



HAL
open science

A Meta-Heuristic Approach for Distributed Trajectory Planning for European Functional Airspace Blocks

Sameer Alam, Supatcha Chaimatanan, Daniel Delahaye, Eric Féron

► To cite this version:

Sameer Alam, Supatcha Chaimatanan, Daniel Delahaye, Eric Féron. A Meta-Heuristic Approach for Distributed Trajectory Planning for European Functional Airspace Blocks. *Journal of Air Transportation*, 2018, 26 (3), pp.81-93. 10.2514/1.D0078 . hal-01951070

HAL Id: hal-01951070

<https://enac.hal.science/hal-01951070v1>

Submitted on 11 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Meta-Heuristic Approach for Distributed Trajectory Planning for European Functional Airspace Blocks

Sameer Alam¹

Nanyang Technological University, Singapore

Supatcha Chaimatanan²

Geo-informatics and Space Technology Development Agency, Bangkok, Thailand

Daniel Delahaye³

Ecole Nationale de l'Aviation Civile, Toulouse, France

Eric Feron⁴

Georgia Institute of Technology, Atlanta, GA, USA

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 the constraints of geographical boundaries. This integration of airspaces allow for flexibility in airspace management and aircraft trajectory planning. This paper proposes a distributed air-traffic flow management model to address four-dimensional (4D) trajectory planning over the European FAB. The proposed method is based on a meta-heuristic approach which uses a hybrid algorithm of simulated annealing and hill-climbing local search to separate a given set of aircraft trajectories in space and time domain (we term this as *flight interaction*), by allocating an alternative flight plan (route and departure time) to each flight. An innovative data structure, termed as FAB-Flight interaction matrix, captures the flight interaction in-

¹ Associate Professor, School of Mechanical and Aerospace Engineering, Email: sameeralam@ntu.edu.sg

² Researcher, Geo-informatics and Space Technology Development Agency, Email: supatcha@gistda.or.th

³ Professor, Laboratory in Applied Mathematics, Computer Science and Automatics for Air Transport, Email: delahaye@recherche.enac.fr

⁴ Professor, School of Aerospace Engineering, Email: feron@gatech.edu

formation between and within FABs. The proposed distributed model is implemented and tested with two air traffic data sets comprising of 4,000 flights (3 hours traffic) and 26,000 flights (one full day traffic data over the European airspace). The performance of the model is then compared with a centralized air-traffic flow management model on scalability and interaction minimization. Results indicates that, though both the approaches were able to achieve interaction-free trajectory planning within computational time acceptable for the operational context, distributed model converges faster to interaction-free solution as traffic size increases; which shows the viability of the distributed model 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 [1]. 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 stakeholder, such as airlines, airports, and the military.

However, fragmentation of airspace along national boundaries creates structural inefficiencies, sub-optimal flight routing and makes implementation of ATFM strategies quite challenging as highlighted by Button et al. 2013 [2] and Dubot et al. 2015 [3]. To address these challenges, a concept of Functional Airspace Blocks (FAB) is being 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"* [4].

As illustrated in Figure 1, nine FAB initiatives have been taken: UK-Ireland FAB, Baltic FAB, BLUE MED FAB, Danube FAB, Denmark-Sweden FAB, FAB CE, FABEC, North European FAB



Fig. 1 European airspace is divided into nine Functional Airspace Blocks viz. UK-Ireland FAB, Baltic FAB (Lithuania, Poland), BLUE MED FAB (Albania, Cyprus, Greece, Italy and Malta), Danube FAB (Bulgaria, Romania), Denmark-Sweden FAB, FAB CE (Austria, Bosnia & Herzegovina, Croatia, Czech Republic, Hungary, Slovak Republic, Slovenia), FABEC (Belgium, France, Germany, Luxembourg, the Netherlands and Switzerland), North European FAB (Estonia, Finland, Latvia, and Norway) and South West FAB (Portugal, Spain).

and South West FAB, two of these have already been implemented, namely the UK-Ireland and Denmark-Sweden FABs and rest are in progress [5].

Establishment of FABs will have consequences for NMOC operations. Angenendt, 2005 [6] has argued that one of the key challenges will be to implement the centralized ATFM strategies in the individual FAB and to develop the basis for the cooperation among the FABs. 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.

As per regulation (EC) No 551/2004 of the European Parliament on the organization and use of the airspace in the single European sky by European Commission, one of the specific requirements for the implementation of FABs is that their implementation should be supported by a safety case i.e. implementing of FAB must not result in increased instances of loss of separation. Subsequent

evaluation of Functional Airspace Block initiatives and their contribution to performance improvement by Eurocontrol Performance Review Report 2008 [7], Adler et al., 2014 [8] and Luppo et al., 2016 [9] have concluded that without increased coordination and information exchange amongst Airlines, ANSPs, Military and Airports FAB may not be able to deliver its objectives.

Arijana et al., 2014 [10] in their analysis of expected ATM changes in central Europe have concluded that higher degree of collaboration amongst the ANSPs and that the states must ensure that ATM can provide services in conjunction with their neighbors with greater deal of cooperation. Button and Neiva, 2013 [2] examined the economic efficiency of functional airspace blocks and found that greater local coordination of ANSPs may play in the longer-term policies of establishing a full Single European Sky. Alvarez et al., 2013 [11] suggested that one of the indicators of FAB success is that ATM performances at network level are achieved in a collaborative distributed decisions making context.

The above research motivated us to establish a distributed ATFM model which forms the basis of interaction and information exchange among FABs to implement ATFM strategies. The goal of distributed ATFM model will be to generate strategic 4D trajectories which minimize flight interaction, a concept which considers both space and time elements of overlapping trajectories. Thus we attempt to address both, the safety and coordination, issues in FAB implementation.

The proposed method is based on a meta-heuristic approach which uses a hybrid algorithm of simulated annealing and hill-climbing local search to separate a given set of interacting aircraft trajectories by allocating an alternative flight plan (route and departure time) to each flight. The distributed decision making for flight interaction resolution stems from an innovative data structure, termed as *FAB-Flight interaction matrix*, which captures the flight interaction information between and within the FABs.

Simulated Annealing belongs to a special class of search methods, known as evolutionary algorithms which were introduced by Holland in 1992 [12]. Since, air traffic flow management problems are difficult to solve by other traditional methods because of non-convex, non-linear, or non-analytic characteristics, evolutionary search methods offers an effective approach of addressing open research problems in this domain [13]. Further, seminal work of Cheng et al., [14], Du et al. [15] and Delahaye

et al., [16] on optimization of airspace sectoring, airspace network optimization, traffic assignment, and en-route conflict resolution has added empirical support for evolutionary algorithms as an optimization methodology that is both effective and efficient in analyzing complex air traffic flow management problems.

The paper also extends a trajectory planning method, based on route and departure-time allocation technique, developed by co-authors in Chaimatanan, 2014 [17] by including a local search strategy for better exploitation of search knowledge and a neighborhood structure for effective identification of FABs and associated flights which causes high interaction. This enables interaction resolution of flights from FABs which causes higher interaction in the overall airspace. This leads to faster convergence of algorithm especially when number of flights is large.

The paper is organized as follows: Section II describes the concept of the proposed distributed ATFM model for Functional Airspace Blocks. 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 stakeholder 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 sequence of flight arrivals can be quite different as compared to the original published flight schedules [18]. Because of this deviation from the original flight ordering, it may become 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) [19] and Ground Delay Program (GDP) [20, 21]. 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

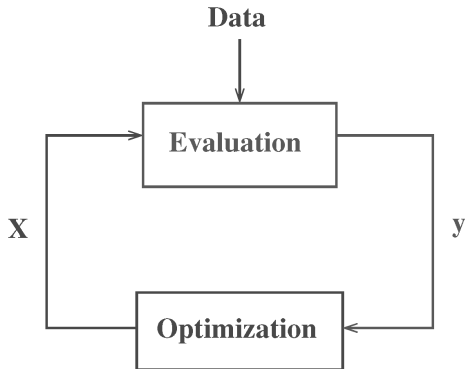


Fig. 2 Trajectory planning and optimization for FABs in Centralized ATFM.

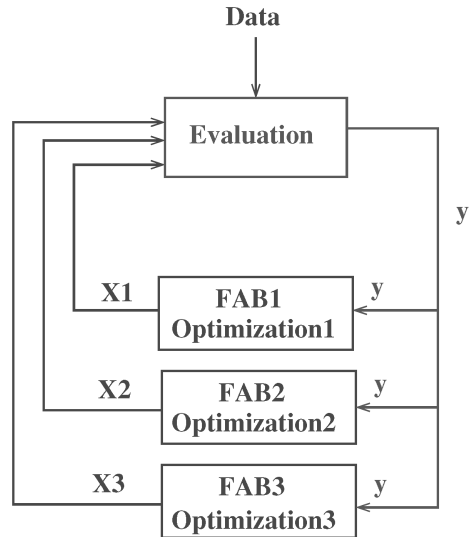


Fig. 3 Trajectory planning and optimization for FABs in Distributed ATFM.

interactions.

Figure 2 and 3 illustrates the difference in the centralized and distributed ATFM approach in terms of trajectory planning and optimization. In centralized ATFM the optimization algorithm controls the vector of decision variables, X , which are used by the simulation process in order to compute the performance (quality), y , of such decisions. However, In this approach, evolutionary algorithms may not be adapted to address such problems, mainly when the simulation environment requires huge amount of memory space as is often the case in nowadays real-life complex systems. As a matter of fact, in the case of a evolutionary algorithm based approach, the simulation environment has to be duplicated for each individual of the population of solutions, which may require an excessive amount of memory. In order to avoid this drawback, one may think about having distributed approach where only one simulation environment which could be used each time a point in the population has to be evaluated as follows. In order to evaluate one population, one first consider the first individual. Then, the simulation environment is initiated and the simulation associated with the first individual is run. The associated performance is then transferred to the optimization algorithm. After that, the second individual is evaluated, but the simulation environment must be first cleared from the events of the first simulation. The simulation is then run for the second

individual, and so on until the last individual of the population is evaluated. In this case the memory space is not an issue anymore, but the evaluation time may be excessive and the overall process too slow, due to the fact that the simulation environment is reset at each evaluation.

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 ensure conflict-free trajectory for each aircraft. This will then translate into trajectory based operations where aircraft are required to fly a negotiated conflict-free trajectory through respective FABs.

In long term planning, airline operation center (AOC) generates flight schedules and flight plans and takes into consideration seasonal demand/trends, passenger projection, spoke-hub configurations, maintenance Schedule, pilots/Technical staff roster and available slots at airports. While in medium term planning Air Navigation Service Providers project traffic loads based on airline schedules and flight plans for 2 to 12 Hours and takes into consideration weather patterns. In Europe, this is done centrally by Eurocontrol Central Flow Management Unit (CFMU), now known as NMOC, for all the member states. CFMU also ensures that in the medium term all flight plans are conflict and congestion free. In short term planning (Airspace Flow Planning) flight coordination is done on a regional scale with 30 Minutes to 2 Hour Timescale. Flow restrictions are used for overcrowding which includes Miles-in-Trail and Time-in-Trail. While in Tactical planning (Sector Traffic Planning) Sector Air Traffic Controller Executive (R-Side) is responsible for managing aircraft through the use of radar (5 minutes interval) and Sector Air Traffic Controller Planning (D-Side) is responsible for coordinating hand-offs with surrounding sectors (5 to 20 Minute Timescale).

The focus of this paper is on medium term planning where we envision that the demand and capacity balancing and initial flight planning will be taken care by NMOC and the responsibilities of ensuring that the flight plans are conflict and congestion free will rest on individual FABs where the flight is originating (i.e. the controlling FAB). This is a loose form of distributed air traffic flow management where the decision of de-confliction/de-congestion (collectively, we call it flight interaction) rests with FAB and not with a centralized body.

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 interaction 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 flight plan (route and departure time) to each flight in a given FAB.

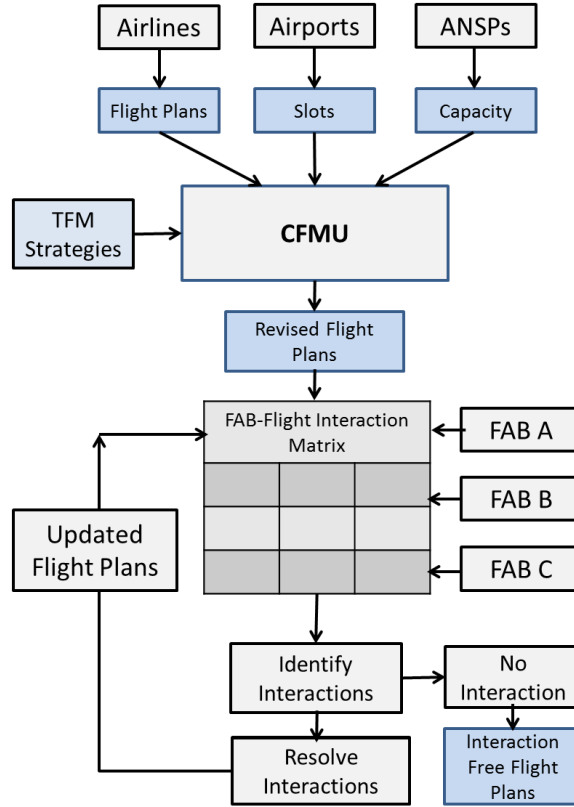


Fig. 4 Proposed Concept for information sharing among the FABs for interaction-free trajectory planning.

Since the focus of this paper is on medium-term planning, we assume that the problem of demand-capacity balancing are addressed in the strategic long-term planning. In medium term planning the key issues is minimizing the interaction between trajectories. 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 4, 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 airspace), whereas an intermediate FAB is defined as a FAB through which a given flight traverse (over fly), terminates (lands) or exits the European airspace.

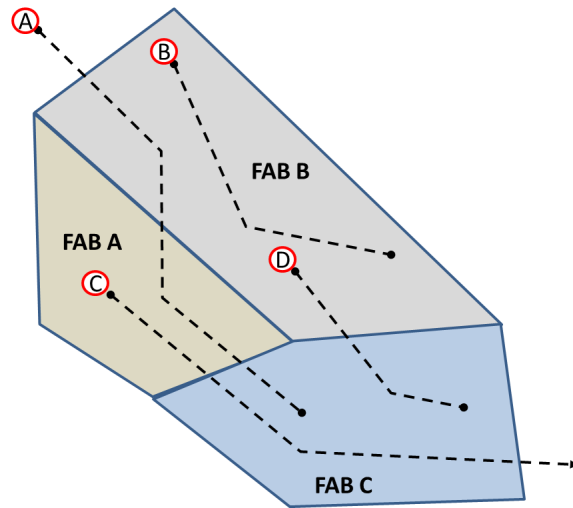


Fig. 5 An airspace divided into three FABs (FAB A, FAB B and FAB C) with four flight (A, B, C and D) scenario.

As illustrated in Figure 5, Flight A enters FAB B from outside so it is 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 6, 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 interactions 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 interactions caused by the Controlling FABs C_j for $j = 1$ to N in an Intermediate FAB I_i

		Intermediate FABs		
		FAB I_A	FAB I_B	FAB I_C
Controlling FABs	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. 6 FAB-Flight Interaction Matrix

$$I_i = \left[INT(C_j, I_A) \quad INT(C_j, I_B) \quad \dots \quad INT(C_j, I_i) \right] \quad (1)$$

$$C_j = \begin{bmatrix} INT(C_A, I_i) \\ INT(C_B, I_i) \\ \dots \\ INT(C_N, I_i) \end{bmatrix} \quad (2)$$

For example, as illustrated in Figure 6, 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

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

The average contribution of each FAB, in overall flight interactions, for a given airspace A can be given by:

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

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).

$$FAB \ I_i = MAX(U_A, U_B, \dots, U_N) \quad (6)$$

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

$$FAB \ C_j = MAX(INT(C_A, I_i), INT(C_B, I_i), \dots, INT(C_N, I_i)) \quad (7)$$

The ATFM strategies (Space-Time separation) are then applied on randomly selected (fitness-proportional selection) flights in Controlling FAB C_j . The fitness proportional method implies that the probability of a flight to be selected is simply proportionate to the number of interactions it causes. 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 7 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.

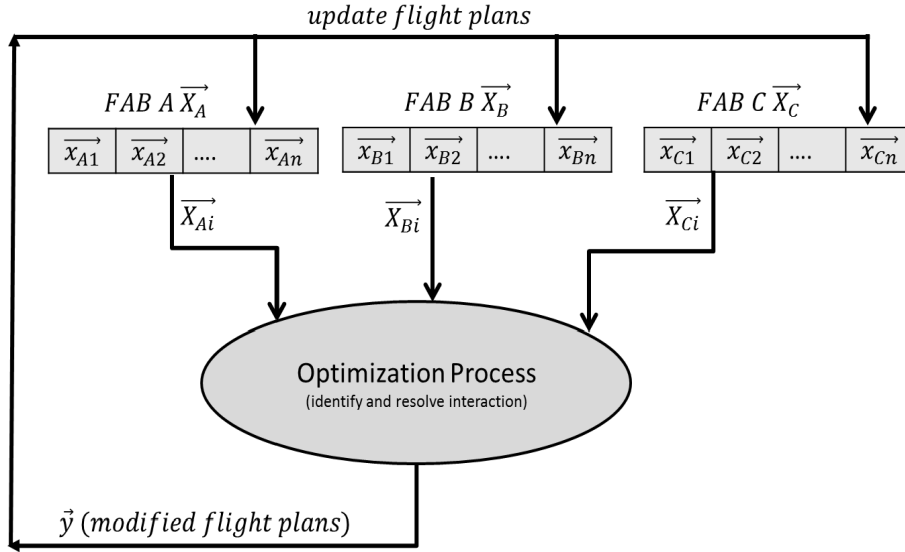


Fig. 7 FAB-Flight Interaction Matrix update process.

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.

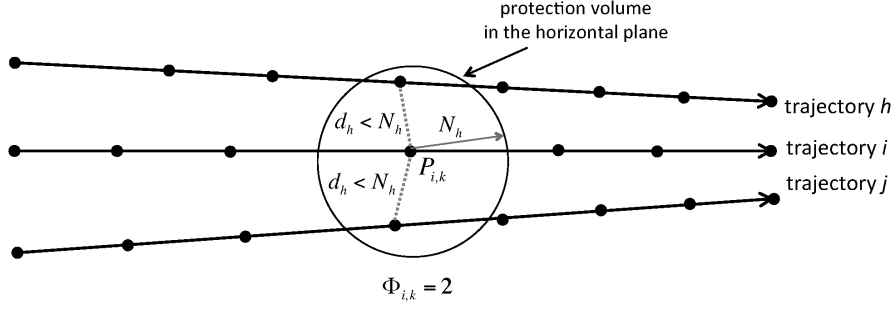


Fig. 8 Interactions, $\Phi_{i,k}$, at sampling point $P_{i,k}$ of trajectory i .

Then, the route / departure-time allocation techniques adapted for the distributed ATFM model is presented.

A. Interaction between trajectories

Interaction between trajectories 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). In this paper, the concept of interaction takes into consideration the duration of conflict. However, additional separation conditions, such as time separation, topology of trajectory intersection, distance between trajectories, etc. can also be taken into account.

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 $k = 1, \dots, K_i$, 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 8 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:

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

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

as:

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

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 formulate a route/departure-time allocation techniques based on [17]. 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 ;
- 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 , $0 \leq d_i \leq 1$;
- 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

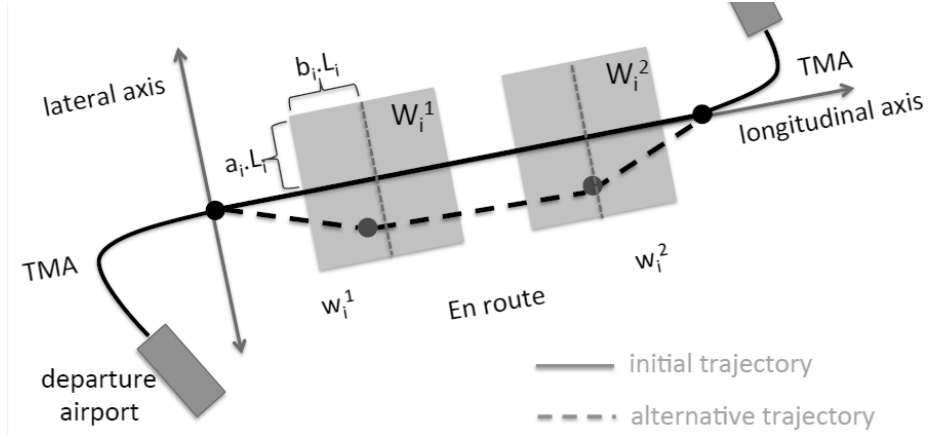


Fig. 9 Initial and alternative trajectories with rectangular-shape possible location of $M = 2$ virtual waypoints.

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

$$\Delta_i := \{-N_a^i \cdot \delta_s, -(N_a^i - 1) \cdot \delta_s, \dots, -\delta_s, 0, \delta_s, \dots, (N_d^i - 1) \cdot \delta_s, N_d^i \cdot \delta_s\}. \quad (10)$$

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

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

near the initial en-route segment and then by reconnecting the successive waypoints with straight-line segments as illustrated in Fig. 9. To limit the route length extension, the alternative en-route profile of flight i must satisfy:

$$L_i(w_i) \leq (1 + d_i), \quad (12)$$

where $L_i(w_i)$ is the length of the alternative en-route profile determined by w_i . Fig. 9 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:

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

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.

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

and hence the user should choose b_i so that

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

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:

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

where $0 \leq a_i \leq 1$ is a (user-defined) model parameter that defines the range of possible normalized 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: $\boldsymbol{\delta} := (\delta_1, \delta_2, \dots, \delta_N)$, and $\mathbf{w} := (w_1, w_2, \dots, w_N)$,

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:

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

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

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

subject to

$$\delta_i \in \Delta_i, \tag{P1}$$

$$w_{ix'}^m \in W_{ix'}^m,$$

$$w_{iy'}^m \in W_{iy'}^m,$$

$$\text{for all } i = 1, \dots, N, m = 1, \dots, M,$$

where $W_{ix'}^m$, and $W_{iy'}^m$ 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 use a grid-based interaction detection scheme [17] which is implemented in a so-called *hash table*.

First, the airspace is discretized using a four-dimensional grid (3D space + time) for conflict detection, as illustrated in Figure 10, 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). 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

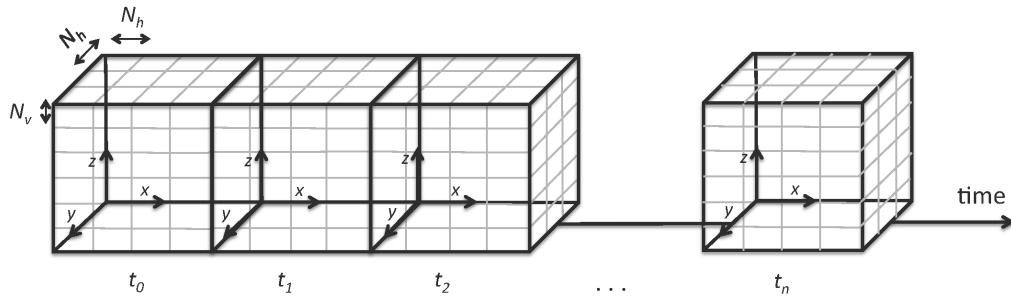


Fig. 10 Four-dimension (3D space + time) grid for conflict detection, illustrated as a time series of 3D grids.

algorithm, detecting interaction between two 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 concept of Simulated annealing (SA) in combinatorial optimization was introduced by S. Kirkpatrick et al. in 1983 [22]. It is a metaheuristic stochastic method of optimization that is well known for its ability to escape from local minima by allowing occasional moves that deteriorate the value of the objective function, such deteriorating moves being less and less as the number of iterations grows.

The concept of SA is based on a strong analogy with the physical annealing of materials. This process involves bringing a solid to a low energy state after raising its temperature. It can be summarized by the following two steps

- Bring the solid to a very high temperature until *melting* of the structure;
- Cool the solid according to a very particular temperature decreasing scheme in order to reach a solid state of minimum energy.

In the liquid phase, the particles are distributed randomly. It is shown that the minimum-energy state is reached provided that the initial temperature is sufficiently high and the cooling time is sufficiently long. If this is not the case, the solid will be found in a metastable state with non-minimal energy; this is referred to as hardening, which consists in the sudden cooling of a solid. Temperature in a physical system is analogous to Control parameters in the optimization problem, where the objective function of the problem is analogous to the energy state of the system. As, from an initial solution, Simulated Annealing algorithm can proceed in several iterations and at each iteration, a random neighbor is generated. Moves that improve the cost function are always accepted. Thus, at a given temperature, the lower the increase of the objective function, the more significant the probability of accepting the move and higher the temperature, the more significant the probability of accepting a worst move.

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 11.

Previously, hybrid optimization algorithm have been applied to solve 4D trajectory planning at continent scale in [17], however the air traffic flow were managed based on a centralized decision making. In order to apply the proposed hybrid SA-LS algorithm to distributed ATFM model, the LS search strategy and the neighborhood structure are modified according to proposed ATFM strategy, as described in Subsection II C.

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,

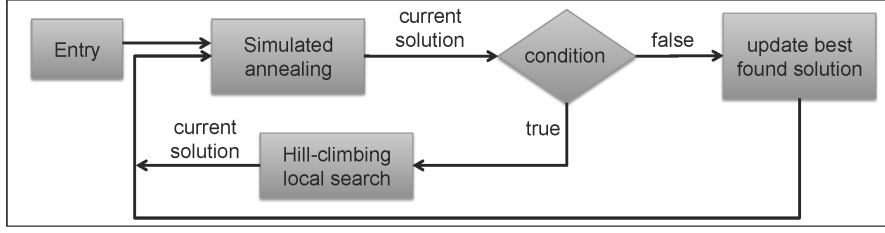


Fig. 11 Structure of the proposed hybrid algorithm of simulated annealing and hill-climbing local search methods.

$(\mathbf{w}, \boldsymbol{\delta})_N$, 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 [23].

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 proposed hybrid algorithm relies on a neighborhood structure to determine the next move. First, the controlling FAB, which generates the highest proportion of interaction, is identified. Then, a flight from the identified FAB is chosen such that the interaction associated with trajectory i , $\Phi_i \geq \tau \cdot \Phi_{avg}$, 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:

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

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:

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

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:

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

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 1. First, it is tested with a set of traffic consisting 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 2. The parameter values that specify the resolution algorithm are given in Table 3. Then, it is tested with a full day of 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 2 and 3, except that this time the number of iteration N_I is set to 2,700.

Table 1 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 FAB
8	SW Portugal-Spain FAB
9	FAB UK Ireland

Table 2 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 3 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_{TLOC}	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_r	$0.5 \Phi_{avg}$

The parameters in Table 2 which were domain dependent, were chosen based on the feedback from domain experts. For example, the number of flight level changes is kept at maximum of two flight levels (2000 feet) to minimize major changes in originally filed *cruise flight level*. Similarly, the maximum departure time shift is kept to two hours keeping in line with airline practices. The parameters in Tables 2 and 3 are adopted from the centralized AFTM methodology developed by co-authors in [17]. These parameters were determined by initial set of experiments and fine tuning. These parameters are kept same in this paper for comparison purpose.

Table 4 Initial and final interaction between 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		0	100%	8,306	2.47
		Centralized	48,272	0	100%	5,035	2.08
2	26,122	Distributed		0	100 %	509,924	369.67
		Centralized	266,318	0	100%	632,002	563.53

We have used following matrices for comparison:

- Initial and final interaction between trajectories,
- The number of iterations performed resolves the interactions,
- The computation time to resolve the interactions,
- How the solution converged (Color Map)

Time to converge and resolution of interactions for continental scale traffic were the two key metrics for the algorithm evaluation. Since this paper also introduces a new data structure- FAB-Flight Interaction matrix, which forms the basis for information exchange and identification of the flights that need interaction resolution, one of the key hypothesis is how the distributed algorithm will scale with increase in traffic from country level (4,000 flights) to continent level (26,000 flights) traffic sample.

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 [17]) are reported in Table 4. The FAB-Flight interaction matrix is presented as color map showing the number of interaction caused by controlling FAB into the Intermediate FAB. To better visualize the evolution of interaction resolution, the FAB-Flight interaction matrix color maps are reported at four key stages of experimentation: at the start, at 30%, 70% and at the end of the process.

A. Small Flight Sample: 4000 flights

Results indicate that, in terms of iterations the distributed model takes 8,306 iterations as compared to 5,035 iterations in centralized model to solve 48,272 interactions with 4000 flights. However in terms of CPU time the difference between the two approaches is not significant.

Figure 12 and Figure 14 presents the color map showing how interactions are resolved at different stages of the optimization process. Initially, the FAB that has the highest level of initial interaction is FAB 5 (FAB Europe Central). The controlling FABs which generates the highest number of interactions in FAB 5 are FAB 1 (Baltic FAB), 4 (Danube FAB), and 5 (FAB Europe Central). In

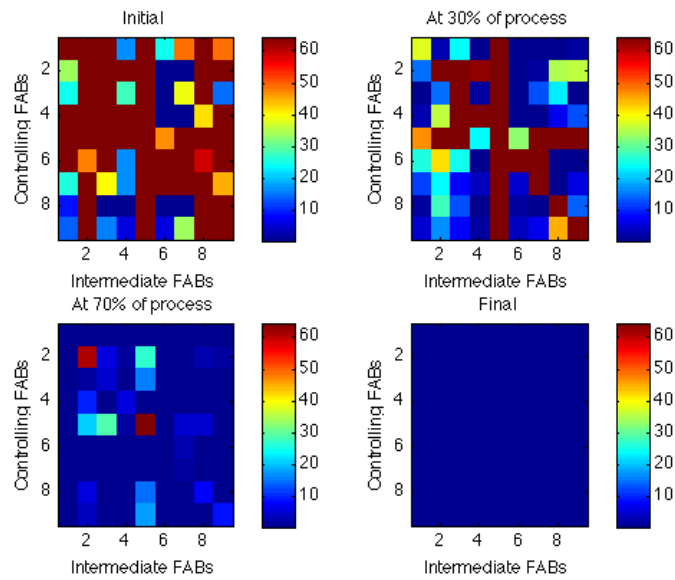


Fig. 12 Evolution of FAB-flight interaction matrix for 4,000 trajectories using distributed ATFM model.

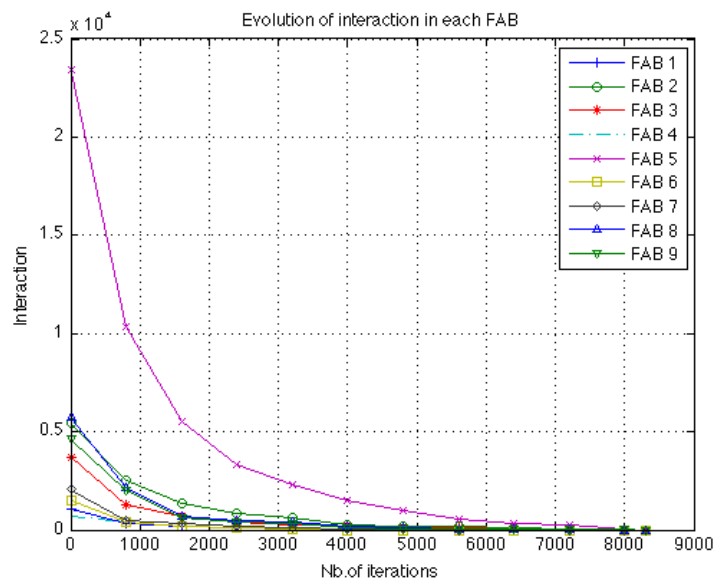


Fig. 13 Evolution of interaction in each FAB for 4,000 trajectories using distributed ATFM model.

the case of distributed model, the resolution algorithm tries to minimize the interaction by modifying flight plans of flights associated to FAB number 1, 4, and 5 before resolving interactions in other FABs.

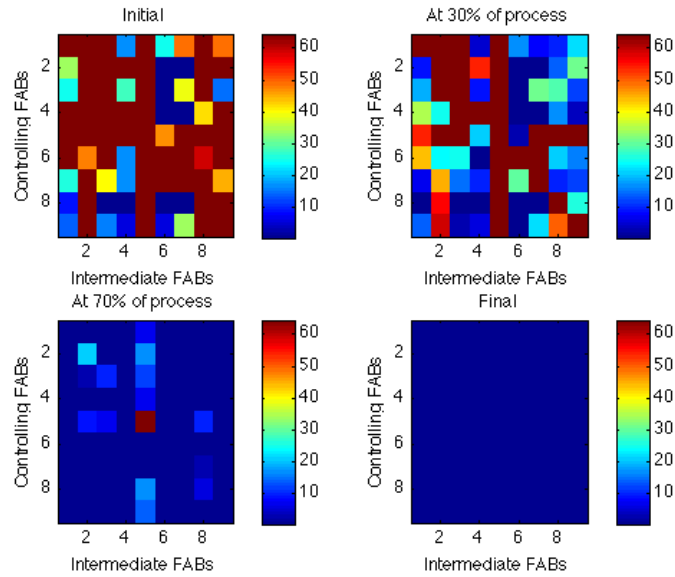


Fig. 14 Evolution of FAB-flight interaction matrix for 4,000 trajectories using centralized ATFM model.

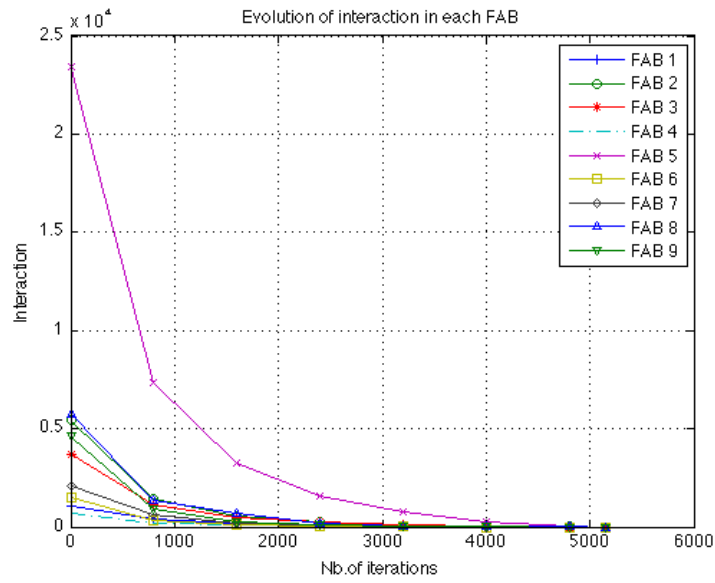


Fig. 15 Evolution of interaction in each FAB for 4,000 trajectories using centralized ATFM model.

Figure 13, and Figure 15 presents the resolution of interaction over number of iterations in each FAB during the optimization process for distributed and centralized models respectively. It can be seen that for with small traffic sample (4,000 flights) the centralized model converges faster to

interaction-free trajectories than distributed model. In distributed model, after 900 iteration, the number of interactions in FAB 5 were down to 1.05×10^4 as compared to distributed model, where the interactions in FAB 5 were reduced to 0.75×10^4 . This was expected, as centralized model do not takes into consideration the underlying FAB structure and the interactions among them for trajectory identification and resolution decision making. With large traffic sample (continental-scale traffic), a scenario this algorithm is specifically intended to address, the distributed algorithm converges faster than the centralized algorithm. In distributed model, the FAB-Flight Interaction matrix information processing overhead which dynamically updates the flight interaction information after each trajectory resolution results in better performance with large traffic sample.

B. Large Flight Sample: 26,000 flights

Results indicate that, in terms of iterations the distributed model takes 509,924 iterations as compared to 632,002 iterations in centralized model to solve 266,318 interactions with 26,122 flights. Distributed algorithm demonstrate approx 20% improvement in performance over centralized model with large traffic sample. In terms of CPU time, the difference between the two approaches is significant at 194 minutes (over 3 hours). This is a significant improvement over centralized approach as in medium term planning Air Navigation Service Providers project traffic loads based on airline schedules and flight plans for 2 to 12 Hours and takes in to consideration weather patterns.

Figure 16 and Figure 18 shows the color map of trajectory interaction and their subsequent resolution at the start of process, 30%, 70% and at the end of process. The start of the process shows that both the algorithms start with the same set of interactions. At the end of 30% of the process, it can be observed that the distributed model first resolves the high interactions caused by the controlling FAB (for example FAB 5) on other intermediate FABs. The color map also demonstrate the advantage of incorporating the Local Search (LS) strategy with neighbourhood search to facilitate the exploration of the problem search space initially (till 70% of process) and then exploitation of knowledge gained as demonstrated in quick resolution of most of the interaction during last stage of optimization process.

Figure 17 and Figure 19 shows the resolution of interaction over number of iterations in each FAB

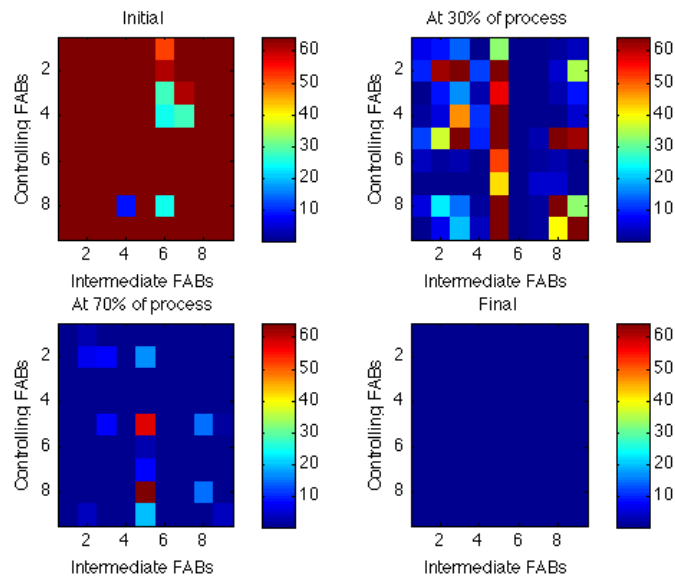


Fig. 16 Evolution of the FAB-flight interaction matrix for 26,122 trajectories using distributed ATFM model.

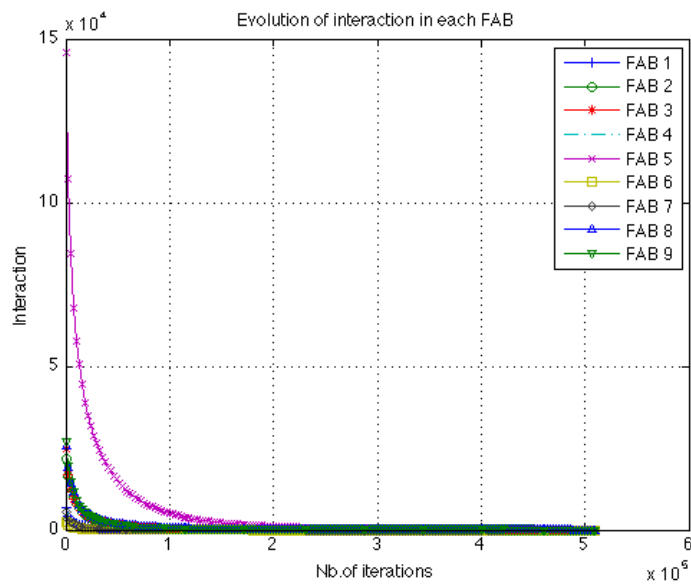


Fig. 17 Evolution of the interaction in each FAB for 26,122 trajectories using distributed ATFM model.

during the optimization process for distributed and centralized models respectively with large traffic sample. Though Centralized model initially able to resolve higher number of interaction but slows down as less flights with interacting trajectories are left. Whereas, in Distributed model the effective

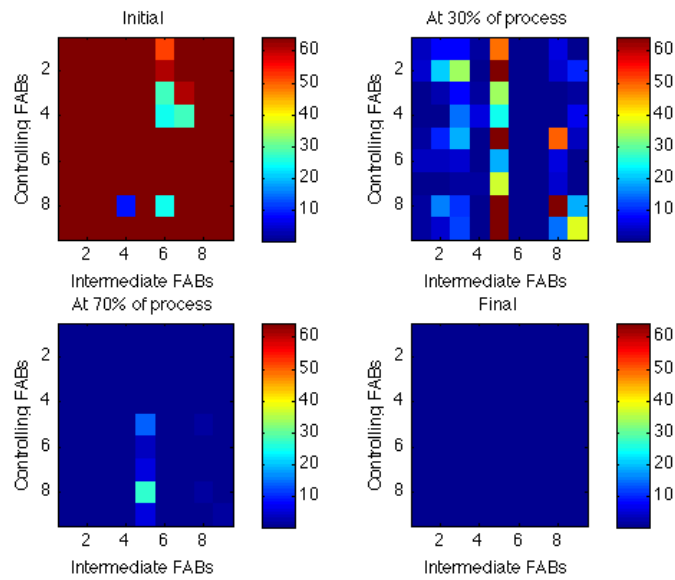


Fig. 18 Evolution of the FAB-flight interaction matrix for 26,122 trajectories using centralized ATFM model.

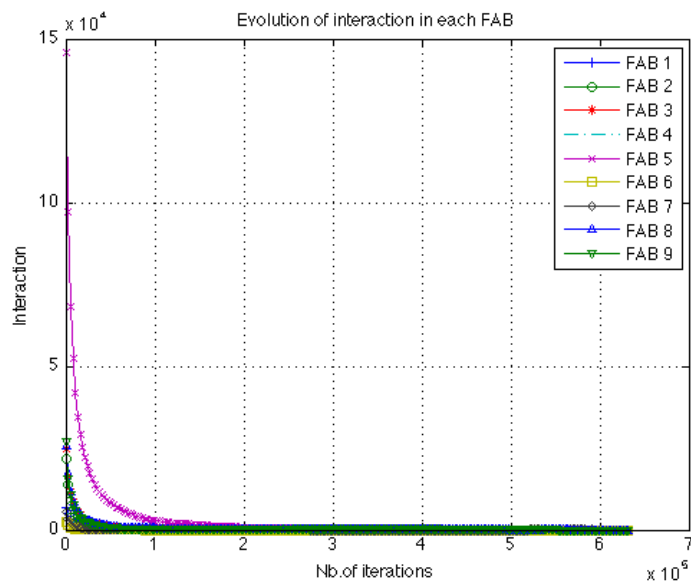


Fig. 19 Evolution of the interaction in each FAB for 26,122 trajectories using centralized ATFM model.

search mechanism of identification of FAB and interacting flights leads to better convergence.

VII. Conclusion

In this paper, we have presented a distributed air traffic flow management model, aiming at minimize interactions 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 with a population based search method based on Simulated annealing and local search.

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. 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 the cases.

In distributed model, the resolution algorithm tries to minimize the total interaction by modifying flight plans of flights associated with the FABs that generate high interaction in the overall air traffic. While, in the centralized model case, the resolution algorithm modifies flight plans of flights that involves high interaction, without taking into account the FAB-Flight interaction information. 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.

In the case of small data set, the computation time to reach interaction-free solution is not significantly different in the distributed and centralized model. However, when the traffic size increases, the distributed model converges to interaction-free solution 3 hours faster than the centralized model. This is a significant improvement over centralized model as in medium term planning Air Navigation Service Providers project traffic loads based on airline schedules and flight plans for 2 to 12 Hours and takes in to consideration weather patterns.

The proposed method of distributed air traffic flow management using FAB-Flight interaction matrix demonstrate that a trade-off between exploration and exfoliation in a population based search method for interaction resolution yields better results than the centralized traffic flow management approach. This trade-off is achieved using a heuristic algorithm consisting of a population based search method augmented with a local search and a neighbourhood search function. This approach dynamically assess trajectory interactions from origination and controlling FABs while resolving them. This ensures that the convergence time, specifically when it comes to plan continental scale

traffic, is small. 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.

Acknowledgement

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

VIII. References

- [1] International Civil Aviation Organization, Montreal, Quebec, Canada, *ICAO Doc 9971 AN/485, Manual on Collaborative Air Traffic Flow Management*, 2nd ed., 2014.
- [2] Button, K. and Neiva, R., "Single European Sky and the functional airspace blocks: Will they improve economic efficiency?" *Journal of Air Transport Management*, Vol. 33, 2013, pp. 73–80.
- [3] Dubot, T., Aubry, S., and Bedouet, J., "Building a smooth and dynamic opening scheme from graph partitioning-Exploring dynamic airspace configurations," *15th AIAA Aviation Technology, Integration, and Operations Conference, Dallas, Texas*.
- [4] "Single European Sky legislative package," Tech. Rep. Regulation (EC) No. 1070/2009, European Commission, Strasbourg, 2009.
- [5] "Eurocontrol Functional Airspace Block," <http://www.eurocontrol.int/dossiers/fabs>, Accessed: 2017-01-06.
- [6] Angenendt, A., "DFS on Central Flow Management Unit," *Skyway magazine Eurocontrol*, Vol. 9, No. 39, 2005.
- [7] EUROCONTROL, P., "Performance review report," *Performance Review Commission*, 2008.
- [8] Adler, N., Hanany, E., and Proost, S., "Managing European Air Traffic Control Provision," *Fourth SESAR Innovation Days, 25th–27th November 2014*, 2014.
- [9] Luppo, A., Argunov, G., Gorlenko, N., and Chaika, V., "Optimisation of the performance of the ATM network in Europe," *Proceedings of the National Aviation University*, , No. 2, 2014, pp. 37–43.
- [10] Modić, A., Steiner, S., and Mihetec, T., "Performance Scheme Implementation in Functional Airspace Block Central Europe," *22nd International Symposium on Electronics in Transport ISEP 2014: ITS for Seamless and Energy Smart Transport*, 2014.

- [11] Ichoua, S., "A scenario-based approach for the air traffic flow management problem with stochastic capacities," *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, Vol. 7, No. 8.
- [12] Holland, J. H., *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press, 1992.
- [13] Hansen, J. V., "Genetic search methods in air traffic control," *Computers & Operations Research*, Vol. 31, No. 3, 2004, pp. 445–459.
- [14] Cheng, V., Crawford, L., and Menon, P., "Air traffic control using genetic search techniques," *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, Vol. 1, 1999, pp. 249–254.
- [15] Du, W., Liang, B., Yan, G., Lordan, O., and Cao, X., "Identifying vital edges in Chinese air route network via memetic algorithm," *Chinese Journal of Aeronautics*, Vol. 30, No. 1, pp. 330–336.
- [16] Delahaye, D., Durand, N., Alliot, J.-M., and Schoenauer, M., "Genetic algorithms for air traffic control systems," *IFORS 1996, 14th Triennial Conference of the International Federation of Operational Research Societies*.
- [17] Chaimatanan, S., Delahaye, D., and Mongeau, M., "Large Scale 4D Trajectory Planning," *IEEE Computational Intelligence Magazine, Institute of Electrical and Electronics Engineers*, Vol. 9, 1989.
- [18] Bertsimas, D. and Gupta, S., "Fairness in air traffic flow management," *INFORMS Meeting, San Diego-CA, USA*, Vol. 54, p. 100.
- [19] Barnhart, C., Fearing, D., Odoni, A., and Vaze, V., "Demand and Capacity Management in Air Transportation," *EURO Journal on Transportation and Logistics*, Vol. 1, No. 1-2, 2012, pp. 135–155.
- [20] Glover, C. and Ball, M., "Stochastic optimization models for ground delay program planning with equity–efficiency tradeoffs," *Transportation Research Part C: Emerging Technologies*, Vol. 33, 2013, pp. 196–202.
- [21] Mukherjee, A., Grabbe, S., and Sridhar, B., "Predicting Ground Delay Program at an airport based on meteorological conditions," *14th AIAA Aviation Technology, Integration, and Operations Conference*.
- [22] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al., "Optimization by simulated annealing," *science*, Vol. 220, No. 4598, 1983, pp. 671–680.
- [23] Dreoj, J., Petrowski, A., Siarry, P., and Taillard, E., *Metaheuristics for hard optimization*, Springer, 2006.