ENRI Int. Workshop on ATM/CNS. Tokyo, Japan (EIWAC2019)

[EN-A-37] Coordinated Sequencing of Traffic on Multiple En-route Constraint Points

+ S. Abba Rapaya * P. Notry ** D. Delahaye **

* OPTIM, ENAC-LAB, ENAC
French Civil Aviation University
7 Avenue Edouard Belin, 31055, Toulouse, France

* souleymanr@gmail.com

** philippe.notry@enac.fr—delahaye@recherche.enac.fr

Abstract: Air transportation traffic is progressively increasing over the years and dealing with it is an essential task to guarantee fluid flights in the future. Several works already indexed multiple aspects of aviation, among them, the E-MAN system. It introduced the sequencing of arriving traffic, starting from early stages of the En-route phase. This change facilitated the work for the approach controllers but also increased the workload of the En-route controllers. To handle that workload, controllers are now assisted by tools that consider the new constraints introduced by the arrival management system and propose advisories. From that same perspective, our project focuses on an algorithm for a helper tool that will combine both aspects of traffic sequencing in the En-route phase and conflict resolution. With this novel approach, we automatically generate near-to- optimal flight decisions, given that we can modify the speed and the flight level to respect the sequencing constraints and cut down potential conflicts. We categorize the problem as a mathematical optimization case. Thus, we describe a detailed mathematical model which covers all the aspects of the problem. This model gives a basis for the implementation of the flight optimizer. Later, we propose a solution based on a sliding window simulated annealing algorithm which reduces the complexity and takes into account uncertainties. Finally, we successfully test an implementation of the solution with real-life traffic data. It corresponds to flights within France going towards Paris CDG airport over a period of 24 hours. The results demonstrate a total conflicts resolution with satisfying compliance with sequencing constraints.

Keywords: Air Traffic Management, Traffic sequencing, Arrival Management, Conflict resolution, Multi-objective optimization, Simulated annealing, Sliding window

1 Introduction

The air traffic has been increasing lately. In its Global Market Forecast (GMF) [1], Airbus has predicted that the global air traffic will encounter an annual growth of 4.4% for the next coming 20 years. With that increase of traffic, some major airports will now rely on arrival management systems such as the XMAN system to help them manage their flow of traffic and generate efficient flights sequencing. However, most of the workload is now transferred to the En-route controllers since they will have to deal with occurring conflicts and also try to respect new constraints imposed by the XMAN to flights.In En-route, the airspace is characterized by several route flows crossing into potential conflict points. To avoid those conflicts, some separation constraints are defined for the flights and controllers need to make sure they are respected by using classic methods like vectoring, speed controlling or flight level changes. Each decision applied to an aircraft can have an impact on surrounding flights.This research aims to develop an algorithm that will assist the controllers. Based on the traffic flows analysis, this algorithm will provide near optimum flight decisions in order to remove conflicts at the crossing points.From the controller's side, the main objective is to reduce the workload and minimize the frequency occupation. This will be done by automatically eliminating conflicts due to lack of separation at crossing points and route links. It also implies reducing the number of flight level changes and simplifying the instructions given to pilots.Another objective is to integrate our tool with an XMAN system by generating decisions compatible with XMAN constraints.Finally, from the airlines point of view, we will need to take into consideration the preferred flight profile by generating decisions not far from that profile.The first part of the paper introduces some previous related works linked to arrival manager optimization in a wide sense. The second part presents the associated mathematical model. The meta-heuristic (simulated annealing) used for solving the underlying optimization problem is presented in the third part. Finally, the fourth part presents the results given by the algorithm on a real case at Paris CDG.

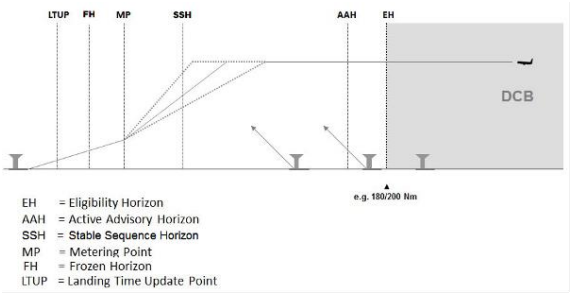

Figure 1 Simple Horizon Extension

2 State of the art

This section presents the concepts and previous works related to our problem. It focuses on points around air traffic sequencing, conflict resolution and mathematical optimization methods. It is concluded by a synthesis which serves as baselines for our modeling and solution approach.

2.1 Extended AMAN

The Cross-border SESAR Trials for Enhanced Arrival Management (X-STREAM) project main objective is to extend the current horizon of the Arrival Management System beyond 200 NM towards the upstream Area Control Center (ACC). Extended-AMAN (E-AMAN) [6] allows for the sequencing of arrival traffic much earlier than is currently the case, by extending the AMAN horizon from the airspace close to the airport to further upstream and so allowing more smooth traffic management. Controllers in the upstream sectors, which may be in a different control center or even a different Functional Airspace Block (FAB), obtain system advisories to support an earlier pre-sequencing of aircraft. Controllers implement those advisories by, for example, instructing pilots to adjust the aircraft speed along the descent or even before top-of-descent, thus reducing the need for holding and decreasing fuel consumption.

2.1.1 Characteristics

The traffic begins to be processed at the Eligibility Horizon (EH) and controllers are provided with AMAN advisories from the Active Advisory Horizon (AAH) onward [5]. With the extension of the horizon of the AMAN to the En-route phase, as illustrated on Figure 1, the EH goes further to around 180-200NM which supports the controllers in applying more efficient arrival management techniques. The typical optimum top of descent is approximately 100-120NM from touchdown which implies that up to 100NM of flight within the extended AAH will be in En-route airspace.

2.1.2 Decisions on flights

AMAN advice may be in the form of a Target Time at the Initial Metering Point (IMP) or Time To Lose (TTL)/Time To Gain (TTG) advisories to controllers calculated by AMAN working back from the runway time [5]. Upstream ACC controllers can then provide instructions to pilots to make adjustment of speed to meet TTL/TTG need, by executing advisories for speed coming from tools like Speed And Route Advisory (SARA) or to delegate responsibility for adherence to a Controlled Time of Arrival (CTA) to flight crew.

2.2 Previous works

Optimization problems involve the minimization (or maximization) of a function by choosing input values from a given set of data and computing the value of the function. Nowadays, many studies have been conducted on the resolution of air traffic conflicts and the sequencing using various mathematical optimization techniques. This section will present a few of those works. [10] presented a new approach based on concepts of speed control and flight-level assignment for conflict resolution over predefined routes. It was based on a Mixed Integer Linear Program (MILP). The way it was formulated helped to reduce fuel burn over time horizons between 15-45 minutes. [7] worked on a flight scheduler for En-route air traffic. It was based on the application of delays or route changes to flights to produce an ordered merged sequence of flights at the exit nodes. They didn't tackle the conflict resolution aspect in that paper. During the 2017 SESAR Innovation Days, [3] presented a method to perform small changes on aircraft speed in order to resolve conflicts. In their model, they considered that the True Air Speed (TAS) was constant for each aircraft in the airspace and the uncertainties appeared due to wind components, with wind data collected from MétéoFrance PEARP (Prévision d'Ensemble ARPège). A protection area, as illustrated in Figure 2, is defined around each aircraft to represent its potential positions. Thus, two overlapping protection areas implies a potential conflict between the two aircraft that must be dealt with. The solution was implemented in Python using a simulated annealing algorithm. As a result, the algorithm was able to solve at least 70% of virtual conflicts with a computing time of 30 minutes to 2 hours for a 4 hours traffic data depending on the refining of the parameters. On the other hand, [2] presented their work at the 29th Congress of the International Council of the Aeronautical Sciences (ICAS). It was done on a study for a collaborative tool that aimed at helping controllers with the integration of the Extended Arrival Manager. They studied the aircraft sequencing problem as an optimization problem under different resolution methods (Linear programming, non-linear programming, heuristic methods).

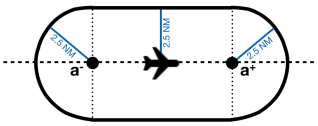

Figure 2 Protected area around the aircraft. If two areas overlap, there is a potential conflict

The objective function took into consideration the runway capacity, the delays and the fuel consumption. They also didn't consider the conflict resolution aspect.

2.3 Synthesis

As seen in this section, the Extended Arrival Management tool gives advisory directives to controllers to help create an efficient sequence of flights starting from a distant En-route point. However, it is still up to the controller to decide how he will implement those directives taking into account the potential occurring conflicts. A tool like SARA, in combination with XMAN, can helps controllers by giving speed directives but no global conflict resolution is provided. Thus, it will be an added value for our solution to provide both sequencing and conflict resolution capabilities. Moreover, most of the previous works done on the optimization of the traffic focused on changes made on aircraft speed or route to delay or advance a flight. Therefore, not many alternatives were given for conflict resolution when there is a dense traffic. In that case, we can introduce changes on flight levels to broaden the solutions possibilities as suggested by [3]. As for the optimization method to apply, the nature of our problem makes it more appropriate for a meta-heuristic method, given the dimension of the problem. Also, the simulated annealing appears as a good choice regarding its stability, its memory consumption profile and its speed. [8] introduced an implementation of the simulated annealing algorithm coupled with the sliding window concept, the base of our implementation will be that algorithm. Another added value we can integrate in our solution is the collaboration with multiple XMAN tools at the same time. The state of the art review enabled us to define basis for our study case. In the next section, we will define a detailed mathematical model for the problem. This will highlight abstract concepts useful for the design of our solution.

3 Mathematical model

This section focuses on an abstract mathematical modeling of our problem. Each aspect of the problem is represented using mathematical notations that can be interpreted into any optimization algorithm later. It presents the global network representation, how uncertainties are modeled, equations related to conflict detection, the optimization objective function definition and an evaluation of the complexity of the problem.

3.1 Constraints

We have identified some constraints for our model. They are elements that will influence the way the tool will handle the optimization process.

Sequencing constraints : those constraints are coming from the sequencing system. The XMAN generates a TTL or TTG for flights. It must be considered during the optimization process.

Aircraft performances : those correspond to the physical performances of the aircraft. It consists of the minimum (maximum) speed and the minimum (maximum) flight level the aircraft can fly. Basically, our tool should not advise a flight to perform outside of its physical limits.

Flight preferred profile : It is a constraint that can be proposed by the airline. In our case, we will consider the preferred cruising flight level. It is a flight level which if cleared to the flight, will help the airline to better achieve their business objective.

Separation constraints : we have three types of separation : the *distance separation* which is a minimum defined distance should be maintained between two aircraft when crossing a specific point, the *time separation* for which two consecutive aircraft should reach a given location with a minimum defined time interval and the *wake turbulence separation* which is the separation distance between two consecutive aircraft flying on the same link at the same flight level and which depends on their wake turbulence categorization as defined by ICAO.

3.2 Network modeling

The network of routes is modeled as an oriented graph with nodes and oriented links.

Nodes A node represents any point of interest on a route. Most of the time, it will be where two routes are crossing. We can define a separation constraint on them (distance or time separation). A node n is characterized by its cartesian coordinates (x, y), $x, y \in \mathbb{R}$, a minimum separation distance d_{sep}, in case of distance separation constraint and a minimum separation time t_{sep}, in case of time separation constraint.The set of nodes will be noted $\mathcal{N}$: $\mathcal{N} = \{n_i \ / \ i \in \mathbb{N},\ i \leq nb_{nodes}\}$ where nb_{nodes} is the total number of nodes in the network

Links A link is a portion of route between two nodes. It is oriented from an origin node to a destination node. Each link is characterized by an entry node n_{ori}, an exit node n_{dest} and a length $d_l = dist(n_{ori}, n_{dest})$. The set of links will be noted $\mathcal{L} : \mathcal{L} = \{l_i \ / \ i \in \mathbb{N},\ i \leq nb_{links}\}$. $\forall l_i \in \mathcal{L},\ l_i = (n_{ori}, n_{dest}) \ / \ n_{ori}$

,$n_{dest} \in \mathcal{N},\ n_{ori} \neq n_{dest}$ where nb_{links} is the total number of links in the network.

Routes A route is an ordered list of adjacent links and is characterized by a parity p which can be odd or even: $p \in \{ODD, EVEN\}$ and a list of available flight levels $fls = \{fl \ / \ fl \in \mathbb{N}\}$. We will use the semi-circular rule to determine those flight levels depending on the parity of the route. The set of routes will be noted $\mathcal{R}$: $\mathcal{R} = \{r_i \ / \ i \in \mathbb{N},\ i \leq nb_{routes}\}$, $\forall r \in \mathcal{R},\ r = \{l_i \ / \ l_i \in \mathcal{L}\}$ where nb_{routes} is the total number of routes in the network.

3.3 Flights modeling

A flight f is characterized by a speed V_f, a flight level FL_f, the time of arrival at the first node (entry in the airspace) $t_{f_{init}}$, a wake turbulence category $wtCat_f$ and a defined route $r_f \in \mathcal{R}$.Apart from those characteristics, a flight is also subject to some constraints: a maximum speed V_f^{max}, a maximum flight level FL_f^{max}, a preferred flight level FL_f^{pref} given by the airline and a time constraint ttl_f given by the sequencing system (a positive value means a time to gain and a negative value refers to a time to lose both before reaching the final exit node). The set of flights will be noted $\mathcal{F}$. $\mathcal{F} = \{f_i \ / \ i \in \mathbb{N},\ i \leq nb_{flights}\}$ where $nb_{flights}$ is the total number of flights in the network.

3.4 Decision variables

During the optimization process, two features of the flights are modified: the speed and the flight level. For that, we are using two decision variables which are the speed delta and the flight level delta.

Speed delta As expressed by [3] in their article, we will also use small speed changes to control flights speed. Each speed delta ΔV is an integer in the range of -6 to +3, expressed in percentage. $\forall f_i \in \mathcal{F},\ \Delta V_i \in \mathbb{Z} \cap [-6, +3]$.The new TAS V_i of flight f_i can be expressed as follow: $V_i = \frac{\Delta V_i}{100} \times V_{0_i}$ where V_{0_i} is the initial TAS.

Flight level delta The second decision variable we are using is the flight level delta. It represents the number of levels an aircraft should climb or descend. It is expressed as an integer varying from -2 to 2. A positive value represents a climb whereas a negative value represents a descend.

3.5 Uncertainties

In our assumptions, we supposed that the TAS of flights is constant all along its route. However, in the real world, each flight is subject to wind influence in the air which can affect the ground speed. This section presents how we model the wind and how it generates uncertainties in our model.

Wind modeling : we use a simplified model for the wind. The wind vector is characterized by a constant direction all over the airspace (it is represented as an angle α_w relative to the geographical north) and an intensity Ws randomly varying within a range of values, with $Ws \in [Ws_{min}, Ws_{max}]$.

Flight uncertainty : the ground speed Gs of the aircraft is the resultant from both the TAS component and the Wind Speed component ($\overrightarrow{Gs} = \overrightarrow{TAS} + \overrightarrow{Ws}$). Let's assume a flight arriving at the entry node n_{ori} of a link l at t_{ori}, its time interval of arrival at the exit node n_{dest} will be: $[t_{dest_{min}}, t_{dest_{max}}]$,= $[t_{ori} + \frac{d_l}{V_{f_{max}}}, t_{ori} + \frac{d_l}{V_{f_{min}}}]$. This interval is built with the earliest time of arrival at the exit node $t_{dest_{min}}$ which depends on $V_{f_{max}}$ which is the maximum ground speed of the aircraft and the latest time of arrival at the exit node $t_{dest_{max}}$ which depends on $V_{f_{min}}$ which is the minimum ground speed of the aircraft.

3.6 Conflicts identification

We will consider two types of conflicts: node conflicts and link conflicts.

Node conflicts They are detected using the separation constraints. Depending on the type of separation constraint at the node (time or distance separation), the equations to detect node conflicts will differ. To detect a distance separation conflict, a circular protection area with a diameter of d_{sep} is defined around the node. Each flight arriving at the node will be monitored using four different times : an early time of entry in the protection area t_{min}^{in}, a late time of entry in the protection area t_{max}^{in}, an early time of exit from the protection area t_{min}^{out} and a late time of exit from the protection area t_{max}^{out}. Two consecutive flights are considered to be in conflict at the node when they are both present within the protection area at the same time. In other words, the trailing flight enters the protection area before the leading flight exits it. Let's consider two flights f and g with f leading.

$$\forall f, g \in \mathcal{F},\ cflt(f,g) = \begin{cases} 1 & \text{if } t_{max}^{f,out} - t_{min}^{g,in} < 0 \\ 0 & \text{otherwise} \end{cases}$$

To detect a time separation conflict, a separation time t_{sep} is defined on the node. Each flight arriving at the node will be monitored using two different times: an early time of arrival at the node t_{min} and a late time of arrival at the node t_{max}.Two consecutive flights are considered to be in conflict at the node when the time interval between the two of them crossing the node is less than the separation time. Let's consider two flights f and g with f leading.

$$\forall f, g \in \mathcal{F},\ cflt(f,g) = \begin{cases} 1 & \text{if } |t_{max}^{f} - t_{min}^{g}| < t_{sep} \\ 0 & \text{otherwise} \end{cases}$$

Link conflicts Link conflicts are essentially detected using wake turbulence category separation. Three situations are analyzed to detect those conflicts: at the

entry node of the link, at the exit node of the link and catch-ups along the link itself. To detect entry conflicts, two different times are monitored at the entry of the link: an early time of arrival at the entry node t^{in}_{min} and a late time of arrival at the entry node t^{in}_{max}. Two consecutive flights are considered to be in conflict at the entry of a link when the distance between them is less than the required wake turbulence separation distance. Let's consider two flights f and g with f leading.

$$\forall f, g \in \mathcal{F},\ cflt(f,g) = \begin{cases} 1 & \text{if } dist_{ori}(f,g) < wTCat_{sep}(f,g) \\ 0 & \text{otherwise} \end{cases}$$

$$dist_{ori}(f,g) = V^{f}_{min} \times (t^{f,in}_{max} - t^{g,in}_{min})$$

With $wTCat_{sep}(f,g)$ the wake turbulence separation distance between f and g. To detect exit conflicts, two different times are monitored at the exit of the link: an early time of arrival at the exit node t^{out}_{min} and a late time of arrival at the exit node t^{out}_{max}. Conflicts at the exit of the link are detected the same way as at the entry of the link. The distance between two consecutive flights is computed by using the exit times : $dist_{dest}(f,g) = V^{g}_{max} \times (t^{g,out}_{min} - t^{f,out}_{max})$ Conflict detection at entry and exit only checks if the aircraft are properly separated when entering or exiting the link. However, within the link, faster aircraft can catch-up slower ones and cross them, thus implicating a loss of separation during the crossing. To detect catch-up conflicts, a comparison is made between the entry and the exit sequence of flights. Let's assume $\mathcal{S}^{l}_{entry}$ and $\mathcal{S}^{l}_{exit}$ respectively the sequences of flights at the entry and at the exit of the link l, $pos : (\mathcal{F}, \mathcal{S}) \rightarrow \mathbb{N}$ the function that gives the position of a flight in a sequence. There is a catch-up conflict if :

$$\exists f \in \mathcal{F} /\ pos(f, \mathcal{S}^{l}_{entry}) \neq pos(f, \mathcal{S}^{l}_{exit})$$

3.7 Objective function

The mathematical modeling of the problem leads to a multi-objective optimization problem which aims at minimizing parameters linked to conflict occurrences, flight level modification, speed modification and also flights timing. The main objective is to reduce the global conflict count both on nodes and links. Let's consider c_{obj} the total conflict objective value.

$$c_{obj} = \sum_{n \in \mathcal{N}} c_n + \sum_{l \in \mathcal{L}} c_l$$

Two values related to flight level modifications are evaluated in the objective function:

$$fl_{obj} = fl_{changes} + fl_{gap}$$

where $fl_{changes}$ represents the total number of aircraft which decision implies FL change and fl_{gap} is the summation of the difference between the final flight level and the preferred flight level of each flight. As for the flight level gap objective, we also want to minimize the gap between the initial speed and the final speed for each flight. For that, the speed objective s_{obj} is expressed as follows.

$$s_{obj} = \sum_{f \in \mathcal{F}} |\Delta V_f|$$

Finally, we also want to make sure that the flights will arrive at their last node at the requested time with a reasonable (minimal) error margin. Thus, the time objective is expressed as follows.

$$t_{obj} = \sum_{f \in \mathcal{F}} |t^{f,last\_node}_{max} - t^{f,last\_node}_{planned,max}|$$

The global objective function f is the sum of all the objectives listed above.

$$f = c_{obj} + fl_{obj} + s_{obj} + t_{obj}$$

The complexity of our model mainly depends on the dimension of the problem (number of flights involved) and the range of the decision variables. If we consider n_s options for the speed changes and n_{fl} options for the FL changes, then for N_f flight the size of the solution space is given by $(n_s \times n_{fl})^{N_f}$. In our model, $\Delta V \in [-6, +3]$ and $\Delta FL \in [-2, +2]$, the size of the solution space can be expressed by $(10 \times 5)^{N_f}$. To address such combinatorial complexity, we have developed a meta-heuristic based on a sliding-window simulated annealing.

4 Simulated annealing

This method, as described by [9], is based on the annealing process which is a physical process consisting of heating up a solid until it melts, then cooling it down in order to obtain a perfect crystallized form. During this process, the resulting structural properties depend on the residual energy in the material which is influenced by the rate of cooling. Thus, the cooling phase must be controlled in order not to get trapped in a locally optimal structure with high energy crystals, resulting in imperfections. In combinatorial optimization, the equivalent process is the simulated annealing which aims at finding a solution with minimal cost.

4.0.1 Principle

Let's consider the following elements [4]. Let Ω be the solution space (the set of all possible solutions). Let $f : \Omega \rightarrow \mathbb{R}$ be an objective function defined on the solution space. The goal is to find a global minimum, ω^* ($\omega^* \in \Omega$ such that $f(\omega) \geq f(\omega^*)$ for all $\omega \in \Omega$). The objective function must be bounded to ensure that

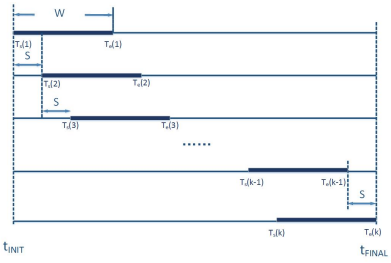

Figure 4 Sliding window principle [8],W: the length of the sliding window, S: time shift of the sliding window

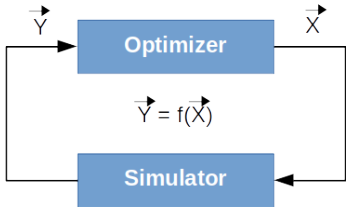

Figure 3 Simulated annealing principle

ω^* exists. Define $N(\omega)$ to be the neighborhood function for $\omega \in \Omega$. Therefore, associated with every solution, $\omega \in \Omega$, are neighboring solutions, $N(\omega)$, that can be reached in a single iteration of a local search algorithm. Simulated annealing starts with an initial solution $\omega \in \Omega$. A neighboring solution $\omega^{'} \in N(\omega)$ is then generated. The candidate neighboring solution is accepted as the current solution based on the acceptance probability.

$$P\{\text{Accept } \omega^{'} \text{ as next solution}\} =$$

$$\begin{cases} exp[-(f(\omega^{'} - f(\omega))/t_k)] & \text{if } f(\omega^{'} - f(\omega) > 0 \\ 1 & \text{if } f(\omega^{'} - f(\omega) \leq 0. \end{cases}$$

As illustrated in figure 3, each solution vector $\vec{X}$ which gather together the decision variables of aircraft is evaluated in a simulator process. This simulation is very powerful in the sense it can take into account many realistic aspects of the operational system. The result of the simulation generates the objective function which will be used by the optimizer (simulated annealing). The initial temperature is essential to define the behavior of the algorithm when it comes to the acceptance of solutions. We have set the initial temperature in order to get 80% of solutions to be initially accepted. We have used a geometric cooling law ($t_{k+1} = \alpha.t_k$ with α in the range $0.8 - 0.99$.Efficient stopping criterion can prevent the simulated annealing algorithm from performing unnecessary computations [4], thus reducing its global execution time. In our case, three criteria are used to stop the cooling procedure: when the final temperature T_{final} reaches a defined fraction of the initial temperature T_{init}, when the evaluated criteria reach the value zero, or when the evaluated criteria remain constant over a defined number of iterations.

4.0.2 Neighborhood function

Small changes on a local solution are performed using a neighborhood function. The efficiency of simulated annealing is highly influenced by the neighborhood function used. In our case, each solution is a set of flight decisions and determining a neighbor solution consists only on randomly modifying a flight decision in our set. The method used to choose the random flight is similar to the roulette wheel selection [8]. On each flight decision, a performance criterion is evaluated, it corresponds to the sum of the total number of conflicts on the flight and the resulting delay. All the decision performances are added up together then a target value is randomly chosen between zero (0) and the total performance sum. Then, the cumulative sum of decision performances is calculated starting from the first decision until the cumulative sum reaches the previously chosen target value. The index at which it stops marks the flight decision on which a change will be done to generate a neighbor solution. Choosing the decision to change with this method guarantees that the selected flight is the one which is most likely to generate more conflicts and a large absolute delay. Given a flight decision, generating a neighbor solution consists in modifying the flight level delta ΔFL and the speed delta ΔV. Both modifications are done within their respective discrete range of values $[-2, +2]$ and $[-6, +3]$, with the same probability of occurrence $p = 0.5$.

4.1 Sliding window

The sliding window approach as proposed by [8] is a technique based on the receding horizon control. It consists into dividing the entire time horizon into smaller equal intervals and thus evaluating the state of the network within the small time intervals. The evaluation interval starts from the earliest time and progressively moves forward in time with a defined step until reaching the latest times. With this approach, it is not necessary to evaluate all the flights at once during the optimization since not all flights are involved in each sliding window interval. This method is more convenient in a dynamic environment with uncertainties and improves the computational time of the optimization process. Figure 4 illustrate how the sliding windows are generated all along the global optimization time interval.

Parameter	Value
Temperature reduction coefficient (α)	0.95
Number of iterations at each temperature step	200
Cooling stopping temperature	$0.0001 \times T_0$
Maximum repetitions	35
Sliding window time shift (S)	30 min
Sliding window time length (W)	2 hours
Probability of speed change	0.5
Probability of flight level change	0.5

Table 2 Simulated annealing / Sliding window test parameters

Criteria	Value
Initial conflict count	407
Solved conflicts	407
Solved conflicts ratio	100%
Climbed flights	61
Descended flights	66
Total FL changes	127
FL changes ratio	18.41%
Accelerated flights	4
Slowed down flights	4
Speed changes	8
Speed change ratio	1.16%
Total speed variation	35.029 kts
Average speed variation	4.38 kts
Total absolute delays	611 seconds
Average absolute delays	76 seconds
Execution time	11min 23sec

Table 3 Optimization results on LFPG data

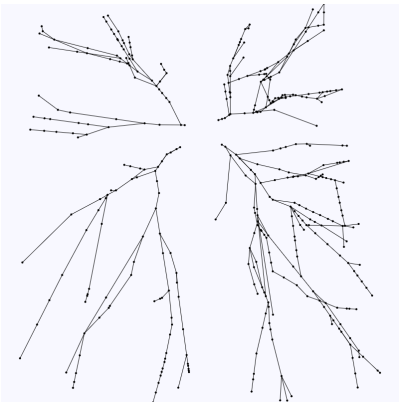

Figure 5 Routes towards Paris CDG airport

	BANOX	LORNI	MOPAR	OKIPA
Medium	85	163	75	167
Heavy	13	57	72	58
Total	**98**	**220**	**147**	**225**

Table 1 Flights distribution by TMA entry point

5 Results

This section presents the results of our algorithm on real-life traffic data at Paris CDG. Paris CDG TMA is accessible through four entry points: OKIPA, MOPAR, LORNI and BANOX. Our experimentation airspace is focused on the En-route portion of traffic arriving at CDG airport which corresponds to four areas each having a TMA entry point as last node as we can see in figure 5. All nodes (waypoints) except the four entry points are constrained with a 5NM separation minimum. As for the TMA entry points, the minimum separation between aircraft is 8NM. Concerning the wind data, we considered wind components with an angle of 30 degrees relative to the geographical north. Its intensity has been set within the range 7-10 kts all over the test airspace. A simple analysis of the network structure shows 742 different nodes with 817 links. This will contribute to increase the complexity and so the resolution time. In our scenario, we will consider a 24-hours traffic data of 690 flights corresponding to the filed flight plans arriving at LFPG on July 28th, 2016. The arrivals are distributed among the four entry points of CDG Terminal Area (TMA) with the majority of flights arriving at LORNI and OKIPA nodes (Table 1). There is also a distribution of medium and heavy wake turbulence category aircraft with a majority of medium aircraft.

5.1 Statistics

Our algorithm written in Java 8 has been executed on an Ubuntu 18.04 operating system PC equipped with an Intel Core i5-3230M processor (4×2.60GHz) and 8GB of memory. The simulated annealing algorithm has been tested with the parameters in Table 2. Given the length of the sliding window (2 hours) and the distribution of flights (Figure **??**), each resolution step on a sliding window will handle around 100 flights on average. A first analysis of the flight data detected 407 conflicts. The optimization algorithm analyzed the 690 flights and has solved the sequencing and the conflicts within a computation time of 11 minutes and 23 seconds. It solved all the 407 initially identified potential conflicts, which gives an efficiency rate of 100%. It changed flight levels on a total of 127 flights: 67 climbs and 66 descents. As for the speed, 04 flights have been accelerated and 04 slowed down which make a total of 08 speed changes. At the end, a total absolute delays value of approximately 611 seconds has been generated for the flights

with speed changes which gives an average of approximately 01 minute per modified flight (the 08 flights with speed modification). Very few flights have been affected by speed variations, also, the generated absolute delays, an average of 76 seconds per modified flight is still acceptable. Table 3 summarizes all those changes.

6 Conclusion

This paper introduced the work done on the sequencing of the air traffic in En-route segments influenced by constraint points. We saw that En-route controllers are getting more involved in the pre-sequencing of arriving flows when they are still in the cruising phase which causes an increase of their workload. On the other hand, airlines wish to have efficient flights with few flight level changes around a certain preferred vertical profile and also less maneuverings due to conflict avoidance. To solve those issues, we have developed a decision support tool which can assist controllers in their tasks for sequencing traffic and solving conflicts in En-route airspace. After reviewing the concepts and previous works related to our subject, we based our study on a mathematical modeling of the problem followed by an optimization algorithm in order to extract traffic sequences. Using the simulated annealing algorithm for optimizing flights decisions appeared to be a good choice given its efficiency and adaptability properties. A first trial of our solution on real traffic data over Paris airspace displayed a resolution ratio of 100% for conflict solving and an acceptable level of speed and flight level changes. Also, the generated delays due to the compliance with sequencing constraints and conflict resolution appeared within an acceptable range. Moreover, a preliminary version of the algorithm was able to generate flights instructions that can be directly applicable by controllers. Apart from this first test scenario, the solution as it has been designed is able to simultaneously provide En-route sequencing for several airports arrival flows. Even if it has b=not been used in the Paris CDG case, the algorithm can also manage time constraints for some points in the airspace (points connecting countries with letter of agreement). On the other hand, it can also be helpful for airports not yet equipped with an arrival management system as long as the constraints at TMA entry points are well defined. As another advantage, our solution will facilitate novel arrival techniques and procedures such as Continuous Descent Operation (CDO) and Point Merge System (PMS).

References

[1] Airbus. *Global Market Forecast : Growing Horizons 2017/2036*. 2017.

[2] Roco Barragan, Javier A. Prez Castan, Mario Sadornil, Francisco Saez Nieto, Javier Crespo, Rosa Arnaldo, and Cristina Cuerno. Optimised algorithms for extended arrival managers. 9 2014.

[3] Valentin Courchelle, Daniel Delahaye, and Daniel Gonzalez-Arribas. Simulated annealing for strategic traffic deconfliction by subliminal speed control under wind uncertainties. 2017.

[4] Darrall Henderson, Sheldon H Jacobson, and Alan W Johnson. The theory and practice of simulated annealing. In *Handbook of metaheuristics*, pages 287–319. Springer, 2003.

[5] SESAR JU. Integrated sequence building/optimization of queues. Technical report, 2015.

[6] SESAR JU. Extended arrival management (aman) horizon, 2018. (Accessed: 03 July 2018).

[7] Qing Li, Yicheng Zhang, and R. Su. A flow-based flight scheduler for en-route air traffic management. 49:353–358, 12 2016.

[8] Ji MA. Aircraft merging and sequencing problems in tma. Master's thesis, 2015.

[9] S.A.-H Soliman and A.-A.H Mantawy. *Modern Optimization Techniques with Applications in Electric Power Systems*, volume XVIII. Springer, 2012.

[10] Adan Vela, Senay Solak, William Singhose, and John-Paul Clarke. A mixed integer program for flight-level assignment and speed control for conflict resolution. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 5219–5226. IEEE, 2009.

7 COPYRIGHT