A Distributed Air Traffic Flow Management Model for European Functional Airspace Blocks

Sameer Alam
School of Engineering and IT,
University of New South Wales, Canberra, Australia
Email: s.alam@adfa.edu.au

Supatcha Chaimatanan
Geo-informatics and Space Technology Development Agency,
Bangkok, Thailand
Email: supatcha@gistda.or.th

Daniel Delahaye
Laboratory in Applied Mathematics, Computer Science
and Automatics for Air Transport,
Ecole Nationale de l'Aviation Civile, Toulouse, France
Email: delahaye@recherche.enac.fr

Eric Feron
Decision and Control Laboratory,
School of Aerospace Engineering,
Georgia Institute of Technology, Atlanta, USA
Email: feron@gatech.edu

***Abstract*—Functional airspace blocks (FAB) concept is adopted by the European airspace to allows cooperation between airspace users to manage the air traffic flow, while ensuring efficiency, safety, and fairness without constraints of geographical boundaries. This integration of airspace introduces more flexibility to manage aircraft trajectory and airspace usage. In this paper, we propose a distributed air traffic flow management model to address four-dimensional trajectory planning over the European FAB. The goal is to enable effective information sharing between airspace blocks in strategic planning, and to minimize interaction between trajectories. The proposed model and overall methodology is implemented and tested with a full day air traffic data over the European airspace. Interaction-free 4D trajectories are produced within computational time acceptable for the operational context, which shows the viability of the distributed model and interaction minimization approach for effective FAB implementation.**

I. Introduction

The main objective of an Air Traffic Flow Management (ATFM) system is to ensure safety, reduce delays, and balance demand and capacity among different components of the air transportation system [10]. In European airspace, ATFM activities are carried out by Eurocontrol's Network Manager Operations Centre (NMOC, previously called CFMU) which acts as a coordinator between air navigation service providers (ANSPs) and key stake holders, such as airlines, airports, and military.

However, fragmentation of airspace along national boundaries creates structural inefficiencies, sub-optimal flight routing and makes implementation of ATFM strategies quite challenging [6]. To address these challenges, a concept of Functional Airspace Blocks (FAB) is developed by Eurocontrol. A FAB is defined in the Single European Sky legislative package as *"an airspace block based on operational requirements and established regardless of state boundaries, where the air navigation services and related functions is performance-driven and optimised through enhanced cooperation among ANSPs or an integrated provider"* [2]. Nine FAB initiatives have been taken (Figure 1); two of these have already been implemented, namely the UK-Ireland and Denmark-Sweden FABs [1].

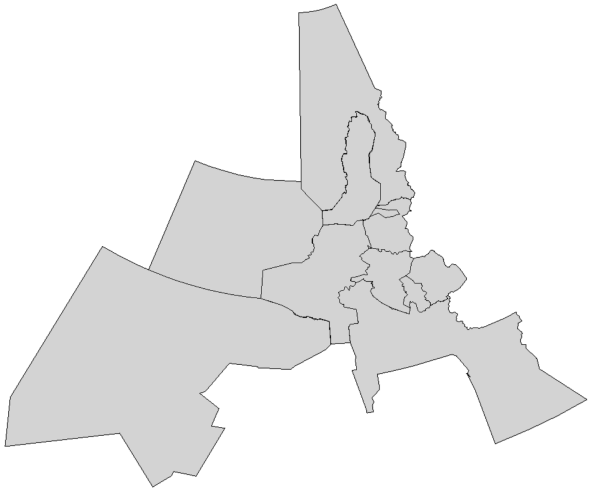

Fig. 1. European airspace is divided into nine Functional Airspace Blocks: NEFAB (North European FAB), Denmark-Sweden FAB, BALTIC FAB, FABEC (FAB Europe Central), FABCE (FAB Central Europe), DANUBE FAB, UK-IRELAND FAB and South West FAB.

Establishment of FABs will have consequences for NMOC operations. One of the key challenges will be to implement the centralized ATFM strategies in the individual FAB and develop the basis for the cooperation among the FABs [3]. Delegation of ATFM to individual FABs can lead to demand-capacity imbalances as each FAB may try to

optimize traffic flow in its own airspace without considering other FAB requirements.

In this paper, we propose a distributed ATFM model which form the basis for interaction among FABs to implement ATFM strategies. The goal of distributed ATFM model is generate strategic 4D trajectories which minimize flight interaction in both three-dimensional space and in the time domain by extending a trajectory planning method developed by co-authors in [7] and [11].

The paper is organized as follows: Section II describes the concept of the proposed distributed ATFM model. Section III presents the proposed model and methodology in a mathematical framework. Then, a method to compute interaction between trajectories is presented in Section IV. Resolution algorithm to the problem is explained in Section V. Finally, numerical results are presented and discussed in Section VI.

II. Concept Description

Centralized ATFM though offers a fair distribution and demand-capacity balance at a global level but stake-holders can only provide inputs and are not part of decision making. It may also lead to large number of pairwise reversals, i.e., the resulting order of flight arrivals can be quite different as compared to the original published flight schedules [5]. Because of this deviation from the original flight ordering, it may becomes difficult to implement such a solution locally.

In a distributed ATFM the decision-making responsibilities are shared between a number of airspace users (airlines, ANSPs, Military, Airports). Some examples of distributed ATFM are Ration by Schedule (RBS) [4] and Ground Delay Program (GDP) [9]. However the current state-of-the art is limited mostly to strategic planning, and the users' participation in planning reduces as the planning interval becomes smaller. With the gradual implementation of FAB concept in the European airspace, any distributed ATFM systems must also take into consideration multiple FAB interactions.

In a FAB scenario, the traffic flow management will be highly interdependent and will demand a significant cooperation with other FABs. One way to achieve this is by having distributed 4D trajectory planning amongst FABs which can ensures conflict-free trajectory for each aircraft. This will then translates into trajectory based operations where aircraft are required to fly a negotiated conflict-free trajectory through respective FABs.

A. Proposed Model

The main idea proposed in the paper is a distributed decision making model which can enable effective information sharing among the FABs for 4D trajectory planning which are conflict free. The goal of the proposed method is to separate a given set of aircraft trajectories in space and time domain by allocating an alternative fight plan (route and departure time) to each fight in a given FAB.

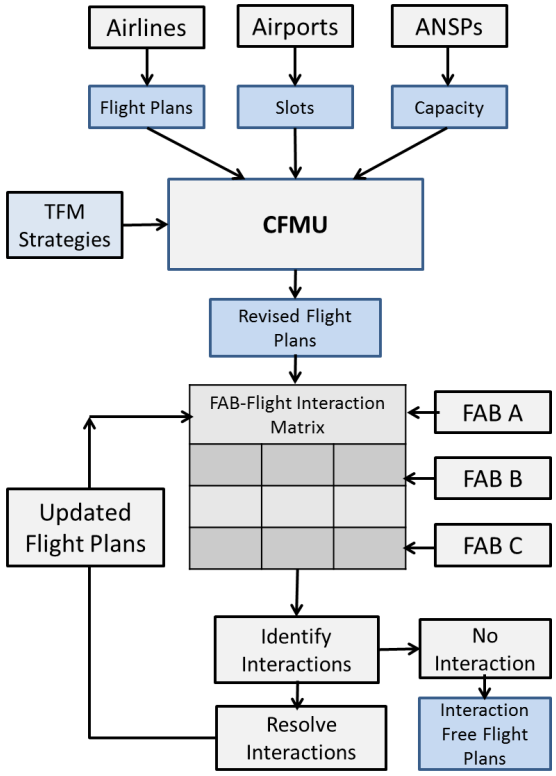

Fig. 2. Proposed Concept for information sharing among the FABs for interaction-free trajectory planning.

Instead of trying to satisfy the capacity constraint, we focus on minimizing the interaction between trajectories. Where an interaction between trajectories occurs when two or more trajectories have an effect on each other; for instance, when trajectories occupy the same space at the same period of time. Therefore, contrary to the concept of conflict, the measurement of interaction does not only refer to the violation of minimum separation requirements. It also allows us to take into account other separation criteria such as minimum separation time between aircraft crossing at the same point.

The proposed concept is developed as follows: as illustrated in Figure 2, airspace users (airlines, airports, ANSPs etc.) submits relevant information (flight plan, slots, capacity) to NMOC, which then applies centralized traffic flow management strategies to match demand with capacity and other airspace constraints and generate revised flight plans. These flight plans are then used as an input to a *FAB-Flight interaction Matrix*. It is a 2D matrix which captures the flight interaction information between and within FABs. One dimension of the matrix is termed *Controlling FABs* and the other dimension is termed *Intermediate FABs*. A

Controlling FAB is defined as a FAB where a given flight is originated or activated (in case of an enroute flight entering European airsapce), whereas an intermediate FAB is defined as a FAB through which a given flight traverse (over fly), terminates (lands) or exits the European airspace.

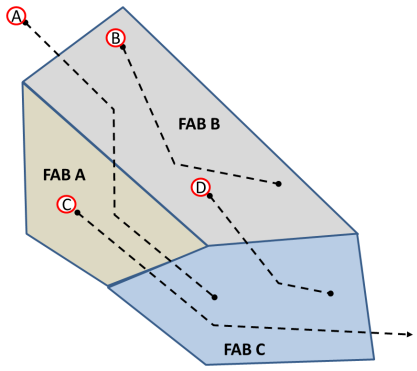

Fig. 3. An airspace divided into three FABs (FAB A, FAB B and FAB C) with four flight (A, B, C and D) scenario.

As illustrated in Figure 3, Flight A enters FAB B from outside so it is the the Controlling FAB. Flight A traverse through FAB A and terminates in FAB C so they are termed as Intermediate FAB. Similarly, for Flight B originates and terminates in FAB B, so FAB B is both Controlling and Intermediate FAB for flight B. For Flight C the origin is in FAB A (Controlling FAB) and it traverse through FAB C (intermediate FAB) before exiting the airspace. Thus a flight may have multiple intermediate FABs but will have only one controlling FAB.

The FAB-Flight interaction Matrix captures how many interactions are caused by flights from Controlling FAB in the Intermediate FABs. The flight interactions can then be resolved by implementing time-space separation in the Controlling FAB (from where the flight originated/activated) and the flight plans are updated accordingly. Once resolved, flight interactions are recomputed (using revised flight plans) and the FAB-Flight Interaction Matrix is updated. This process continues until all the flight interactions are resolved.

B. FAB-Flight Interaction Matrix

The FAB-Flight Interaction Matrix is developed as follows: as illustrated in Figure 4, for N FABs in a given airspace A, a 2D matrix of N rows and N columns is developed. The row vector of the matrix represents the number of flight interaction caused by a Controlling FAB C_j in the Intermediate FABs I_i for $i = 1$ to N. The column vector of the matrix represents the number of flight interaction caused by the Controlling FABs C_j for $j = 1$ to N in an Intermediate FAB I_i

Intermediate FABs

Controlling FABs	FAB I_A	FAB I_B	FAB I_C
FAB C_A	$INT(C_A, I_A)$	$INT(C_A, I_B)$	$INT(C_A, I_C)$
FAB C_B	$INT(C_B, I_A)$	$INT(C_B, I_B)$	$INT(C_B, I_C)$
FAB C_C	$INT(C_C, I_A)$	$INT(C_C, I_B)$	$INT(C_C, I_C)$
$\sum V =$	$\sum U_A$	$\sum U_B$	$\sum U_C$

Fig. 4. FAB-Flight Interaction Matrix

$$I_i = \left[\begin{array}{cccc} INT(C_j, I_A) & INT(C_j, I_B) & \ldots & INT(C_j, I_i) \end{array} \right] \quad (1)$$

$$C_j = \left[\begin{array}{c} INT(C_A, I_i) \\ INT(C_B, I_i) \\ .... \\ INT(C_N, I_i) \end{array} \right] \quad (2)$$

For example, as illustrated in Figure 4, flight interactions due to flights controlled by FAB C_A in the Intermediate FAB I_A is given by row FAB C_A and column FAB I_A and denoted by $INT(C_A, I_A)$. Similarly, the number of flight interactions due to flights controlled by FAB C_C in the same intermediate FAB i.e. FAB I_c is given by row FAB C_c and column FAB I_c.

Therefore, the total number of flight interactions U, in a given FAB i, can be given by summing the column vector:

$$U_i = \sum_{j=1}^{N} INT(C_j, I_i) \quad (3)$$

The total flight interaction V in a given airspace A (which comprises of N FABs) can be given by

$$V = \sum_{j=1}^{N} U_j \quad (4)$$

This can be further normalized to find out the relative contribution of each FAB, in overall flight interactions, for a given airspace A.

$$U_{j_{rel}} = U_j / V \quad (5)$$

for j = 1 to N.

C. Traffic Flow Management Strategy

The distributed Traffic Flow Management strategy is developed as follows: first the Intermediate FAB with highest number of flight interaction is identified as a candidate FAB (equation 6).

$$FAB \quad I_i = MAX(U_A, U_B..., U_N) \tag{6}$$

Then for the FAB I_i, the Controlling FAB C_j which generated highest number of flight interaction is identified (equation 7).

$$FAB \quad C_j = MAX(INT(C_A, I_i), INT(C_B, I_i), ..., INT(C_N, I_i)) \tag{7}$$

The ATFM strategies (Space-Time separation) are then applied on randomly selected (fitness-proportional selection) flights in Controlling FAB C_j. Flight interactions are recomputed given the revised flight plans and the FAB-Flight Interaction Matrix is updated. This process is repeated until the FAB-Flight Interaction Matrix is interaction free. Figure 5 illustrates the updated process, where the decision made by each FABs are evaluated by the optimization process. Then, the information of interaction based on FABs-flight interaction matrix is feed back to each FABs, which then make new decision and repeat the process until a solution that leads to minimum overall interaction is reached.

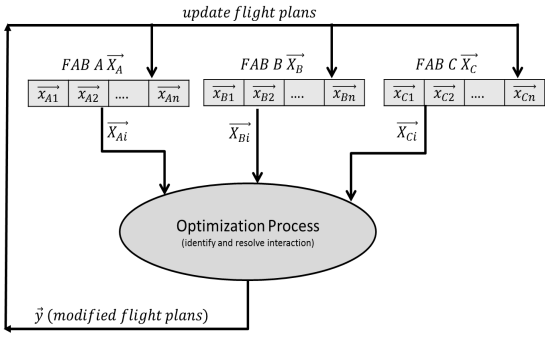

Fig. 5. FAB-Flight Interaction Matrix update process.

III. Mathematical Modelling

This section sets the mathematical framework of the distributed air traffic flow management methodology we are proposing. First, a definition of *interaction* between trajectories is given. Then, the route / departure-time allocation techniques adapted for the distributed ATFM model is presented.

A. Interaction between trajectories

The concept of interaction between trajectories is introduced in [7] and [11]. It is a measurement that indicates when two or more trajectories occupy the same space at the same period of time. It is different from the *conflict* situation, which corresponds simply to a violation of the minimum *separation* (i.e. 5 NM horizontally and 1,000 ft vertically). Additional separation conditions, such as time separation, topology of trajectory intersection, distance between trajectories, etc. can also be taken into account in the concept of interaction.

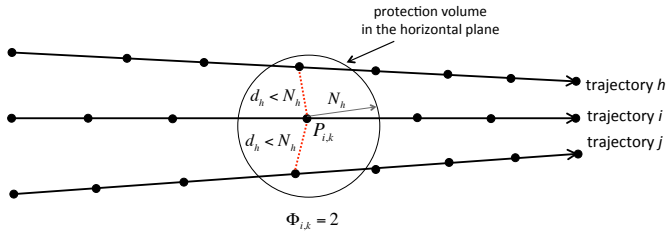

Fig. 6. Interactions, $\Phi_{i,k}$, at sampling point $P_{i,k}$ of trajectory i.

Consider a given set of N discretized 4D trajectories, where each trajectory i is a time sequence of 4D coordinates, $P_{i,k}(x_{i,k}, y_{i,k}, z_{i,k}, t_{i,k})$, specifying that aircraft must arrive at a given point $(x_{i,k}, y_{i,k}, z_{i,k})$ at time $t_{i,k}$, for $k = 1, \ldots, K_i$, and K_i is the number of *sampling points* of trajectory i.

Consider a point k of trajectory i, *interactions at point* $P_{i,k}$, denoted $\Phi_{i,k}$, may be defined as the total number of times that the protection volume around point $P_{i,k}$ is violated. Figure 6 illustrates interaction in the horizontal plane between $N = 3$ trajectories measured at point $P_{i,k}$.

The *interaction associated with trajectory i*, denoted Φ_i, is therefore defined to be:

$$\Phi_i = \sum_{k=1}^{K_i} \Phi_{i,k}. \tag{8}$$

Finally, the *total interaction between trajectories*, Φ_{tot}, for a whole traffic situation is simply defined as:

$$\Phi_{tot} = \sum_{i=1}^{N} \Phi_i = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \Phi_{i,k}. \tag{9}$$

One can observe that the measurement of the interaction between trajectories implicitly take into account the duration of conflict between trajectories. A practical methodology to compute the value of the interaction between trajectories in a large-scale context is presented in Section IV.

B. Route/Departure-time allocation

In order to separate the trajectories in 3D space and time domain, we rely on a route/departure-time allocation techniques introduced in [7] and [11]. The objective is to find alternative 4D trajectory for each flight, so as to minimize the total interactions between trajectories.

Given data. A problem instance is given by:

- A set of initial N discretized 4D trajectories with associated controlling FAB;
- The discretization time step, Δt;
- The number of allowed virtual waypoints, M;

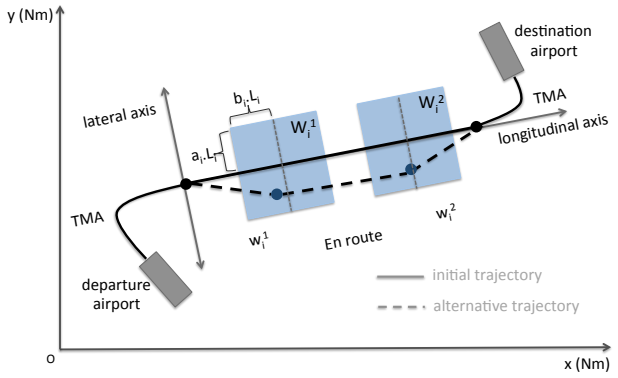

Fig. 7. Initial and alternative trajectories with rectangular-shape possible location of $M = 2$ virtual waypoints.

- The maximum allowed advance departure time shift of each flight i, $\delta_a^i < 0$;
- The departure time shift step size, δ_s;
- The maximum allowed delay departure time shift of each flight i, $\delta_d^i > 0$;
- The maximum allowed route length extension coefficient of each flight i, $0 \le d_i \le 1$;
- The length of the initial en-route segment of each flight i, $L_{i,0}$.

The alternative departure time and the alternative route to be allocated to each flight are modelled as follows.

Alternative departure time. The departure time of each flight can be shifted by a positive (delay) or a negative (advance) time shift. Let $\delta_i \in \Delta_i$ be a departure time shift attributed to flight i, where Δ_i is a set of acceptable time shifts for flight i. The departure time t_i of flight i is therefore $t_i = t_{i,0} + \delta_i$, where $t_{i,0}$ is the initially-planned departure time of flight i. The departure time shift δ_i will be limited to lie in the interval $\Delta_i := [\delta_a^i, \delta_d^i]$. Common practice in airports conducted us to rely on a discretization of this time interval using time-shift step size δ_s. This yields $N_a^i := \frac{-\delta_a^i}{\delta_s}$ possible advance slots and $N_d^i := \frac{\delta_d^i}{\delta_s}$ possible delay slots of flight i. Therefore, we define the set, Δ_i, of all possible departure time shifts of flight i by

$$\begin{aligned}\Delta_i :=& \{-N_a^i.\delta_s, -(N_a^i-1).\delta_s, \ldots, \\ & -\delta_s, 0, \delta_s, \ldots, (N_d^i-1).\delta_s, N_d^i.\delta_s\}.\end{aligned} \tag{10}$$

Alternative trajectory design. In this work, an alternative trajectory is constructed by placing a set of virtual waypoints, denoted

$$w_i = \{w_i^m | w_i^m = (w_{ix'}^m, w_{iy'}^m)\}_{m=1}^M, \tag{11}$$

near the initial en-route segment and then by reconnecting the successive waypoints with straight-line segments as illustrated in Fig. 7. To limit the route length extension, the alternative en-route profile of flight i must satisfy:

$$L_i(w_i) \le (1+d_i), \tag{12}$$

where $L_i(w_i)$ is the length of the alternative en-route profile determined by w_i. Fig. 7 illustrated initial and alternative trajectories, constructed with $M = 2$ waypoints, where the location of each waypoint is constrained to be in a rectangular-shape possible location. Let $W_{ix'}^m$ be a set of all possible normalized longitudinal locations of the m^{th} virtual waypoint on trajectory i. For each trajectory i, the normalized longitudinal component, $w_{ix'}^m$, is set to lie in the interval:

$$W_{ix'}^m := \left[\left(\frac{m}{1+M} - b_i\right), \left(\frac{m}{1+M} + b_i\right)\right], \tag{13}$$

where b_i is a (user-defined) parameter that defines the range of possible normalized longitudinal component of the m^{th} virtual waypoint on trajectory i. To obtain a regular trajectory, the normalized longitudinal component of two adjacent waypoints must not overlap, i.e.

$$\left(\frac{m}{1+M} + b_i\right) < \left(\frac{m+1}{1+M} - b_i\right) \tag{14}$$

and hence the user should choose b_i so that

$$b_i < \frac{1}{2(M+1)}. \tag{15}$$

Let $W_{iy'}^m$ be a set of all possible normalized lateral locations of the m^{th} virtual waypoint on trajectory i. Similarly, the normalized lateral component, $w_{iy'}^m$, is restricted to lie in the interval:

$$W_{iy'}^m := [-a_i, a_i], \tag{16}$$

where $0 \le a_i \le 1$ is a (user-defined) model parameter that defines the range of possible normalize lateral location of the m^{th} virtual waypoint on trajectory i, chosen a priori so as to satisfy (12).

Let us set the compact vector notation: $\delta := (\delta_1, \delta_2, \ldots, \delta_N)$, and $\mathbf{w} := (w_1, w_2, \ldots, w_N)$,

We shall denote by u_i the components of u. It is a vector whose components are related to the modification of the i^{th} trajectory, thereby our decision variable is:

$$u := (\delta, \mathbf{w}).$$

Finally, the interaction minimization problem can be formulated as a mixed-integer optimization problem, as follows:

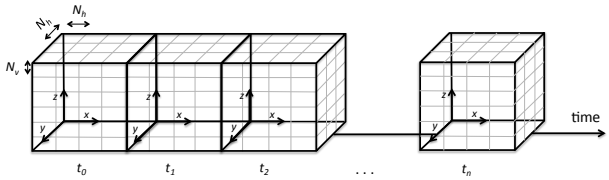

Fig. 8. Four-dimension (3D space + time) grid for conflict detection, illustrated as a time series of 3D grids which is sampled with discretization time step $\Delta t = t_n - t_{n-1}$. The size of each cell in the 3D grids is defined by the minimum separation requirements (N_h and N_v).

$$\min_{u=(\delta,\mathbf{w})} \Phi_{tot}(u)$$

subject to

$$\begin{aligned} \delta_i \quad &\in \Delta_i, \\ w^m_{ix'} &\in W^m_{ix'}, \\ w^m_{iy'} &\in W^m_{iy'}, \\ \text{for all } i &= 1,\ldots,N, m = 1,\ldots,M, \end{aligned} \tag{P1}$$

where $W^m_{ix'}$, and $W^m_{iy'}$ are defined by (13), and (16) respectively.

IV. INTERACTION DETECTION

In order to evaluate the objective function, at a candidate solution, (u), one needs to compute interaction between the N aircraft trajectories. To avoid the $\frac{N(N-1)}{2}$ time-consuming pair-wise comparisons, which are prohibitive in our continental-scale application context, we propose a grid-based interaction detection scheme which is implemented in a so-called *hash table* as presented in [7] and [11].

First, the airspace is discretized using a four-dimensional grid (3D space + time), as illustrated in Figure 8. The size of each cell in the 4D grid is defined by the minimum separation requirement and the discretization time step, Δt (see below). Then, for each given 4D coordinate $P_{i,k}(x_{i,k}, y_{i,k}, z_{i,k}, t_{i,k})$ of each trajectory i, we identify which cell, says $C_{i,j,k,t}$, of the 4D grid contains $P_{i,k}(x_{i,k}, y_{i,k}, z_{i,k}, t_{i,k})$.

Next, we consider each such cell $C_{i,j,k,t}$ and we successively check its surrounding cells (there are $3^3 = 27$ such neighbouring cells, including cell $C_{i,j,k,t}$ itself). If one cell is occupied by an aircraft other than aircraft i itself, the horizontal distance (d_h) and the vertical distance (d_v) between the corresponding aircraft coordinates are measured. A violation of the protection volume is identified when both $d_h < N_h$ and $d_v < N_v$.

In order not to underestimate interaction, and to avoid using small value to Δt which leads to large number of trajectory samples and long computational time, we propose an inner-loop algorithm, detecting interaction between two

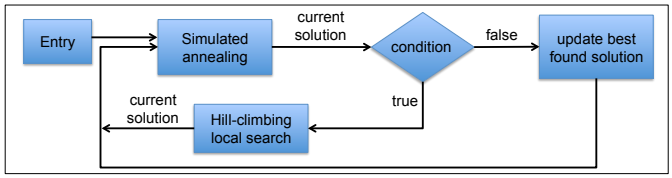

Fig. 9. Structure of the proposed hybrid algorithm of simulated annealing and hill-climbing local search methods.

sampling times, t and $t + \Delta t$, by *interpolating* aircraft positions with a sufficiently small step size, t_{interp}. The interpolation is performed only if no interaction is detected at time t. Then, one checks each pair of these interpolated points. The algorithm stops when an interaction is identified or when every pair of the interpolated points has been checked.

V. RESOLUTION ALGORITHMS

The 4D trajectory planning methodology for a distributed ATFM model, presented in this paper, relies on the interaction minimization problem introduced in Section III whose objective function values are obtained by simulation through the interaction detection scheme developed in Section IV. To solve the problem, a hybrid metaheuristic approach adapted to handle an air-traffic assignment problem at the continent scale is used. It relies on a classical simulated annealing (SA) algorithm and two different local-search (LS) modules. The LS allows the system to intensify the search around a potential candidate solution while the SA allows the system to escape from a local trap and thereby ensuring the exploration of the solution space. The proposed hybrid algorithm combines the SA and the local search algorithm such that the LS is considered as an inner-loop of the SA, which will be performed when a pre-defined condition is satisfied. The structure of the proposed hybrid algorithm of SA and LS methods is illustrated in Figure 9.

This hybrid SA-LS optimization algorithm has been applied to solve 4D trajectory planning at continent scale in [7] and [11]. In these works, the objective is, also, to minimize the total interaction between trajectories. However, the air traffic flow were managed based on a centralized decision making. In order to apply the hybrid SA-LS algorithm to our proposed distributed ATFM model, the LS search strategy and the neighborhood structure are modified according to our proposed ATFM strategy described in Subsection II-C (to be described below).

For our problem, the simulated annealing proceeds as follows. First, we evaluate the objective function at the current configuration $(\mathbf{w}, \delta)_C$. It is denoted Φ_C. Then a neighboring solution, $(\mathbf{w}, \delta)_N$, is generated by a neighborhood function. Then, a new solution for this chosen flight is generated according to a pre-defined neighborhood

structure. If the neighborhood solution improves the objective function value, then it is accepted. Otherwise, it is accepted with a probability $e^{\frac{-\Delta\Phi}{T}}$, where $\Delta\Phi = \Phi_N - \Phi_C$ is the difference of energy between current state C and new state N. When the maximum number of iterations, n_T, at a given temperature is reached, the temperature is decreased according to the user-provided pre-defined schedule, and the process is repeated until the pre-defined final temperature, T_{final}, is reached. More detail on simulated annealing can be found, for instance, in [8].

Local search modules. The local search modules we use are heuristic methods that accepts a new solution only if it yields a decrease of the objective function. The process repeats until no further improvement can be found or until the maximum number of iterations $n_{T_{LOC}}$ is reached. The two local-search modules correspond to the two following strategies:

- **Intensification of the search on one Particular Trajectory (PT).** Given a flight i, this state-exploitation step focuses on improving the current solution by applying a local change from the neighborhood structure only to flight i (only the decision variables (w_i, δ_i) are affected).
- **Intensification of the search on the Interacting Trajectories (IT).** Given a flight i, this state-exploitation step applies a local change, from the neighborhood structure, to every flight that is both subjected to the same controlling FABs as flight i, and currently interacting with flight i.

Neighborhood structure. The hybrid algorithm we are proposing relies on a neighborhood structure to determine the next move. First, one has to determine which flight to be modified. In the framework of a distributed ATFM we are proposing, first one has to determine the controlling FABs which generates the highest proportion of interaction. Then, one chose one of the interacting flights in such FABs if $\Phi_i \geq \tau \cdot \Phi_{avg}$, where τ is a user-defined parameter and $\Phi_{avg} = \Phi_{tot}/N$ is the average value of interaction.

In order to generate a neighborhood solution for a given flight, i, from the current configuration $(w_i, \delta_i)_C$, one has to determine whether to modify the location of waypoints or to modify the departure time in the next move. In general, searching for the solution in the time domain would be more preferable since it does not induce extra fuel consumption. However, empirical tests show that limiting the search to only that degree of freedom results in prohibitive computational time. Therefore, we introduce a user-defined parameter P_w to control the probability to modify the location of the waypoints w_i and such that the probability to modify rather the departure time is $1 - P_w$. For a given flight i, the neighborhood operator generates a new set of virtual waypoints or a new alternative departure time according to this probability P_w.

Hybrid algorithm (SA and LS). Here is how the above-mentioned methods are combined. The methods are carried out according to pre-defined probabilities, which are proportional to the control temperature, T. The probability to carry out simulated annealing step, P_{SA}, is:

$$P_{SA}(T) = P_{SA,min} + (P_{SA,max} - P_{SA,min}) \cdot \frac{T_0 - T}{T_0}, \qquad (17)$$

where $P_{SA,max}$ and $P_{SA,min}$ are the maximum and minimum probabilities to perform the SA (pre-defined by the user). The probability of running the LS module, P_{Loc}, is given by:

$$P_{Loc}(T) = P_{Loc,min} + (P_{Loc,max} - P_{Loc,min}) \cdot \frac{T_0 - T}{T_0}, \qquad (18)$$

where $P_{Loc,max}$ and $P_{Loc,min}$ are the maximum and minimum probabilities to perform the local search (defined analogously). And, finally the probability of carrying out both SA and the local search (successively), P_{SL}, is:

$$P_{SL}(T) = 1 - (P_{SA}(T) + P_{Loc}(T)). \qquad (19)$$

A key factor in tuning this hybrid algorithm is to reach a good trade off between exploration (diversification) and exploitation (intensification) of the solution space.

VI. Numerical experiments

The proposed distributed ATFM model is implemented in Java. The overall methodology is tested with air traffic data involving flights over the European FABs, consisting of nine FABs listed in Table I. First, it is tested with a set of traffic consist of 4,000 flights over the European FABs on a UNIX platform with 1.7 GHz processor and 8 GB memory. The parameter values chosen to specify the optimization problem are given in Table II. The parameter values that specify the resolution algorithm are given in Table III. Then, it is tested with a full day en-route air traffic over the European FABs consisting of 26,122 flights on a UNIX platform with 2.4 GHz processor and 32 GB memory, using the same parameters as given in Table II and III, except that this time the number of iteration N_I is set to 2,700.

TABLE I
European FABs.

No.	FAB name
1	Baltic FAB
2	Blue Med
3	FAB Central Europe
4	Danube FAB
5	FAB Europe Central
6	NEFAB
7	NUAC program
8	SW Portugal-Spain FAB
9	FAB UK Ireland

TABLE II
CHOSEN (USER-DEFINED) PARAMETER VALUES FOR THE OPTIMIZATION PROBLEM.

Parameter	Value
Discretization time step, Δt	20 seconds
Discretization time step for possible departure-time shift, δ_s	20 seconds
Maximum departure time shift, $delta_a^i = \delta_d^i := \delta$	120 minutes
Maximum allowed route length extension coefficient, d_i	0.20
Maximum allowed flight level shifts, $l_{i,max} := l_{max}$	2
Maximum number of virtual waypoints, M	3

TABLE III
EMPIRICALLY-SET (USER-DEFINED) PARAMETER VALUES OF THE RESOLUTION METHODOLOGY.

Parameter	Value
Minimum probability to perform SA step, $P_{SA,min}$	0.8
Maximum probability to perform SA step, $P_{SA,max}$	0.9
Minimum probability to perform local search step, $P_{Loc,min}$	0.4
Maximum probability to perform local search step, $P_{Loc,max}$	0.6
Number of iterations at each temperature step, N_I	400
Number of iterations of the inner-loop local search step, $n_{T_{LOC}}$	5
Geometrical temperature reduction coefficient, β	0.99
Final temperature, T_f	$(1/500).T_0$
Inner-loop interpolation sampling time step, t_{interp}	5 seconds
Probability to modify horizontal flight profile, P_w	1/3
Probability to modify flight level, P_l	1/3
Threshold value, Φ_τ	0.5 Φ_{avg}

The initial and final total interaction between trajectories, the computation time, and the number of iterations performed to solve the distributed problems compared to the one solved based on centralized decision making methodology (proposed in [7] and [11]) are reported in Table IV. Evolution of the interaction between the 4,000 trajectories in each FAB and in the FAB-Flight Interaction Matrix during the optimization process using distributed and centralized models are presented in Figure 10, 11, 12, and 13 respectively. One can observe in Figure 10 and 12 that the FABs that has the highest level of initial interaction is FAB number 5 (FAB Europe Central). The controlling FABs which generates the highest interaction in FAB number 5 are FAB number 1,4, and 5. In the case of distributed model, the resolution algorithm tries to minimize the total interaction by modifying flight plans of flights associated to FAB number 1,4, and 5 before other FABs. While, in the centralized model case, the resolution algorithm modifies flight plans of flights that involves in high interaction, without taking into account the FAB-flight interaction information.

TABLE IV
INITIAL AND FINAL TOTAL INTERACTION BETWEEN TRAJECTORIES FOR 4,000 TRAJECTORIES.

case	N	ATFM strategy	initial Φ_{tot}^D	final Φ_{tot}^D	solved interactions	no. of iterations	cpu time (mins)
1	4,000	Distributed	48,272	0	100%	8,306	2.47
		Centralized		0	100%	5,035	2.08
2	26,122	Distributed	266,318	0	100 %	509,924	369.67
		Centralized		0	100%	632,002	563.53

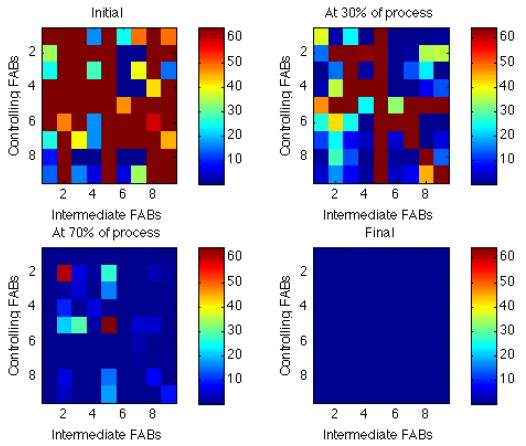

Fig. 10. Evolution of the FAB-flight interaction matrix for 4,000 trajectories during the optimization process using distributed ATFM model.

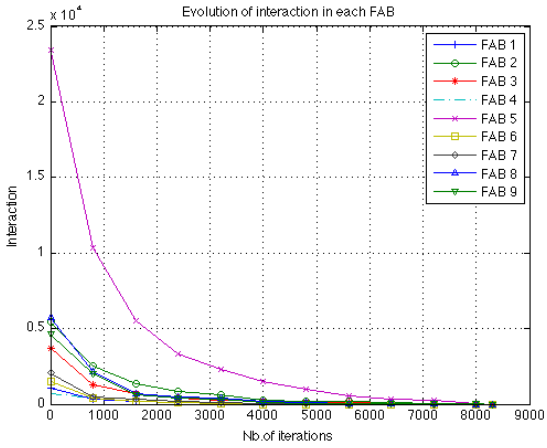

Fig. 11. Evolution of the interaction in each FAB for 4,000 trajectories during the optimization process using distributed ATFM model.

Similarly, the evolution of the interaction between the full-day traffic, consisting of 26,122 trajectories, in each FAB and in the FAB-flight matrix during the optimization process using distributed and centralized models are presented in Figure 14, 15 , 16, and 17 respectively.

Again, one can observe that the FABs that in the case of distributed model, the resolution algorithm tries to minimize the total interaction by modifying flight plans of flights associated to the FABs that generate high interaction to the global system before other FABs. Although the trajectories can be separated only by modifying the horizontal flight profile and the departure time of each flight, the resolution algorithm finds an interaction-free solution in both

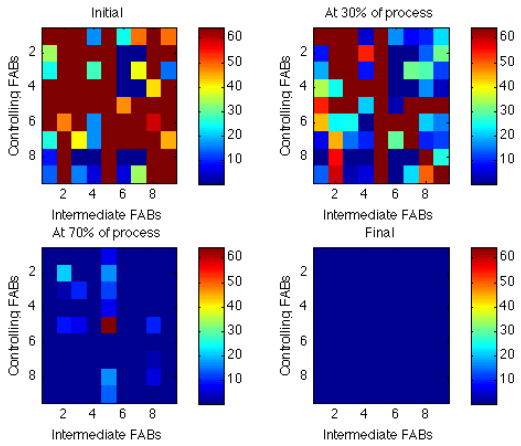

Fig. 12. Evolution of the FAB-flight interaction matrix for 4,000 trajectories during the optimization process using centralized ATFM model.

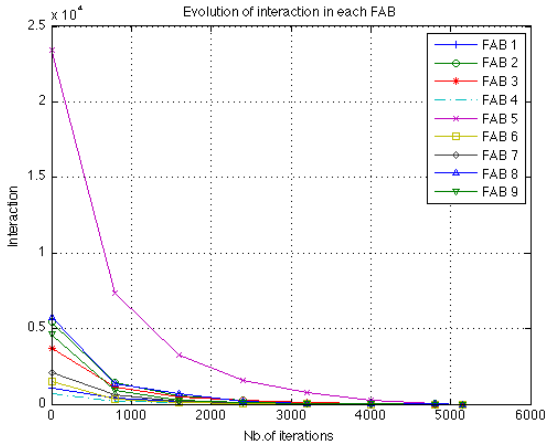

Fig. 13. Evolution of the interaction in each FAB for 4,000 trajectories during the optimization process using centralized ATFM model.

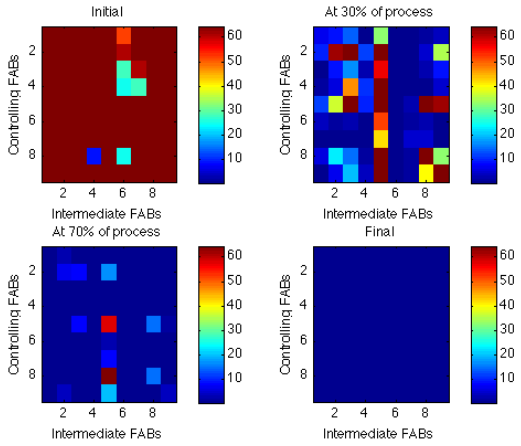

Fig. 14. Evolution of the FAB-flight interaction matrix for 26,122 trajectories during the optimization process using distributed ATFM model.

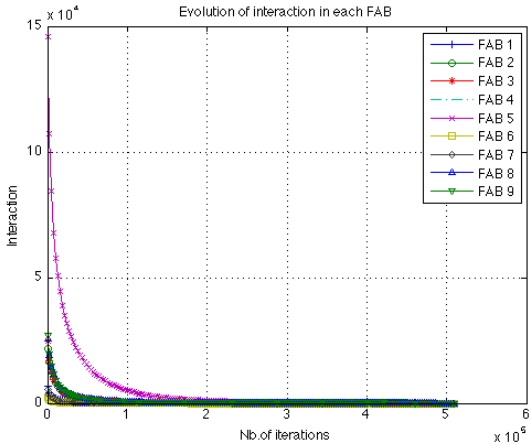

Fig. 15. Evolution of the interaction in each FAB for 26,122 trajectories during the optimization process using distributed ATFM model.

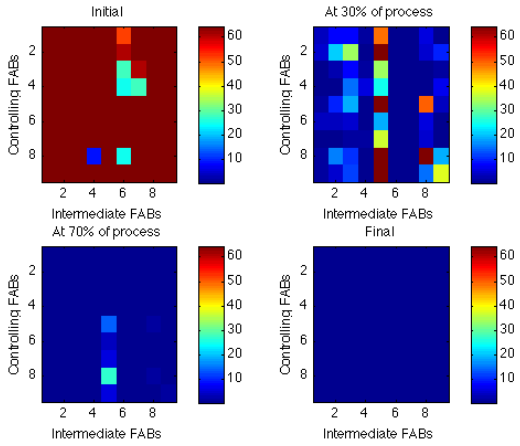

Fig. 16. Evolution of the FAB-flight interaction matrix for 26,122 trajectories during the optimization process using centralized ATFM model.

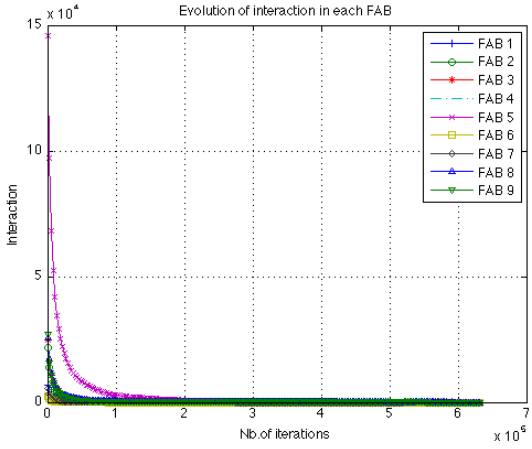

Fig. 17. Evolution of the interaction in each FAB for 26,122 trajectories during the optimization process using centralized ATFM model.

case 1 and case 2. In the case of small data set (case 1), the computation time to reach interaction-free solution are not significantly different when using distributed and centralized model. However, when the number of traffic increases (case 2), the distributed model converges to interaction-free solution faster than the centralized model. This is because the distributed ATFM strategy targets more on the FABs that generates high interaction. Nevertheless, both models yield interaction-free solution with computation time that is still viable for strategic planning and pre-tactical planning purpose.

VII. Discussions

In this paper, we have presented a distributed air traffic flow management model, aiming to minimize total interaction between aircraft 4D trajectories at continent-scale. The objective was to develop a basis of information exchange and interaction among FABs for implementing distributed AFTM. The overall methodology is implemented and tested on a continent-size air traffic over the European FABs, and then compared with the one obtained using centralized ATFM model. The methodology based on distributed model converges to interaction-free solution faster than the centralized model. Furthermore, the color maps shows that the convergence rate of the distributed model is much better than centralized model and is viable for strategic planning as well as pre-tactical planning purpose. However, in practice, interfaces and different implementations, time delays and so forth need to be managed which present difficulties. These difficulties may not be compensated by these theoretical advantages.

References

[1] Eurocontrol Functional Airspace Block. http://www.eurocontrol.int/dossiers/fabs. Accessed: 2017-01-06.

[2] Single european sky legislative package. Technical Report Regulation (EC) No. 1070/2009, European Comission, Strasbourg, 2009.

[3] Andreas Angenendt. DFS on Central Flow Managament Unit. *Skyway magazine Eurocontrol*, 9(39), 2005.

[4] Cynthia Barnhart, Douglas Fearing, Amedeo Odoni, and Vikrant Vaze. Demand and capacity management in air transportation. *EURO Journal on Transportation and Logistics*, 1(1-2):135–155, 2012.

[5] Dimitris Bertsimas and Shubham Gupta. Fairness in air traffic flow management. In *INFORMS Meeting, San Diego-CA, USA*, volume 54, page 100, 2009.

[6] Kenneth Button and Rui Neiva. Single european sky and the functional airspace blocks: Will they improve economic efficiency? *Journal of Air Transport Management*, 33:73–80, 2013.

[7] S. Chaimatanan, D. Delahaye, and M. Mongeau. Large Scale 4D Trajectory Planning. *IEEE Computational Intelligence Magazine, Institute of Electrical and Electronics Engineers*, 9, 1989.

[8] J. Dreo, A. Petrowski, P. Siarry, and E. Taillard. *Metaheuristics for hard optimization*. Springer, 2006.

[9] Charles N Glover and Michael O Ball. Stochastic optimization models for ground delay program planning with equity–efficiency tradeoffs. *Transportation Research Part C: Emerging Technologies*, 33:196–202, 2013.

[10] ICAO Doc 9971 AN/485, International Civil Aviation Organization, Montral, Quebec, Canada. *Manual on Collaborative Air Traffic Flow Management*, 2 edition, 2014.

[11] A. Islami, S. Chaimatanan, and D. Delahaye. Large Scale 4D Trajectory Planning. In *EIWAC 2015, 4th ENRI International Workshop on ATM/CNS*, Tokyo, Japan, November 2015. ENRI.

Acknowledgement

Authors would like to thank Leilla Zerrouki from Eurocontrol for useful discussions on FAB design that greatly improved the manuscript.

Authors Biography

Sameer Alam is a senior lecturer in aviation at the University of New South Wales, Canberra, Australia and a visiting researcher at ENAC (French Civil Aviation University). He obtained his BS degree in maths from University of Rajasthan, India (1994), M.Tech in computer sc. from Birla Institute of Technology, India (2000) and PhD in artificial intelligence from University of New South Wales, Australia (2008). His research interests are in nature inspired computation and multi-objective optimization applied to risk assessment of advanced ATM concepts.

Supatcha Chaimatanan obtained her Aerospace engineering degree from Kasetsart university in 2007, MS in Aerospace mechanics and avionics from ISAE in 2010 and PhD in applied mathematics and computer sc. from the Aeronautics and Astronautics doctoral school in 2014. Currently, she is a researcher at Geo-Informatics and Space Technology Development Agency in Thailand, and a visiting researcher at ENAC (French Civil Aviation University). Her research interest includes mathematical optimization for air traffic flow management, trajectory design and satellite mission planning.

Daniel Delahaye is Professor and head of the optimization team of MAIA laboratory at ENAC (French Civil Aviation University). He obtained his engineer degree from the ENAC and master of science in signal processing from the national polytechnic institute of Toulouse in 1991. He obtained his PhD in automatic control from the Aeronautic and Space National School in 1995 and did a post-doc at the Department of Aeronautics and Astronautics at MIT in 1996. He get his tenure in applied mathematics in 2012. He conducts research on mathematical optimization for airspace design (sector design, SID-STAR design).

Eric Feron graduated from Ecole Polytechnique, France, Ecole Normale Suprieure, France, and Stanford University, USA with the BS, MS and PhD, respectively. Eric Feron has held tenured faculty appointments at MIT (1993-2005), Georgia Tech (2005-Current), ONERA (1993-1998), and ENAC (09/2011-Current). His current research interests include airport and airspace dynamics,UAVs and control theory. Currently, he is a Professor in the School of Aerospace Engineering at Georgia Tech.