



HAL
open science

Large Scale Adaptive 4D Trajectory Planning

Pierre Dieumegard, Supatcha Chaimatanan, Daniel Delahaye

► **To cite this version:**

Pierre Dieumegard, Supatcha Chaimatanan, Daniel Delahaye. Large Scale Adaptive 4D Trajectory Planning. DASC 2018, 37th AIAA/IEEE Digital Avionics Systems Conference, Sep 2018, Londres, United Kingdom. hal-01887286

HAL Id: hal-01887286

<https://enac.hal.science/hal-01887286>

Submitted on 3 Oct 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.

Large Scale Adaptive 4D Trajectory Planning

Pierre Dieumegard

Université de Toulouse ENAC
7 Avenue Edouard Belin
31055 Toulouse, France
pierre.dieumegard@outlook.fr

Supatcha Chaimatanan

Geo-Informatics and Space Development Agency
88 Moo 9, Thung Suk La
Si Racha, Chonburi, 20230 Thailand
supatcha@gistda.or.th

Daniel Delahaye

Université de Toulouse ENAC
7 Avenue Edouard Belin
31055 Toulouse, France
delahaye@recherche.enac.fr

Abstract—Global air-traffic demand is continuously increasing. To handle such a tremendous traffic volume while maintaining at least the same level of safety, a more efficient strategic trajectory planning is necessary. Static 4D trajectory planning with constant 4D segments, where aircraft have to stay all along their flights, ensures a strong predictability of traffic and may reduce congestion in airspace. The main limitation of this approach is linked to the 4D constraint associated to aircraft. As a matter of fact, each aircraft has to comply to this 4D segment to maintain separation from other aircraft, but this induces a real control of the engine in order to stay all the time in this 4D segment. This could result in extra fuel consumption and shorter engine life. In this work, we present an adaptive 4D strategic trajectory planning methodology which aims to minimize interaction between aircraft at the European-continent scale. The main purpose of this work is to associate to each aircraft a 4D bubble which is adapted to the current traffic situation. When aircraft are located in low density areas, the size of such bubbles can extend (with a maximum range of 20 minutes) and when aircraft enter high congestion areas, such bubbles can shrink until a minimum size of 2 minutes. The size of bubbles is then optimized according to the local density of aircraft. This adaptive process, avoid to constrain aircraft in 4D all along their trajectories. The proposed methodology separates aircraft by modifying their trajectories and departure times. This route/departure-time assignment problem is modeled as a mixed-integer optimization problem. Due to the very high combinatorics involved in the continent-scale context (involving more than 30,000 flights), we develop and implement a hybrid-metaheuristic optimization algorithm. This first optimization is done with a minimum bubble size of 2 minutes. A second optimization loop uses the solution produced by the first algorithm in order to optimally extend the size of the 4D bubbles along trajectories in order to minimize the time constraint of aircraft.

Index Terms—air traffic management, 4D aircraft trajectory, strategic planning, adaptive trajectory planning

I. INTRODUCTION

Nowadays, every commercial flight is performed under instrument flight rules¹. These air traffic regulations impose aircraft to be always separated horizontally by N_H and vertically by N_V . Separation norms are illustrated in Fig. 1. Aircraft are considered to be in conflict if the prescribed separation norms are violated.

In Europe, every (non-military) flight must submit its flight plan to the Central Flow Management Unit (CFMU) of Eurocontrol which checks the availability of the airspace. If

¹Instrument Flight Rules (IFR) is a set of regulations concerning aircraft operated when the pilot is unable to use visual reference navigation.

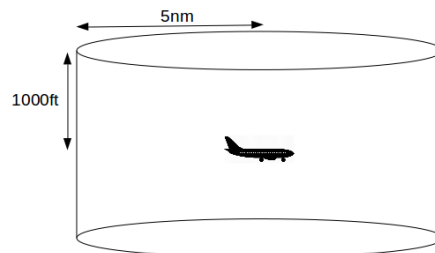


Figure 1: Cylindrical protection volume

the request is compatible with the capacity limit, the flight plan will be accepted. Otherwise, the CFMU will suggest alternative routes. This task is performed one year down to one week before real-time operations : it is called strategic planning. The CFMU aims at minimizing conflicts between aircraft. As air traffic grows exponentially, more conflicts appear at the same time. In order to deal with this increasing number of conflicts and to reduce controller's workload during the tactical phase, the Air Traffic Management (ATM) system has to rely on an efficient strategic 4D trajectory planning to assign a conflict-free 4D trajectory to each aircraft. The 4D trajectory planning allocates, to each aircraft, conflict-free 4D constraints in which aircraft has to stay all along its trajectory. Fig. 2 illustrates the 4D constraint, defined in 3D space and time domain. This will be called *4D bubble* in the sequel of this paper. The size of the 4D bubble defines a maximum allowable navigation error to remain in conflict-free condition. The size of the 4D bubbles are defined by the separation norms N_H in the horizontal dimension, N_V in the vertical dimension, and the *time separation* N_T , in the time dimension. However, as the number of conflicts decreases, the number of constraints soars. Indeed, to comply with its 4D constraint, the pilot must often act on the engines. This may cause extra fuel consumption, or worse a reduction in engines lifetime.

Since air traffic density is not the same all above Europe, aircraft may cross both high and low traffic density areas all along their flights. Why aircraft would have the same restrictions in dense areas as in empty areas ? In this work, we propose a methodology to adapt the strategic 4D trajectory planning according to the air traffic situation. The goal is to reduce constraints on trajectories in low density areas by extending the time constraint with a maximum range of 20

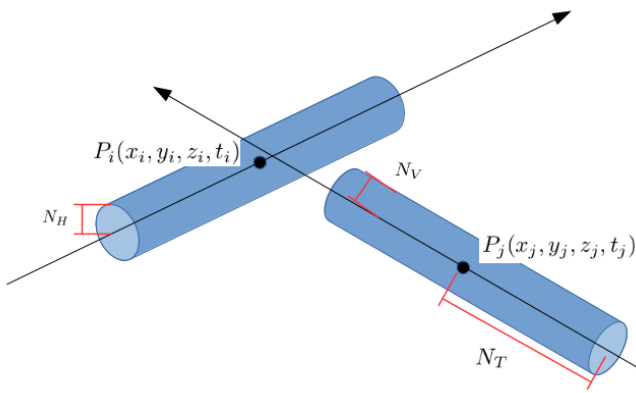


Figure 2: Conflict-free 4D bubbles around the nominal points P_i and P_j

minutes. In congested areas, the constraint might reduce to a minimum of 2 minutes. The result is an optimized 4D trajectory assigned to each aircraft depending on the air traffic environment.

The paper is organized as follow. Section II reviews classical decongestion methods and previous works on strategic 4D trajectory planning. Section III presents, first, the mathematical modelization and focuses on the existing solution of *static* 4D trajectory planning. Section IV introduces a methodology to adapt the 4D constraint to each aircraft taking into account the surrounding air traffic density. Section V presents and discusses numerical results of the application of this new concept at European-continent scale including statistics on engines effort computed under different wind conditions.

II. PREVIOUS RELATED WORKS

Here we present existing strategies in the literature considering air traffic decongestion and strategic 4D trajectory planning.

Historically, two main methods have been used for traffic decongestion. The first one is to adapt the airspace capacity to the demand at the controller scale. High density sectors are split into different sectors controlled by different controllers so as to reduce controller's workload. Therefore, they have less issues to cope with and conflicts are resolved more easily. This method is presented, for instance, in [1]. Unfortunately, a controller needs sufficient time and space to be able to manage traffic in his sector. It imposes the sector not to be too small. This is the main limitation to this decongestion strategy. The second approach consists in regulating the air traffic demand to the current capacity. It uses different methods like regulating speed, allocating delays or modifying routes of aircraft. Some of these strategies are considered, for instance, in [2].

One of the first project dealing with 4D trajectory has been launched by Eurocontrol, ARC2000. This work is detailed in [3]. It is an automation of tactical planning which modelizes aircraft trajectories in 4D pipes. In order to separate aircraft, pipes must not intersect. In this project, a greedy algorithm

is used: considering n pipes without any intersection, a $n+1^{\text{th}}$ pipe is added with an empty intersection with the n previous ones. This project has been abandoned because it relied on the policy of "*last arrived, last served*": the first trajectories were never questioned while the last ones could sustain large deviations. Another project, named SAINTEX, lead by CENA (Centre d'Etudes de la Navigation Aérienne) in the 1990s, used a similar algorithm. It is presented in [4].

Lead by the Single European Sky ATM Research (SESAR) programme, the european's ATM system has been modernizing for several years. This new ATM concept relies on Trajectory Based Operations (TBOs) and focuses on adapting the air traffic demand to the current air traffic capacity, providing a negotiated 4D trajectory to each aircraft. With the improvement of the surveillance, aircraft will be able to follow their 4D trajectory with high accuracy. Nevertheless, this solution needs a reliable 4D strategic deconfliction method that aims at generating conflict-free trajectories. Several projects have been working on this tough problem.

Launched in 2004, Innovative Future Air Transport System (IFATS) project imagines a new air transport system without any pilot. Every airline has to send the aircraft type, the departure time, departure and destination airports for each flight to a central system (like the CFMU) which will allocate a 4D pipe to each aircraft. If an aircraft must deviate from its pipe due to meteorological conditions, for example, it may ask for another pipe and it will be provided with one. More informations are available in [5].

In [6], Project 4 Dimension COntact-Guidance and Control (4DCO-GC) is introduced. As a part of SESAR, it aims to soften the environmental footprint of air transport by generating optimal 4D trajectories. Each aircraft is provided with a 4D trajectory which minimizes the environmental impact of the flight depending on ATM constraints.

In [7], the authors address a 4D trajectory deconfliction problem using only a ground-holding method. Conflicts are detected by pair-wise comparison, leading to high computation time in case of large-scale problem. Moreover, considering time uncertainties, significant delays must be allocated in order to solve all the conflicts. To increase the degree of freedom, the authors added an option to assign alternative flight level² in [8].

To solve potential conflicts between 4D trajectories while avoiding congested and adverse weather areas, a Light Propagation Algorithm (LPA) is introduced in [9]. Optimal trajectories are provided by a branch-and-bound (B&B) algorithm. This algorithm cannot solve all the conflicts considering uncertainties. Remaining conflicts are eliminated by adding supplemental time constraints.

However, considering the tremendous air traffic demand, none of these strategies is efficient enough. A more adapted and automated method is needed to both reduce airspace congestion and controller's workload. Nowadays, an efficient

²Flight Level (FL) is a pressure altitude, expressed in hundreds of feet, e.g. an altitude of 34,000 feet is referred to as FL 340.

strategic 4D trajectory deconfliction method exists and is considered in [10]. A hybrid-metaheuristic optimization algorithm is used to solve conflict between all involving trajectories. A conflict-free and robust 4D trajectory planning is finally produced within an acceptable computation time at strategic level. The work presented in this paper relies on the solution from the work presented in [10]. This initial solution will be called *static* because it provides each aircraft with the same constant-size 4D bubble. In this work, we propose a methodology to dynamically adapt the 4D bubble of each aircraft according to air traffic situation.

III. STATIC 4D TRAJECTORY PLANNING

This section presents, first, the mathematical model of the static 4D trajectory planning methodology and then presents the resolution algorithm based on hybrid metaheuristic optimization.

A. Mathematical model

Every flight is described by its own trajectory made of a set of successive 4D points (x, y, z, t) sampled with a constant sampling time. Each nominal trajectory is defined by a direct route from origin to destination airport. The given data are the following:

- A set of N discretized trajectories
- The sampling time step: t_s
- For each flight $i = 1, 2, \dots, N$:
 - The number of 4D discretized points: K_i
 - The initial planned departure time: $t_{i,0}$
 - The maximum allowed advance departure time shift: $\delta_a^i < 0$
 - The maximum allowed delay departure time shift: $\delta_d^i > 0$
 - The length of the initial planned en-route segment: $L_{i,0}$
 - The maximum route length extension coefficient: $0 \leq d_i \leq 1$
 - The initial planned flight level: $FL_{i,0}$
 - The maximum flight level shift: $l_{i,max}$
- The number of allowed virtual waypoint: M
- The discretization time step for the departure time shift interval: δ_s
- The interpolating sampling time step: t_{interp}

This strategy uses two ways to separate aircraft. In the time domain, it plays on the departure time shift, δ_i , associated with each flight i . In the 3D space, it relies on a vector, w_i , of virtual waypoint locations, $w_i := (w_i^1, w_i^2, \dots, w_i^M)$ and on a flight level shift, l_i , associated with each flight i . The decision variables are obviously:

$$\begin{aligned} \boldsymbol{\delta} &:= (\delta_1, \delta_2, \dots, \delta_N) \\ \mathbf{l} &:= (l_1, l_2, \dots, l_N) \\ \mathbf{w} &:= (w_1, w_2, \dots, w_N) \end{aligned}$$

The decision variables are gathered in one decision vector: $u := (\boldsymbol{\delta}, \mathbf{l}, \mathbf{w})$. These decision variables must satisfy the following constraints:

- **Allowed departure time shifts.** As it is not conceivable to delay or to advance departure times for too long, for each flight i , the departure time shift, δ_i , must stay in the allowed interval $[\delta_a^i, \delta_d^i]$, which is discretized. The set, Δ_i , of all possible departure time shifts is defined as follows:

$$\Delta_i := \{\delta_a^i, \dots, -\delta_s, 0, \delta_s, \dots, \delta_d^i\} \quad (1)$$

- **Allowed flight level changes.** The cruise flight level shift, l_i , must also be limited. The set, ΔFL_i , of all possible flight level changes, for each flight i , is:

$$\Delta FL_i := \{-l_{i,max}, \dots, -1, 0, 1, \dots, l_{i,max}\} \quad (2)$$

- **Maximal route extension.** The alternative trajectory cannot induce too much route extension because it would increase fuel consumption. Then, the normalized length of the alternative route, $L_i(w_i)$, for each flight i must satisfy:

$$L_i(w_i) \leq (1 + d_i) \quad (3)$$

- **Allowed waypoint locations.** Two sets W_{ix}^m and W_{iy}^m are defined to be compliant with the previous constraint. The m^{th} virtual waypoint of trajectory i , w_i^m , must be in these sets respectively for the longitudinal and the lateral locations.

The methodology aims to minimize interaction between trajectories. The word *interaction* corresponds to a situation which occurs at strategic level, when more than one trajectory compete for the same space at the same period of time. It is different from a conflict since at strategic level, we focus on separating trajectories rather than solving conflicts, which are just a violation of the minimum separation norms during the tactical phase. Supplemental conditions are taken into account at strategic level such as time separation. An *interaction* at the point $P_{i,k}$ is defined to be the sum of all the violation of the 4D bubble involving $P_{i,k}$. It is denoted $\Phi_{i,k}(u)$:

$$\Phi_{i,k}(u) := \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{l=1}^{K_j} C(P_{i,k}(u_i), P_{j,l}(u_j)), \quad (4)$$

where $C(P, Q) := \begin{cases} 1 & \text{if point } P \text{ and } Q \text{ are in conflict} \\ 0 & \text{otherwise} \end{cases}$

Finally, interaction between trajectories, Φ_{tot} , for the whole traffic is defined as:

$$\Phi_{tot}(u) := \sum_{i=1}^N \Phi_i(u) = \sum_{i=1}^N \sum_{k=1}^{K_i} \Phi_{i,k}(u) \quad (5)$$

The methodology can be formulated as an optimization problem aiming to determine values for the decision variables δ_i , l_i and w_i for each flight $i = 1, 2, \dots, N$ in order to minimize, $\Phi_{tot}(u)$, interaction between the N involved flights. It is formulated as follows:

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

subject to,

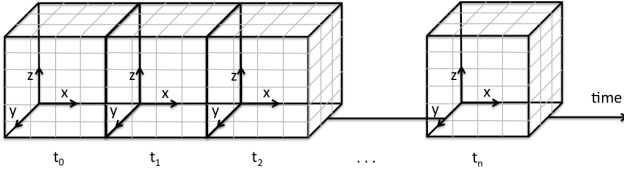


Figure 3: 4D (space - time) grid

$$\begin{aligned} \delta_i &\in \Delta_i, \text{ for all } i = 1, \dots, N \\ l_i &\in \Delta FL_i, \text{ for all } i = 1, \dots, N \\ w_i^m &\in W_{ix}^m \times W_{iy}^m, \text{ for all } i = 1, \dots, N, m = 1, \dots, M \end{aligned}$$

Computing the objective function is up to check interaction between the given trajectories. To avoid pair-wise comparison, trajectories have been classified in a 4D grid represented in Fig. 3. The size of each cell in the x , y and z directions is given by the minimum separation norms, N_H and N_V . Each 4D point (x, y, z, t) of a trajectory is located in a cell of that grid with the following coordinates : (I_x, I_y, I_z, I_t) . A hash table, which is a data structure that maps keys to values or entries, is built to represent the 4D grid. Each cell of the grid is an entry in the hash table. The key associated to the cell (I_x, I_y, I_z, I_t) is defined by:

$$key = N_y * N_x * N_t * I_z + N_x * N_t * I_y + N_t * I_x + I_t \quad (6)$$

The corresponding value for this key is an array of information about all the points located in the corresponding cell in the 4D grid. According to the definition of the grid and the minimum separation norms, a conflict between two aircrafts can only appear if both are in the same cell or in adjacent cells. To find whether or not a trajectory is in interaction with others, for each point of this trajectory, only neighbor points, which are in the same cell or in the surrounding cells, of other trajectories must be checked. Thus, the number of calculation is significantly smaller than in a pair-wise comparison where all pairs would have been checked.

However, if the sampling time between two consecutive points is too large, there could be undetected interaction during one time step. That is why the sampling time, t_s , must be small enough not to underestimate the number of interaction. Unfortunately, this significantly increases computation time and memory. The solution is to use an inner-loop algorithm called *interp*. It detects any interaction between two sampled times by interpolating aircraft positions with a sufficiently small time step t_{interp} .

B. Hybrid-metaheuristic optimization algorithm

To solve the strategic static 4D trajectory planning problem, a hybrid-metaheuristic optimization algorithm, introduced in [10], is presented in this subsection. The algorithm combines a simulated annealing (SA) with a simple local search algorithm. Since SA is able to escape from local traps, it will be responsible of the *diversification* of the search, while the local search algorithm will *intensify* the search in a region provided by

SA. The local search relies on an Iterative Improvement Local Search (IILS) algorithm that allows only strict improvement of the objective function value.

First, the 4D grid is filled with all initial nominal trajectories. Interaction are checked thanks to the method described in the previous subsection. Trajectories competing for the same space at the same time should be modified. To avoid adjusting trajectories involved in few interaction, we set a threshold value of interaction, denoted Φ_τ , such that a flight $i \in \llbracket 1, N \rrbracket$, would be modified only if

$$\Phi_i(u) \geq \Phi_\tau. \quad (7)$$

At each iteration of the hybrid algorithm, one flight is randomly chosen among $\{i = 1, \dots, N : \Phi_i \geq \Phi_\tau\}$. Then, the algorithm randomly determines whether to perform a classical SA step, or to start the IILS, or to execute both of them successively. It depends on the temperature T of SA, initialized at T_0 thanks to the initial probability of accepting a degrading solution, τ_0 :

- The probability of performing a SA step, P_{SA} , is:

$$P_{SA}(T) := P_{SA,min} + (P_{SA,max} - P_{SA,min}) \cdot \frac{T_0 - T}{T_0} \quad (8)$$

- The probability of running the IILS, P_{loc} , is:

$$P_{loc}(T) := P_{loc,min} + (P_{loc,max} - P_{loc,min}) \cdot \frac{T_0 - T}{T_0} \quad (9)$$

- The probability to carry out both SA and IILS successively, P_{SL} , is:

$$P_{SL}(T) := 1 - (P_{SA}(T) + P_{loc}(T)) \quad (10)$$

A neighborhood function is in charge of generating modified trajectories. It determines randomly whether to change the location of waypoint or the departure time. The neighborhood function is the same for both SA and IILS. Considering the current solution $\Phi_{tot,c}$, the new generated solution $\Phi_{tot,n}$ is accepted or not under SA acceptance criteria:

- if $\Phi_{tot,n} < \Phi_{tot,c}$, $\Phi_{tot,c} = \Phi_{tot,n}$
- else, $\Phi_{tot,c} = \Phi_{tot,n}$ with a probability $e^{\frac{\Phi_{tot,c} - \Phi_{tot,n}}{T}}$

At each temperature level, we perform N_I iterations of the hybrid algorithm. The chosen cooling scheme is a geometric cooling:

$$T_{i+1} = \beta * T_i, \quad (11)$$

where β is the temperature reduction coefficient. The algorithm operates while $T \geq T_{final}$, where T_{final} is the final temperature set by the user. For the IILS algorithm, at each of the N_{loc} iterations, a new solution is generated by the neighborhood function. It is accepted only if it strictly improves the objective function value. The whole hybrid algorithm is presented in Fig. 4.

This algorithm provides each aircraft with a conflict-free trajectory taking into account uncertainties, especially in the time domain. However, every aircraft are provided with the same 4D bubble, N_T set to be a constant of 1 minute. The

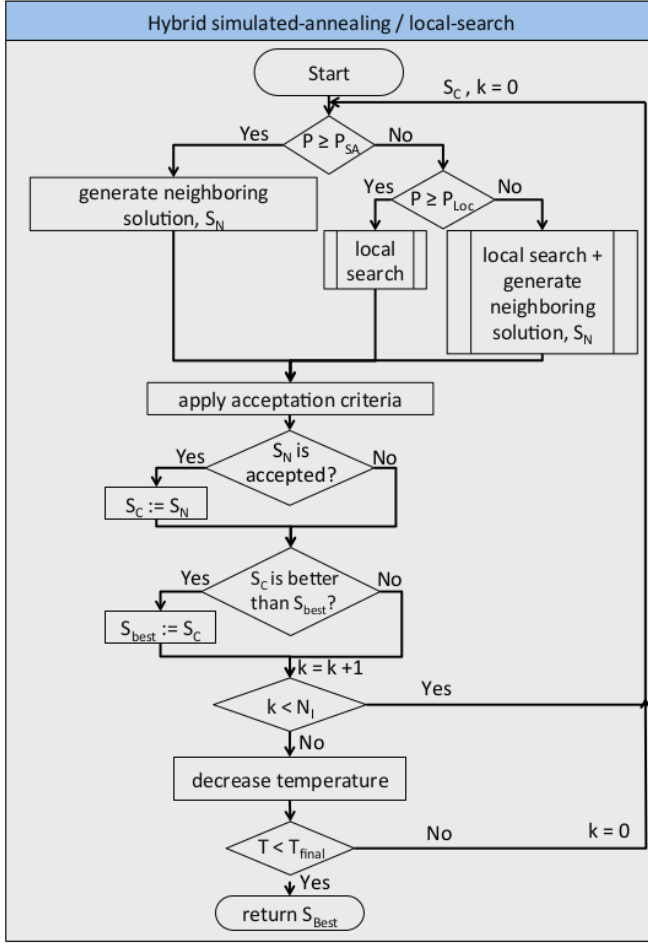


Figure 4: Hybrid simulated annealing and local search algorithm

extension of these bubbles, depending on the local air traffic density, is presented in the following section.

IV. ADAPTIVE 4D TRAJECTORY PLANNING

This section introduces the proposed method to optimally adapt the 4D trajectory planning to the air traffic situation.

A. Mathematical model

We start from the initial solution introduced in the previous section with the minimum time separation N_T set to be a constant of 1 minute. The initial conflict-free 4D trajectories are stored in the hash table. Consider trajectory $i \in \llbracket 1, N \rrbracket$, each point $P_{i,k}$, $k \in \llbracket 1, K_i \rrbracket$, of this trajectory has the following coordinates $(x_{i,k}, y_{i,k}, z_{i,k}, t_{i,k})$ and can be accessed in the hash table with its key computed according to (6). We choose to focus on the temporal aspect of the 4D bubble. This method aims to get a time interval around $t_{i,k}$ in which the aircraft may reach location $(x_{i,k}, y_{i,k}, z_{i,k})$, while maintaining conflict-free condition. This time extension interval will be denoted $[\delta_{i,k}^e, \delta_{i,k}^l]$ where the boundaries as defined as follows:

$$\delta_{i,k}^e := -t_s * d_{i,k}^{early}, \quad (12)$$

with $d_{i,k}^{early}$ the number of allowed time steps for aircraft i to be early at point $P_{i,k}$, and

$$\delta_{i,k}^l := t_s * d_{i,k}^{late}, \quad (13)$$

with $d_{i,k}^{late}$ the number of allowed time steps for aircraft i to be late at point $P_{i,k}$.

For the convenience of the computation, the decision variables $d_{i,k}^{early}$ and $d_{i,k}^{late}$ can only take discrete values in $\llbracket 0, b_{max} \rrbracket$. A new decision vector is introduced: $v = (\delta, l, w, d)$, where the new decision variable, d , is defined by:

$$d := (d_1, d_2, \dots, d_N), \quad d_i = (d_{i,1}, d_{i,2}, \dots, d_{i,K_i}), \\ d_{i,k} = (d_{i,k}^{early}, d_{i,k}^{late})$$

The algorithm aims to maximally extend 4D bubbles, while maintaining conflict-free condition. The problem is formulated as follows:

$$\max_{v=(\delta, l, w, d)} \sum_{i=1}^N \sum_{k=1}^{K_i} (d_{i,k}^{early} + d_{i,k}^{late})$$

subject to,

$$d_{i,k}^{early} \in \llbracket 0, b_{max} \rrbracket, \quad \text{for all } i = 1, \dots, N, k = 1, \dots, K_i \\ d_{i,k}^{late} \in \llbracket 0, b_{max} \rrbracket, \quad \text{for all } i = 1, \dots, N, k = 1, \dots, K_i \\ \Phi_{tot}(v) = \Phi_{tot}(u)$$

B. 4D Bubble extension

This subsection details the proposed algorithm to optimally extend the 4D bubble for each aircraft depending on the surrounding air traffic density.

As we want to focus on the time aspect of the 4D bubble, the algorithm needs to iterate the bubble computation on time $t \in [t_{min}, t_{max}]$. By looking through the hash table, we get *pointsToDeal*: the array of active 4D points at time $t_{i,k}$. For each point $P_{i,k}$ in *pointsToDeal*, the 4D bubble is built around the current time $t_{i,k}$.

We introduce the *bubble width*, which is the time width of the bubble, i.e., the duration $\delta_{i,k}^l - \delta_{i,k}^e$. Practically, the bubble width can vary between 0 seconds in case of a non-resolved conflict especially in high density areas and $b_{t,max}$ in low traffic areas. The bubble width is initially reduced to zero. We will try to extend it step by step. We set a limit, b_{max} , for the extension of the 4D bubble. Thanks to this limit, aircraft cannot deviate too far from their initial negotiated trajectories.

If time $t_{i,k}$ is not the first, respectively the last time sample, we introduce two virtual points:

- *pointEarly*. It has the same location as point $P_{i,k}$, but at time $t_{i,k-1}$.
- *pointLate*. It has the same location as point $P_{i,k}$, but at time $t_{i,k+1}$.

We suppose aircraft i to be at *pointEarly*, respectively *pointLate* and we check whether or not any separation norm with another aircraft would be violated. To detect potential

conflicts with other trajectories, we use the same principle than for static 4D trajectory planning. Trajectories which are in conflict with trajectory i are located in the same cell or in the direct neighbor cells in the 4D grid. Thus, to get a list of neighbors of $pointEarly$, respectively $pointLate$, we look through all their surrounding nine cells. Once we get the list, we must check among the neighbors whether or not there are in conflict with the current hypothetic $pointEarly$, respectively $pointLate$. If there is at least one conflict, we stop the iteration and we remember the flag limit $d_{i,k}^{early}$, respectively $d_{i,k}^{late}$, while if not, hypothetic position is validated. In this second case, flag $d_{i,k}^{early}$, respectively $d_{i,k}^{late}$ is incremented. We *activate* also point $P_{i,k+1}$, respectively $P_{i,k-1}$ in the hash table. Actually, aircraft i may occupy these positions at time $t_{i,k}$ so we need to remember that these positions are *activated* when we are looking for neighbors of other aircraft. At every new time step potential positions are *desactivated* and the computation is performed again.

This process is performed for every point in $pointsToDeal$ by incrementing $d_{i,k}^{early}$, respectively $d_{i,k}^{late}$ one by one. We iterate it while there are no conflict and while we have not reached the limitation b_{max} , still at time $t_{i,k}$, as shows Algorithm 1.

Algorithm 1 4D bubble computation

```

for  $t = t_{min}$  to  $t_{max}$  do
  Get  $pointsToDeal$ ;
  for  $P_{i,k}$  in  $pointsToDeal$  do
    Initialize  $d_{i,k}^{early} := d_{i,k}^{late} := 0$ ;
    Initialize  $tEarly_{i,k} := t - 1$ ;
    Initialize  $tLate_{i,k} := t + 1$ ;
  end for
  while  $pointsToDeal$  not empty do
    for  $P_{i,k}$  in  $pointsToDeal$  do
      Compute  $pointEarly(tEarly)$ ,  $pointLate(tLate)$ ;
      if  $pointEarly$  and  $pointLate$  not conflict-free then
        Remove  $P_{i,k}$ ;
      else
        if  $pointEarly$  is conflict free then
          Increment  $d_{i,k}^{early}$ ;
          Decrement  $tEarly$ ;
        end if
        if  $pointLate$  is conflict free then
          Increment  $d_{i,k}^{late}$ ;
          Increment  $tLate$ ;
        end if
      end if
    end for
  end while
end for

```

Finally, we get an adapted interval, $[d_{i,k}^{early}, d_{i,k}^{late}]$. It is converted into a continuous time shift range $[\delta_{i,k}^e, \delta_{i,k}^l]$, calculated thanks to (12) and (13). This data mean that aircraft i , which normally reach location $(x_{i,k}, y_{i,k}, z_{i,k})$ at time $t_{i,k}$, is allowed

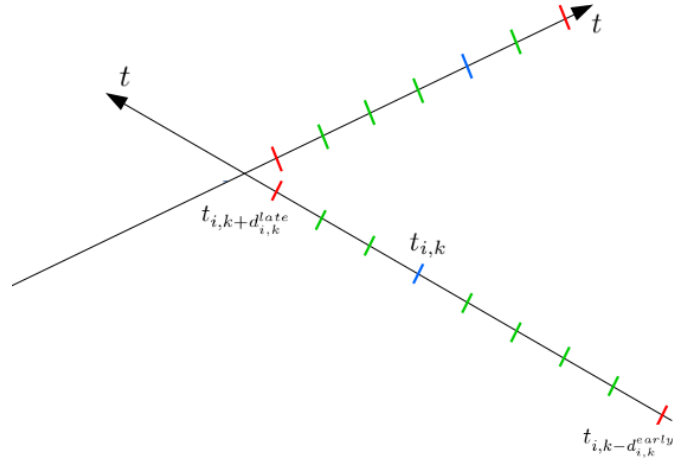


Figure 5: 4D bubbles computing

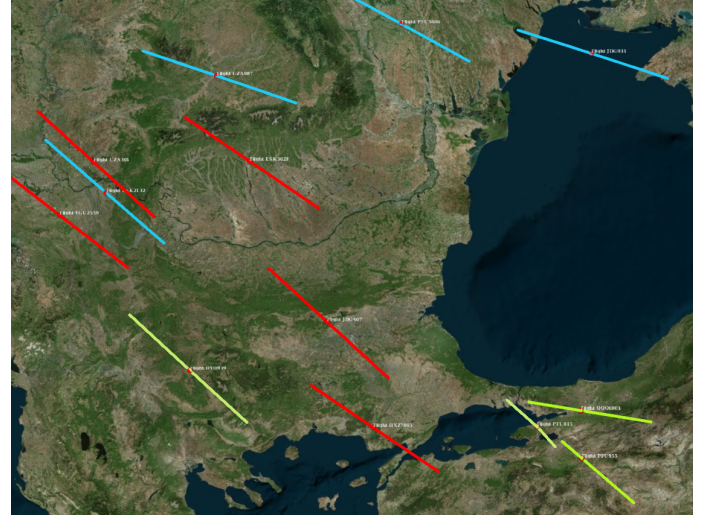


Figure 6: Visualization of 4D bubbles

to reach this position with a time shift $\delta_{t,i,k} \in [\delta_{i,k}^e, \delta_{i,k}^l]$. The results of the process are illustrated in Fig. 5. Green plots are allowed times for the aircraft to reach the nominal location, and red ones are limit points of the 4D bubble.

Finally, as the bubble is computed for each point $P_{i,k}$ of trajectory i , we have a moving bubble in which aircraft $i \in \llbracket 1, N \rrbracket$ must stay all along the flight. Part of the results are represented in Fig. 6. At the bottom-right corner we can see 3 aircrafts for which 4D bubbles are not extended to the maximum due to high air traffic density in their sector while almost all others are extended to the maximum bubble width, $b_{t,max}$.

V. RESULTS

A. Continental-size en-route air traffic

The proposed algorithm solves a European-continental scale en-route air traffic problem. The data set is a full day of air-

Table I: User-defined parameter values specifying the continental-size air traffic problem

Parameter	Value
Sampling time step, t_s	20 seconds
Discretization time step for 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, d_i	0.25
Maximum number of waypoints, M	3
Maximum bubble extension step, b_{max}	30
Maximum bubble width, $b_{t,max} := 2 * b_{max} * t_s$	20 minutes
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 ILS step, $P_{Loc,min}$	0.4
Maximum probability to perform ILS step, $P_{Loc,max}$	0.6
Number of iterations at each temperature step, N_I	2,000
Number of iterations of the ILS step, N_{Loc}	5
Initial rate of accepting degrading solution, τ_0	0.3
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

traffic over the European airspace on July 1st, 2011. It consists of $N = 30,695$ trajectories simulated with optimal vertical profiles and with direct route. We address this problem instance on a Unix platform with a 2,4 GHz processor with 8.0 Gb RAM (personal computer). User-defined parameter values that specify the problem instance under consideration are given in Table I.

To give an idea about the complexity of the computation of the objective function, when using the sampling time step value $t_s = 20$ seconds, the initial N trajectories are discretized into 7,932,712 sampled 4D points. The initial trajectory set, taking into account a probabilistic-type time uncertainty model, involves $\Phi_{tot} = 1,128,283$ total interaction between trajectories. For this problem instance, the possible values of the decision variables are the following:

- $2MN = 184,170$ (continuous) virtual waypoint variables (component of vector \mathbf{w});
 - $N = 30,695$ (discrete) departure time shift variables (component of vector δ), each involving $(\frac{2\delta}{\delta_s} + 1)^N = 721^{30,965}$ possible values;
- for total 214,865 decision variables.

B. Numerical results

The hybrid-metaheuristic optimization algorithm leads to an almost interaction-free solution for this continent-scale problem instance in a computation time which is acceptable considering a strategic planning application. Conflict-free 4D trajectory planning is provided without any uncertainties in 478.1 minutes. To compute constant 4D bubbles with $N_T = 1$ minute, parameters are the same than in Table I except the number of iterations at each temperature step, N_I , which is set to 4,500. Under these conditions, 92.5% of initial interaction are solved. Remaining interaction can be solved during pre-tactical and tactical phase by lowering the level of uncertainty. This allows to significantly reduce controller's workload during the tactical phase. Numerical results for the problem with and without uncertainties are presented in Table II.

Table II: Numerical results of the static strategic planning algorithm with and without time uncertainties

N_T (seconds)	Initial Φ_{tot}	Final Φ_{tot}	Solved interactions	Computation time (minutes)
0	235,632	0	100%	478.1
120	1,128,283	85,185	92.5%	2,340.7

However, this solution provides each aircraft with a constant 2 minutes wide 4D bubble whatever the air traffic situation. The 4D bubble extension algorithm, presented in subsection IV.B, yields extended 4D bubbles for each aircraft and at each position suited to the surrounding air traffic density. For this continent-scale problem instance, the algorithm is able to extend 7,116,277 4D bubbles among the 7,932,712 sample 4D points in 2,862.1 minutes. The computation time is acceptable since we work for strategic planning. 89.7% of the 4D bubbles have been lengthened, especially 27.7% have been extended to the maximum range, where $N_T = b_{t,max}$. These substantial results show the importance of this second algorithm. Indeed, even if they must still stay inside their 4D bubbles, aircraft will have more freedom along their trajectories since 4D bubbles have been extended. For instance, in case of an unpredictable meteorological event, aircraft could let them drift inside their 4D bubbles without acting on engines as long as they are not too close to the limits. One of the best benefit is the reduction in the number of actions on engines, which helps to reduce maintenance and to prolong their lives. Some statistics about engines action under different wind conditions are presented in the following subsection.

C. Engines statistics

Every aircraft has been provided with a 4D bubble from which it must not go out during the whole flight. This is a non-negotiable constraint if we want to keep the highest safety level otherwise it could reveal some conflicts between trajectories. Navigation is now reduced at staying in the allocated 4D bubble all along the trajectory. As aircraft often face meteorological issues like unpredictable wind conditions, the pilot may act on the engines in order to remain inside its 4D bubble. The more pilots change engine regime, the more fuel consumption increases, and the shorter engines live. Reducing the impact of the pilot on the engines: this is the purpose of this paper. To check whether or not this solution is acceptable and useful, we have computed some statistics on the pilot action on engines.

First, if meteorological conditions are perfect: no wind or exactly the same as forecasted, aircraft will always stay in the center of their 4D bubbles, so pilots do not need to act differently from the plans. On the contrary, if, for instance, unexpected wind conditions occur, it will have an impact on aircraft's speed: it will be faster with tailwind or slower with headwind. Obviously, Wind Speed (WS) impacts aircraft's Ground Speed (GS) while True Air Speed (TAS) is absolute, as shown in Fig. 7.

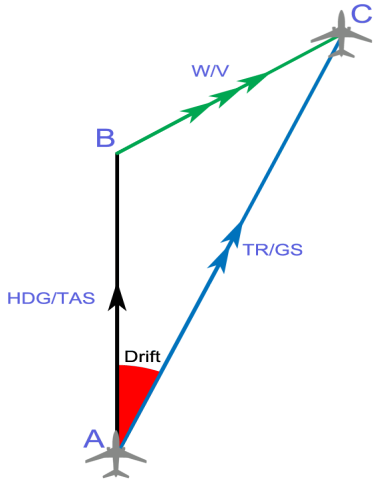


Figure 7: Wind triangle

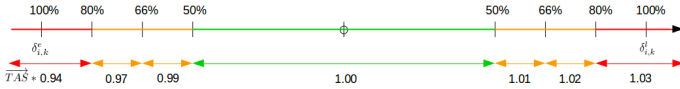


Figure 8: Engines action decision rules

$$\overrightarrow{GS} = \overrightarrow{TAS} * + \overrightarrow{WS} \quad (14)$$

Then, aircraft will drift inside its 4D bubble and pilot may need to perform supplemental actions on engines so as to stay inside the bubble, for safety purpose. As a matter of fact, when the wind is not too strong and the 4D bubble is large enough, the pilot can let the aircraft drift. He must act only if it threatens to leave its 4D bubble. Pilot's action is reflected in a TAS change. In order to know when to act, the pilot must be aware of its location inside its 4D bubble. Let $[\delta_{i,k}^e, \delta_{i,k}^l]$ be the extended 4D bubble in the time domain, allocated to aircraft i at the point $P_{i,k}$ and let $\delta_{t,i,k}$ be the deviation from the nominal planned time. We consider that if aircraft are early, $\delta_{t,i,k} < 0$, pilots act according to the following rules:

$$\begin{aligned} \delta_{t,i,k} > \delta_{i,k}^e/2 &\Rightarrow \text{None} \\ \delta_{t,i,k} \in [\delta_{i,k}^e * 2/3, \delta_{i,k}^e/2] &\Rightarrow \overrightarrow{TAS}_n = 0.99 * \overrightarrow{TAS}_c \\ \delta_{t,i,k} \in [\delta_{i,k}^e * 5/6, \delta_{i,k}^e * 2/3] &\Rightarrow \overrightarrow{TAS}_n = 0.97 * \overrightarrow{TAS}_c \\ \delta_{t,i,k} < \delta_{i,k}^e * 5/6 &\Rightarrow \overrightarrow{TAS}_n = 0.94 * \overrightarrow{TAS}_c \end{aligned}$$

Otherwise if aircraft are late, $\delta_{t,i,k} > 0$, pilots must apply:

$$\begin{aligned} \delta_{t,i,k} < \delta_{i,k}^l/2 &\Rightarrow \text{None} \\ \delta_{t,i,k} \in [\delta_{i,k}^l/2, \delta_{i,k}^l * 2/3] &\Rightarrow \overrightarrow{TAS}_n = 1.01 * \overrightarrow{TAS}_c \\ \delta_{t,i,k} \in [\delta_{i,k}^l * 2/3, \delta_{i,k}^l * 5/6] &\Rightarrow \overrightarrow{TAS}_n = 1.02 * \overrightarrow{TAS}_c \\ \delta_{t,i,k} > \delta_{i,k}^l * 5/6 &\Rightarrow \overrightarrow{TAS}_n = 1.03 * \overrightarrow{TAS}_c \end{aligned}$$

The decision rules are illustrated in Fig. 8.

Then, we consider that when a decision has been taken, the pilot maintains it until the aircraft comes back in the center of the 4D bubble: $\delta_{t,i,k} \in [\delta_{i,k}^e/2, \delta_{i,k}^l/2]$ or until there is a stronger modification to do.

Table III: User-defined parameters values specifying the wind field

Parameter	Value
Number of initial wind vectors, N_w	3
Maximum wind speed, w_{max}^s	60 knots
Standard deviation of wind direction, σ^d	0.5
Standard deviation of wind speed, σ^s	0.3

Table IV: Numerical results of the statistics on engine's effort under different wind conditions

Bubble width (seconds)	Wind scenario	Modified engine regime points	Average engine regime
120	No wind	0%	1
120	Wind	3.80%	0.9992
1,200	No wind	0%	1
1,200	Wind	1.66%	0.9994

As we act in strategic phase, we do not have any information on weather forecasts. Therefore, to compute the statistics of pilot action on the engines, the wind field is randomly simulated. We consider only longitudinal and lateral dimensions, neglecting the vertical dimension. Practically, we generate a N_w wind vectors basis at t_{min} , provided with a random direction, $w_{i,t_0}^d \in [0, 360]^\circ$ and a random speed, $w_{i,t_0}^s \in [0, w_{max}^s]$ kts, for $i = 1, \dots, N_w$. Then, the wind is computed at all positions thanks to an Inverse Distance Weighting (IDW) interpolation method, described in [11]. For the time dimension, we consider that wind evolves according to the following dynamics:

- $w_{i,t_{k+1}}^d$ is generated by a Gaussian normal distribution $\mathcal{N}(w_{i,t_k}^d, \sigma^d)$
- $w_{i,t_{k+1}}^s$ is generated by a Gaussian normal distribution $\mathcal{N}(w_{i,t_k}^s, \sigma^s)$

After the application of the wind field to the whole traffic, the algorithm counts how many times pilot must change engines regime and the average effort of engines along their trajectories. The average engines effort is initialized to 1.0. Finally, for each trajectory $i \in \llbracket 1, N \rrbracket$, we get the number of points where engines regime has been changed and the average effort of engines during the flight. Wind field parameters used for this problem instance are given in Table III. The calculation is performed 10 times and average numerical results are presented in Table IV.

The results show that enlarging 4D bubbles from $N_T = 1$ minute to a maximum size of $N_T = \frac{b_{t,max}}{2}$ yields a reduction in fuel consumption. As we expected, the main reduction is in the number of action on engines from pilots. With a 2 minutes wide 4D bubble, 3.8% of the points are involved in a modification of engines regime. After the extension of 4D bubbles, only 1.66% of the points are concerned. This will definitely reduce engines maintenance frequency and will increase engines durability.

VI. CONCLUSION

This paper has introduced a new approach of strategic 4D trajectory planning. A methodology to adapt the planning for every aircraft has been described. First, static planning has been computed with a constant 2 minutes wide 4D bubble for each aircraft. Then, 4D bubbles has been extended at european-continent scale depending on the air traffic situation. It inputs flexibility for the pilot in his navigation since he may be up to 10 minutes early or late at a point in low density traffic areas. Statistics have been computed under different random wind conditions. It shows how much 4D bubbles extension allows to reduce constraints for pilots while keeping the same level of safety. It presents benefits also for airlines like fuel savings or lengthened engine lifetime.

REFERENCES

- [1] H. D. Sherali and J. M. Hill, "Configuration of airspace sectors for balancing air traffic controller workload," *Annals of Operations Research*, vol. 203, pp. 1–29, 2011.
- [2] D. Bertsimas and S. Patterson, "The air traffic flow management problem with en-route capacities," *Operations Research*, vol. 46, no. 3, pp. 406–422, 1998.
- [3] X. Fron, B. Maudry, and J.-C. Tumelin, "Arc 2000: Automatic radar control," Eurocontrol, Tech. Rep., 1993.
- [4] L. Angerand and H. LeJeannic, "Bilan du projet SAINTEX," Centre d'études de la navigation aérienne, Tech. Rep., 1992.
- [5] H. H. Toebben, C. L. Tallec, A. Joulia, J. Speidel, and C. Edinger, "Innovative future air transport system: Simulation of a fully automated ATS," in *26th International Congress of the Aeronautical Sciences*, Anchorage, Alaska, 2008.
- [6] A. Joulia and C. L. Tallec, "Aircraft 4D contract based operation: The 4DCO-GC project," in *28th International Congress of the Aeronautical Sciences*, Brisbane, Australia, 2012.
- [7] N. Barnier and C. Allignol, "4D-trajectory deconfliction through departure time adjustment," in *ATM 2009, the 8th USA/Europe Air Traffic Management Research and Development Seminar*, Napa (California), 2009.
- [8] —, "Combining flight level allocation with ground holding to optimize 4D-deconfliction," in *ATM 2011, the 9th USA/Europe Air Traffic Management Research and Development Seminar*, Berlin (Germany), 2011.
- [9] N. E. Dougui, "Planification de trajectoires avion: approche par analogie lumineuse," Ph.D. dissertation, Université Paul Sabatier - Toulouse III, 2011, recherche opérationnelle [cs.RO]. Français. <tel-00817551>.
- [10] S. Chaimatanan, "Strategic planning of aircraft trajectories," Ph.D. dissertation, Université Paul Sabatier - Toulouse III, 2014, optimization and Control [math.OC]. English. <tel-01064452>.
- [11] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *23rd ACM National Conference*, New-York, 1968, pp. 517–524.