory based operation. The structure validation model using air traffic complexity metric based on linear dynamical system can quantify the impact of air traffic complexity. To minimize this impact, we can restructure air traffic patterns by modifying the departure time and route structure for each trajectory. To solve this problem, an adaptive metaheuristic using reinforcement learning was proposed. Experimental results from five scenarios represent the potential of the proposed methods. The AMRL can restructure traffic in both time dimension (for flow crossing and Time-based metering traffic) and space dimension (for traffic encounters) with lessened air traffic complexity metric. Especially in traffic encounter situation, the AMRL can reorganize traffic into parallel flows, which are easier for controllers to perceive and manage this situation. A national scale dataset with 8836 trajectories is used as another scenario. Our algorithm can minimize the impact of traffic structure with less computation time (3-4 minutes for 8836 trajectories without uncertainty). In order to increase robustness of trajectories, the time uncertainty of aircraft was also considered. Future work can perform other structuration approaches such as allocation of flight level and speed regulation. In addition to enhance the performance of AMRL, new heuristic operators can be developed with the aim of providing better structuration in air traffic.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore, under the Aviation Transformation Programme. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and the Civil Aviation Authority of Singapore.

REFERENCES

[1] "Annual Network Operation Report 2019," EUROCONTROL, Main Report, Mar. 2019.
[2] B. Hilburn, "Cognitive Complexity in Air Traffic Control - A Literature Review," EEC, Tech. Rep. 04/04, Jan. 2014.
[3] I. V. S. Laudeman, "Dynamic Density: An Air Traffic Management Metric," Tech. Rep. NASA-TM-1998-112226, Apr. 1998.
[4] B. Sridhar, "Airspace Complexity and its Application in Air Traffic Management," Orlando, FL, United States, Jan. 1998.
[5] S. Mondoloni and D. Liang, "Airspace Fractal Dimensions and Applications," Dec. 2001.
[6] K. Lee, E. Feron, and A. Pritchett, "Air Traffic Complexity: An Input-Output Approach," in *2007 American Control Conference*. IEEE, Jul. 2007.
[7] D. Delahaye, P. Paimblanc, S. Puechmorel, J. Histon, and R. Hansman, "A new air traffic complexity metric based on dynamical system modelization," in *Proceedings. The 21st Digital Avionics Systems Conference*, vol. 1, Irvine, CA, USA, 2002, pp. 4A2–1–4A2–12.
[8] N. Durand, J.-M. Alliot, and J. Noailles, "Automatic aircraft conflict resolution using genetic algorithms," in *Proceedings of the 1996 ACM symposium on Applied Computing - SAC '96*. Philadelphia, Pennsylvania, United States: ACM Press, 1996, pp. 289–298.
[9] N. E. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau, "A new method for generating optimal conflict free 4d trajectory," in *ICRAT 2010, 4th International Conference on Research in Air Transportation*, Budapest, Jun. 2010, pp. 185–191.
[10] S. Chaimatanan, D. Delahaye, and M. Mongeau, "A hybrid metaheuristic optimization algorithm for strategic planning of 4D aircraft trajectories at the continental scale," *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 46–61, Nov. 2014.
[11] R. Breil, D. Delahaye, L. Lapasset, and E. Féron, "Multi-agent Systems to Help Managing Air Traffic Structure," *Journal of Aerospace Operations (ISSN : 2211-0038)*, Sep. 2017.
[12] P. Juntama, S. Chaimatanan, S. Alam, and D. Delahaye, "A Distributed Metaheuristic Approach for Complexity Reduction in Air Traffic for Strategic 4D Trajectory Optimization," in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, Singapore, Singapore, Feb. 2020, pp. 1–9.
[13] D. Meignan, A. Koukam, and J.-C. Créput, "Coalition-based metaheuristic: a self-adaptive metaheuristic using reinforcement learning and mimetism," *Journal of Heuristics*, Dec. 2009.
[14] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg, *Hyper-Heuristics: An Emerging Direction in Modern Search Technology*. Springer, 2003, pp. 457–474.
[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, second edition ed., ser. Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2018.
[16] R. Bolczak and K. M. Levin, "Time based metering as a component of performance-based air traffic management," in *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*, 2007.