Balancing fuel efficiency and environmental impact: 4D trajectory optimization through Fast Marching Tree for transatlantic flights considering contrails

Céline Demouge, Andréas Guitart, Daniel Delahaye

Balancing fuel efficiency and environmental impact: 4D trajectory optimization through Fast Marching Tree for transatlantic flights considering contrails

Céline Demouge, Andréas Guitart, Daniel Delahaye *†

Abstract

This paper proposes a method for computing an airliner trajectory in its cruise phase, minimizing environmental impact. In particular, the problem of contrails is addressed. The proposed method is based on an algorithm derived from robotics, the Fast Marching Tree algorithm. After having been tested on Unmanned Aerial Vehicles (UAVs) and on computing trajectories in a wind field for commercial aircraft, it is now adapted to the case of contrails. Several modifications and additions are made to enable it to deal with soft obstacles that evolve over time, as well as the operational constraints associated with the cruise phase. The method is thus able to propose flight level changes in line with operational expectations, adaptable to the strategy of the user (airline for example). Two experiments are proposed on the Paris-Miami and Paris-Denver flights, showing low computation times and satisfactory results.

Contrails, Sampling-based methods, 4D trajectory, Fast Marching Tree.

1 Introduction

In recent years, sampling-based path planning algorithms have been the subject of intense research. Research into this type of method was initially driven by robotics [1]. Initially, the goal was to find the shortest possible path for a robot while avoiding obstacles. This work in robotics showed the effectiveness of these methods in terms of optimality, but also in terms of computational time. In fact, these methods have been improved to be asymptotically optimal with low computation complexity. More recently, these methods have been tested in other fields such as aviation [2–9]. These different studies have shown that these algorithms are adaptable to the constraints of aeronautical problems. In this context, we propose to apply one of these algorithms (Fast Marching Tree) to a new challenge in aviation: contrail mitigation. This paper is a continuation of two previous works [10, 11]. The first study shows that the Fast Marching Tree (FMT*) algorithm can be adapted to the problem of generating optimal cruise trajectories. This study aims to integrate the wind and the great circle distances in the algorithm. The tests carried out during the study show that the FMT* algorithm effectively solves this problem. The second paper [11] deals with the problem of 2D trajectory optimization in the presence of contrails, solved by the FMT* algorithm. The objective is to find an aircraft trajectory that minimizes its total environmental impact considering both CO_2 and non-CO_2

*The first two authors contributed equally to this work. *Corresponding author:* Céline Demouge, demouge.celine@outlook.com

†C. Demouge, A. Guitart, and D. Delahaye are with ENAC, Université de Toulouse, 7 avenue Edouard Belin, Toulouse, 31400 France (e-mail: demouge.celine@outlook.com, andreas.guitart@enac.fr, delahaye@recherche.enac.fr).

Table 1: GWP for contrails with different time horizon [13].

	$H = 20$ years	$H = 100$ years	$H = 500$ years
$\text{GWP}_{\text{contrail}}(H)$	0.74	0.21	0.064
$\text{GWP}_{\text{CIC}}(H)$	2.2	0.63	0.19

effects. The method has been validated on a European and a transatlantic flight cases. Reducing contrails is essential because they contribute to global warming. In fact, under certain conditions (humid and cold air), they can persist and transform into cirrus clouds, leading to radiative forcing [12]. It is then recommended not to cross such contrail areas but with a limited deviation in order to avoid increasing the associated carbon dioxide (CO_2) emissions. In addition, unlike thunderstorm areas, navigating through contrail areas does not raise safety concerns as they are not considered "hard" obstacles, but rather "soft" obstacles. Finally, unlike thunderstorms, condensation trails can be avoided vertically because their size and shape can vary significantly between two flight levels. It is therefore essential to optimize the vertical profile of the trajectory. This paper therefore presents the extension to 4D (lateral and vertical profiles, and time consideration) of the method described in [11] to take into account the optimization of the vertical profile of the trajectory and the time-dependency of weather data.

This paper is organized as follows: Section 2 presents previous related works on contrails avoidance trajectory generation and sampling-based path planning algorithms. Section 3 proposes a mathematical model and describes the resolution method. Section 4 presents results on different instances of transatlantic flights and Section 5 summarizes the contributions of this paper before giving some research perspectives.

2 Previous related works

This section presents previous works related to contrail avoidance methods and sampling-based path planning algorithms.

2.1 Contrail avoidance methods

Contrails form at the rear of aircraft in cold and humid regions, and can become persistent cirrus clouds (Contrails Induced Cirrus - CIC) with a warming effect on the climate [12]. Avoiding contrails by significant detours would increase fuel consumption and CO_2 emissions, making it environmentally inefficient. Therefore, contrails should be treated as soft obstacles. This section introduces contrail metrics and different methods used in the literature.

2.1.1 Contrail metrics

The goal is to minimize the overall environmental impact of a trajectory, which requires metrics to standardize the impact of both CO_2 and non-CO_2 effects.

The *Global Warming Potential* (GWP) serves as a conversion factor, harmonizing the impact of a greenhouse gas with that of CO_2. The GWP is determined by comparing the impulse effect of a gas to 1 kg of CO_2, integrated over a specified time horizon H. In the case of contrails and CIC, GWP is defined on the basis of emissions per kg of CO_2. More details on GWP can be found in [13], and Table 1 gives examples over three different time horizons.

Additional metrics have been developed that rely on more sophisticated models than those previously discussed [14]. While these metrics are more realistic, they need a more sophisticated model, which is used by the trajectory generation, potentially requiring additional computational time.

In [11], we show that the GWP_{CIC} with a time horizon of 20 years is more restrictive than the 100-year time horizon. In the first case, the generated trajectory, if only optimized in 2D, deviates significantly from the wind-optimal trajectory.

2.1.2 Methods used in literature

Several methods have been used for contrail mitigation trajectory design.

Given the critical role of temperature in contrail formation, flying at lower altitudes proves to be a highly effective strategy for contrail mitigation. In [15], a study reveals that reducing the cruise altitude by 2000 ft results in a 1% increase in radiative forcing due to CO_2 effects, but a significant 18% decrease in non-CO_2 effects.

In [16], optimal control is used to compute wind-optimal trajectories for cross-polar flights, avoiding regions favorable to persistent contrail formation, resulting in a gradual reduction in flight time over contrail areas with a corresponding increase in fuel consumption. In addition, Campbell *et al.* [17] use a Mixed Integer Linear Programming (MILP) approach and achieves a significant 48% reduction in persistent contrails with only 0.5% additional fuel burn. Metaheuristics, such as genetic algorithms, are explored in [18]. Deterministic graph methods, as exemplified in [19], where the A* algorithm is used in conjunction with a simulation framework. While these deterministic approaches produce satisfactory results, their reliance on an established network presents challenges in the context of free flight. A dense graph provides a near-optimal solution but at a higher computational cost, while a sparse graph sacrifices optimality for faster computation.

This study is based on previous papers [10, 11] based on the Fast Marching Tree algorithm which is a sampling-based path planning algorithm. The next subsection presents the principle of these algorithms.

2.2 Sampling-based path planning algorithms

The sampling-based path planning methods consist in generating a graph in order to find the optimal path between two points [20–23].

Before discussing the algorithms, it is important to formulate the problem and define the primitive functions that they use.

Let $\chi = (0,1)^d$ be the configuration space, where $d \in \mathbb{N}$ is the space dimension , $(d \geq 2)$. Let χ_{obs} be the obstacle region, such that $\chi \backslash \chi_{obs}$ is an open set, and denotes the obstacle-free space as $\chi_{free} = cl(\chi \backslash \chi_{obs})$, where $cl(\chi)$ denotes the closure of a set χ. The initial condition is denoted by $x_{start} \in \chi_{free}$, and the goal region χ_{goal} is an open of χ_{free}.

Sampling-based path planning algorithms mainly use four functions.

Sampling: the sampling function produces a sequence of points in χ. The distribution of these points can be either uniform or randomly generated. It is important to note that the use of random sampling renders the algorithm's solutions non-reproducible. Ideally, the algorithm should be able to sample points directly from the free space χ_{free} for better performance.

Nearest Neighbor: this function returns the vertex that is the nearest to a point $x \in \chi$ based on a given distance metric.

Near Vertices: this function returns the vertices that are contained in a ball of radius r centered at a point $x \in \chi$.

Collision Test: this function returns True if the straight line connecting two points $x, x' \in \chi$ lies in χ_{free} and False otherwise.

The three main methods are Probabilistic RoadMaps (PRM) [24], Rapidly-exploring Random Tree (RRT) [25] and Fast Marching Tree (FMT*).

This paper only presents the FMT* algorithm which was used in the two previous papers. This algorithm has been introduced by [26] in 2015. The algorithm can be summarized by the three following steps:

1. generate n samples in the search space;
2. perform a forward propagation over the samples to generate the tree until it reaches the destination;
3. perform a back-propagation on the tree to obtain the best path from the origin to the destination.

The forward propagation is the main part of the algorithm, but before going into the details of this step, it is necessary to introduce some sets. The set $V_{unvisited}$ contains nodes that have not been visited yet. V_{closed} contains nodes that have been visited and have an unchangeable optimal cost. The set V_{open} contains nodes that have been visited but have a temporary cost assigned. The next four figures (see Fig. 1), adapted from [26], show the local optimization phase, which is iterated until the goal x_{goal} is reached. The algorithm starts with a graph consisting of only one node, the starting point x_{start}.

All nodes within a distance r of a given node are considered to be its neighbors. The parameter r represents the neighborhood radius and is determined by a function of the number N of samples. It is defined as follows:

$$r = (1+\eta)\gamma_{FMT^*}, \tag{1}$$

where $\eta > 0$ is a parameter which has to be defined and γ_{FMT^*} is the minimal radius computed as follows:

$$\gamma_{FMT^*} = 2\left(\frac{1}{d}\right)^{1/d}\left(\frac{\mu(\chi_{free})}{\zeta_d}\right)^{1/d}\left(\frac{\log N}{N}\right)^{1/d}, \tag{2}$$

where d is the space dimension, $\mu(\chi_{free})$ its Lebesgue measure[1] and ζ_d the volume of the d-dimensional unitary ball[2].

3 Problem modeling and resolution method

This section presents the problem model, contrail areas computation, and the resolution approach.

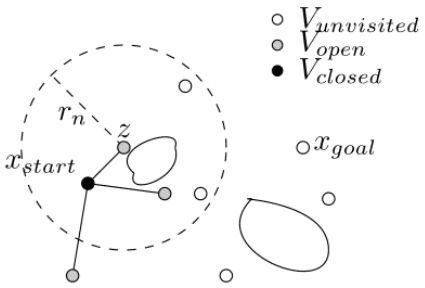
3.1 Problem model

The aim is to compute an aircraft 4D cruise trajectory γ that minimizes its total environmental impact composed of CO_2 and non-CO_2 effects. In this study, the non-CO_2 effect is only represented by the contrails. The problem is written as follows:

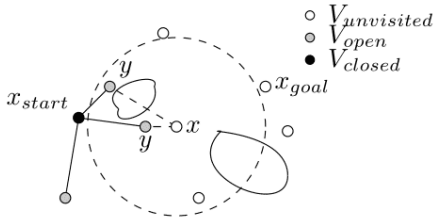
$$\underset{\gamma \in \Gamma}{\text{minimize}} \quad \int_{t_0}^{t_f} C_{CO_2} f(x,y,z,t) + C_r r(x,y,z,t) dt, \tag{3}$$

[1] $\mu([a_1,b_1]\times[a_2,b_2]) = (b_1-a_1)(b_2-a_2)$ where $b_1 > a_1$ and $b_2 > a_2$.

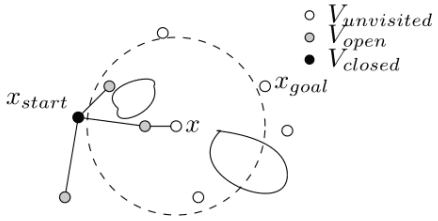
[2] In 2D space, ζ_2 is the surface of a disk of radius 1 ($\zeta_2 = \pi$).



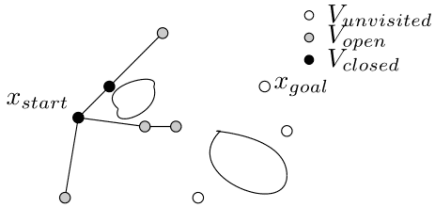
(a) The lowest-cost node z from set V_{open} is selected and the nodes within $V_{unvisited}$ which are near to z are found.



(b) In the second step, the neighbors of the unvisited node x within V_{open} are found, and x is connected to have an optimal cost without going through an obstacle.



(c) A connection is created between x and the neighbor of the locally-optimal step connection.



(d) All neighbors of z are visited and are added to V_{open}. z is added to V_{closed}. FMT* moves to the next iteration with the node which has the lowest cost.

Figure 1: Steps of an iteration of the FMT* Algorithm, adapted from [26].

where f is the fuel flow of the aircraft, C_{CO_2} is the conversion factor from the fuel to the CO_2 emitted (for weighting standard fuel, $C_{CO_2} = 3.16$ kg (CO_2) / kg (fuel) [27]), C_r is a factor used to balance the impact of CO_2 and that of contrails, typically here the GWP is used, and $r(x,y,z,t)$ is a penalty function to determine whether the aircraft is in a persistent-contrail area or not. In our application, this penalty function is set to 1 inside contrail areas, and 0 elsewhere.

The computed trajectory γ must follow the flight dynamics and operational requirements. In particular, aircraft must fly at specific Flight Levels (FL) which are numbered every 1000 ft from mean sea level. For instance, if an aircraft is flying at an altitude of 33,000 ft above mean sea level, its flight level is 330.

3.2 Computing 2D path

In [10], we propose a method based on the FMT* algorithm to find wind-optimal trajectory. This method is modified in [11] to minimize the environmental impact by considering contrails. In this last work, the following assumptions are made:

- the wind and contrails are considered static and only space-dependent;
- the aircraft remains at the same altitude throughout the flight;
- the true air speed is constant

As usually done in aviation, 2D navigation points, called waypoints, are used to define a trajectory. Since FMT* is a method based on the building of a tree from sampled points, the computed trajectory consists of a sequence of these points. Then, the cost of such a discretized trajectory is:

$$\sum_{i=1}^{n_\gamma - 1} c_{i,i+1}, \tag{4}$$

where n_γ is the number of waypoints of the trajectory, and $c_{i,i+1}$ is the cost of flying from point i to point $i+1$. Then, problem (3) becomes:

$$\underset{\gamma \in \Gamma}{\text{minimize}} \quad \sum_{i=1}^{n_\gamma} c_{i,i+1}. \tag{5}$$

To take contrails into account, the cost function is modified. The cost of a link between two nodes i and j is redefined as follows:

$$c_{i,j} = t_{i,j}\ (1 + \lambda_{i,j}\ g_H), \tag{6}$$

where $t_{i,j}$ is the flight time between i and j, $\lambda_{i,j} \in [0,1]$ is the proportion of the edge that is in a persistent contrail area, and g_H is the chosen GWP value, depending on the time horizon H. The method is tested on continental and transatlantic case studies and shows its efficiency. However, some assumptions should be modified to be closer to reality. Indeed, for a transatlantic flight, it is necessary to develop a dynamic version by continuously updating the trajectory, as weather information is received. Moreover, although the altitude is almost constant during the flight, the cruise phase can be optimized and, if it is advantageous, altitude changes could be considered [28]. It is important to note that most changes in flight level are climbs. This is because the aircraft burns less fuel at higher altitudes (see Fig. 2), but pilots must wait until they have burned enough fuel to fly at these levels. Most flights start at FLs below 330 and end at FLs up to 410. Consideration of contrails calls these practices into question. In fact, it may be preferable to fly at lower altitudes due to the low

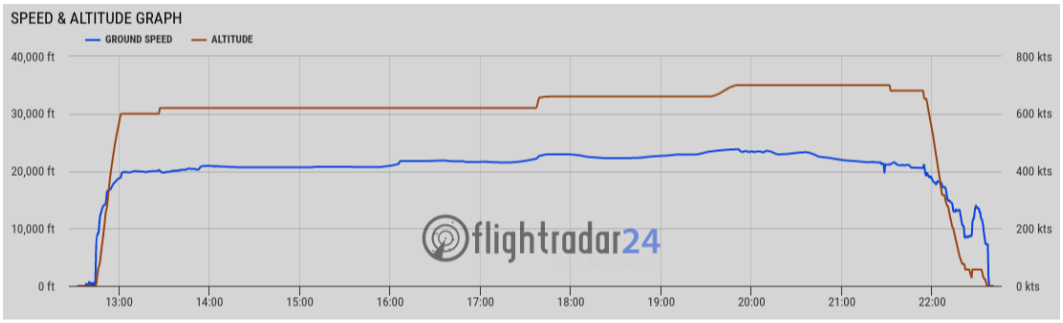


Figure 2: Ground speed and vertical profile of the Paris-Miami flight on 26 January 2024, from [29].

or non-existent presence of contrails. For all these reasons, a 4-dimensional extension is necessary. In our extension, the cost of an arc is:

$$c_{i,j} = f_{i,j}\ (1 + \lambda_{i,j}\ g_H), \tag{7}$$

where $f_{i,j}$ is the fuel consumed between i and j. The fuel consumed over an arc is computed as follows:

$$f_{i,j} = FF_{i,j} t_{i,j}, \tag{8}$$

where $FF_{i,j}$ is the fuel flow over the arc. Both flight time (more precisely the true airspeed) and the fuel flow depends on the flight altitude and aircraft mass. For each 2D optimization, the mass considered is the one at the starting point. The fuel flow is then determined for the concerned altitude from a database. The true airspeed is computed using Equation (10) presented in the sequel.

3.3 4D extension

For the 4D extension of the method proposed in [11], two main challenges arise: updating the weather data and finding the most appropriate flight level for each part of the cruise phase, according to the environmental cost and the flyability of the aircraft. The proposed extension recomputes the trajectory of the aircraft with a constant step. To do this, it is necessary to update the weather and contrail data.

As explained in the previous subsection, the aircraft can reach a higher flight level if it burns enough fuel. It is therefore necessary to check the flight level regularly. The ability to climb depends on many parameters such as take-off mass, speed and aircraft model. These parameters vary from airline to airline and are often confidential. For this reason, the time between two climbs is tunable and has been set at two hours in this study as it reflects the real flight profiles. To take altitude into account in the trajectory generation process, a first idea consists in fully extending the FMT* algorithm in the third dimension. However, this brut force approach may increase strongly the computation time and may sample the search space in regions where the aircraft will not be able to go due to the FL constraints. Instead we propose to restrict the search space to an altitude range compatible with the aircraft performances. We propose to recompute the trajectory at each time step at the current altitude, as well as at the higher (if possible) and lower flight levels. One must remind that changes in altitude have also an impact on the fuel consumption. In this study, the cost of a descent is assumed to be zero. The cost of a level change is therefore defined as follows:

$$\kappa(\Delta z) = 60\frac{\max(\Delta z, 0)}{\text{ROC}}\text{FF}^{\text{c}}, \tag{9}$$

where Δz is the difference in altitude (2000ft in this study), ROC is the rate of climb in ft/min and FF^{c} is the fuel flow during climb in kg/s as a function of ROC, true air speed (TAS) and altitude. The ROC evolves during the flight and can be considered as a tunable parameter. In this study, we consider a ROC from 1000ft/min to 500ft/min, decreasing by 250 every 2 hours. The TAS depends on the flight level and is defined as follows:

$$TAS_l = 0.98^{(l-l_{min})/10} TAS_{min}, \tag{10}$$

where l_{min} is the minimum flight level and TAS_{min} is the true airspeed at that flight level. The fuel flow is a function of the flight level and is updated every flight hour. In fact, it decreases during the flight because the aircraft is lighter.

To summarize the trajectory update procedure, at a given iteration i, the best flight level l_i^* is defined as follows:

$$l_i^* = \underset{l \in L_i}{\arg\min}(c(\gamma_{l,i}) + \kappa(100(l - l_{i-1}))), \tag{11}$$

where L_i is the set of reachable flight levels, $\gamma_{l,i}$ is the best trajectory generated by the FMT* algorithm at the flight level l and c is the environmental cost function. This procedure is repeated at each time step until the aircraft reaches the destination. In this study, the trajectory update time step is the same as the weather data update time step (1 hour). This value is sufficient, as flight levels change at least every hour. However, in our algorithm, this value can be changed by the user. Fig. 3 shows an example of an iteration of the algorithm. In this example, the aircraft has flown from the origin O to the point P at the flight level 330. In this iteration, the aircraft can climb at FL350, remain at FL330 or descend at FL310. The algorithm computes a trajectory from point P to the destination D for each flight level with the new weather and contrail data. It then selects the flight level and the trajectory that minimizes the cost function. In this example, it would be better to change the flight level due to the presence of a large area of contrails at FL330.

The 4D trajectory generation method can be summarized by the following steps:

1. At the beginning of the flight, select the first flight level among those available and compute the associated trajectory that minimizes the cost function, taking into account the initial weather and contrail data.

2. After a time step *i.e.* each time the flight level is called into question, generate a new trajectory from the new point of the aircraft to the destination, taking into account the new data and the ability to climb at the next flight level. Aircraft data are also updated, in particular the mass and the fuel flow at each flight level considered are updated.

3. Repeat the previous step until the aircraft reaches the destination.

4 Results

This section presents the results of two case studies: Paris-Miami and Paris-Denver. Subsection 4.1 introduces the data used, then the results for the flight Paris-Miami are presented in subsection 4.2 and the ones for the flight Paris-Denver in subsection 4.3. The computer used for the experiments is equipped with an Intel Core-i9 processor, with 64 Go RAM.

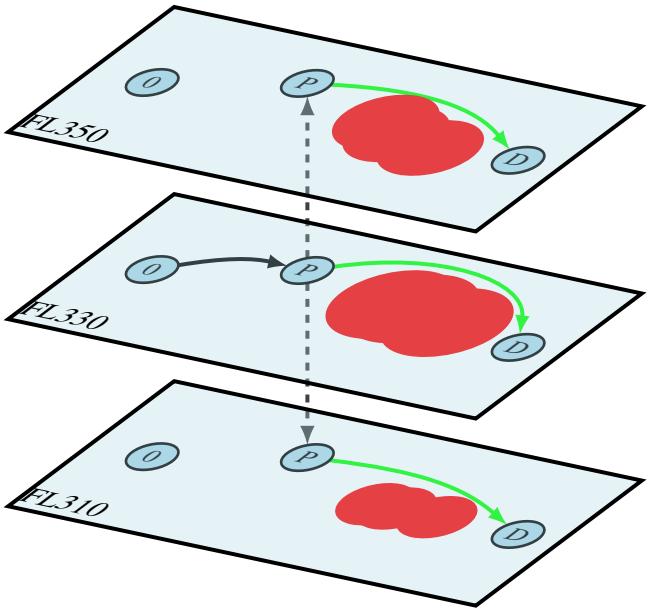


Figure 3: Update iteration: the aircraft has flown from the origin O to the point P at FL330 and can decide to climb at FL350, remain at FL330 or descend at FL310 for the remainder of the flight.

4.1 Data

The weather data used for the proposed experiments are extracted from the ERA5 database [30] and processed for contrail area computation using the CLIMaCCF Python library [31]. The data used are from 1st August 2023, and contain also associated wind. As done in [11], the metric used for contrails quantification is the GWP, with a 20 or 100-year time horizon [13] (see Table 1 for more details).

Data for the aircraft, A388 type, used are extracted from [32], with the true airspeed fixed to $V_a = 440$ kts at Flight Level 310 and an initial mass of 80% of the Maximum Take Off Mass. The fuelflow is extracted from the OpenAP Python library [33].

As explained in subsection 3.3, the weather data are updated every hour, the first flight level authorized is the FL310 and every 2 hours, a new higher level is authorized (with a 2000 ft-step). The rate of climb varies from 1000 ft/min to 500 ft/min, decreasing by 250 every 2 hours. These parameters are fixed for the proposed numerical experiments, however, the final users can adapt them to their situation and, for example, an airline can adapt them to the data of its flights, as they are only inputs of the algorithm.

4.2 Paris - Miami case study

The first study case done is a flight from Paris to Miami. It was tested with different departure times on the same day, with the aircraft and altitude parameters previously mentioned.

Let us first consider the results obtained for a given departure time, here 04:00 UTC. Figs. 4 and 5 show the lateral and vertical profiles of the trajectory without taking contrails into account (only fuel consumption is minimized). Figs. 6 and 7 show the lateral and vertical profiles of the trajectory where contrails with GWP 100 are considered.

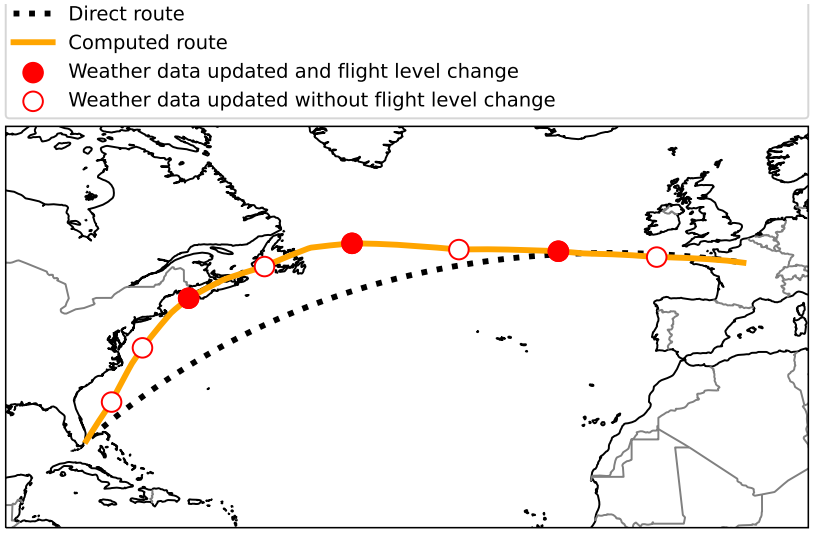


Figure 4: Lateral profile of the computed wind-optimal trajectory (no contrails considered) for a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC. The dotted line is the direct route (great circle) and the plain line is the computed trajectory. The circles are the positions at which weather (and then contrails) data are updated and the filled circles are particular points at which the flight level changes.

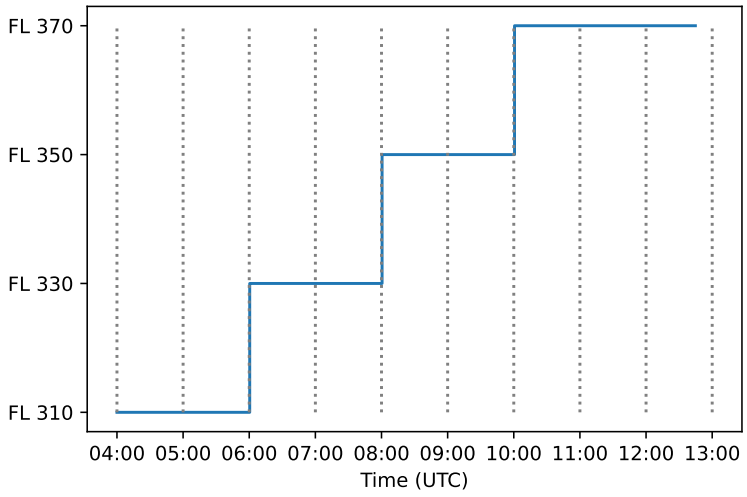


Figure 5: Vertical profile of the computer wind-optimal trajectory (no contrails considered) for a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC.

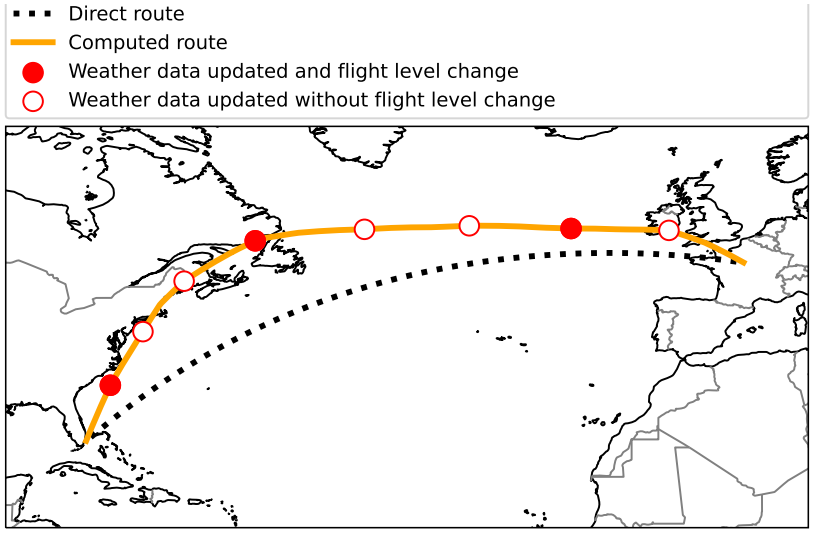


Figure 6: Lateral profile of the computed trajectory with GWP100 for a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC. The dotted line is the direct route (great circle) and the plain line is the computed trajectory. The circles are the positions at which weather (and then contrails) data are updated and the filled circles are particular points at which the flight level changes.

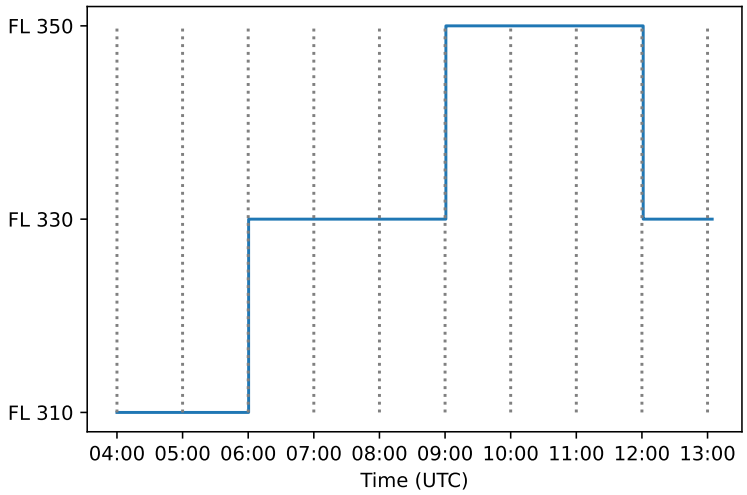


Figure 7: Vertical profile of the computed trajectory with GWP100 for a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC.

Fig. 7 shows a descent at the end of the flight (12:00 UTC). This is an expected behavior, since FL 330 produces fewer contrails at the end of the flight than FL 350, as shown in Fig. 8. The descent

becomes a more attractive option, even if it means higher fuel consumption.

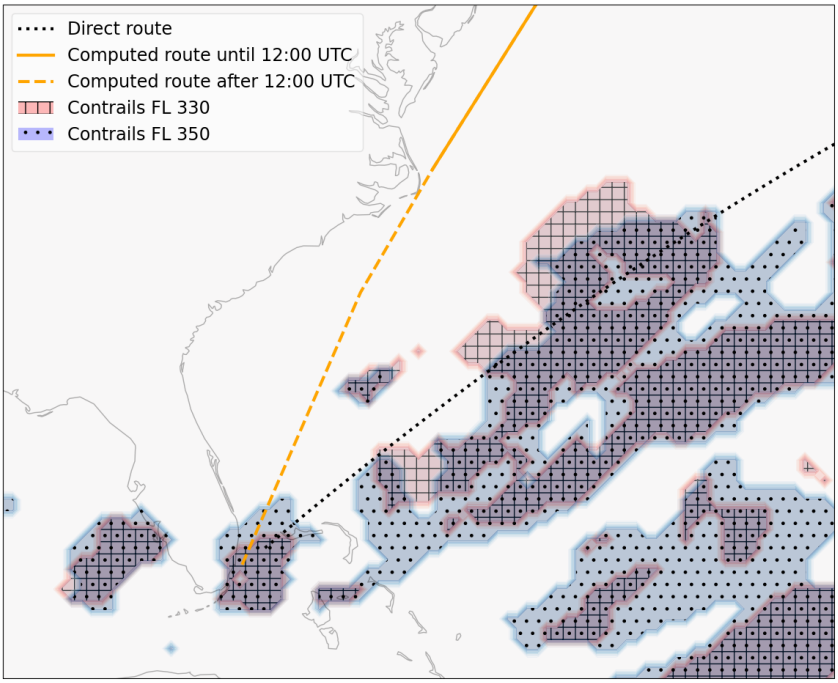


Figure 8: Focus on the end of the computed trajectory for the Paris-Miami flight with departure time 04:00 UTC. The computed trajectory is represented by the orange plain line. The line becomes dashed at the moment at which the decision to level off is made. Areas hatched with grids (respectively with points) are contrail areas at FL330 (respectively at FL350).

The vertical profile without consideration of contrails (Fig. 5) is a standard flight profile and corresponds to what can be operated today. The profile obtained with the contrails is adapted (Fig. 7), allowing to find a balance between the CO_2 and contrails, depending on the chosen metric. The proposed method avoids to change FL too often, which, in addition to higher fuel consumption, would result in discomfort for passengers, additional workload for air traffic control, and, in the long term, premature engine wear.

Several computations have been performed with different start times, between 0:00 and 14:00 UTC. Different results are obtained, depending on the start time and the associated weather conditions, and the time horizon for the GWP calculations (20 or 100 years). Fig. 9 shows the distribution of the total objective function for each time horizon. Figs. 10 and 11 show the distribution of the fuel consumption and flight time respectively for each time horizon. Fig. 12 shows the reduction of flight time in contrail areas. Fig. 13 shows the number of level-offs and level-ups for each departure time for both time horizons.

Figs. 10 to 12 show how the time horizon for GWP computation has an impact on the results obtained. As expected, the impact of contrails is relatively higher with smaller time horizons, resulting in more fuel burnt to avoid flying through contrail areas. When taking contrails into account, fuel consumption increases. However, the additional cost is on average less than the one observed with 2D optimization alone. In the worst cases of the shorter time horizon, the cost remains relatively

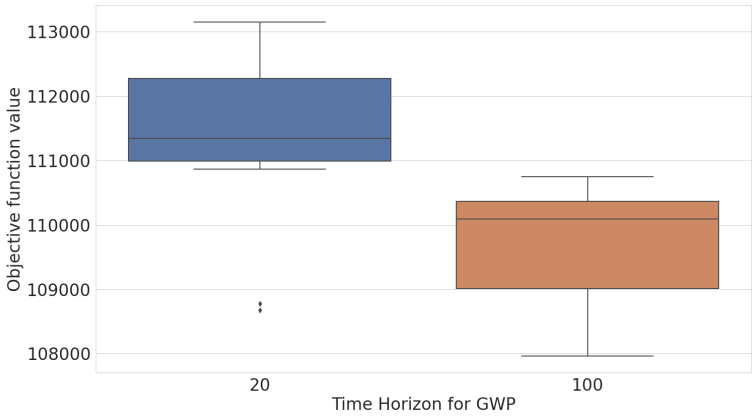


Figure 9: Distribution of the value of the objective function for the Paris-Miami tests, depending on the time horizon chosen for the GWP metric.

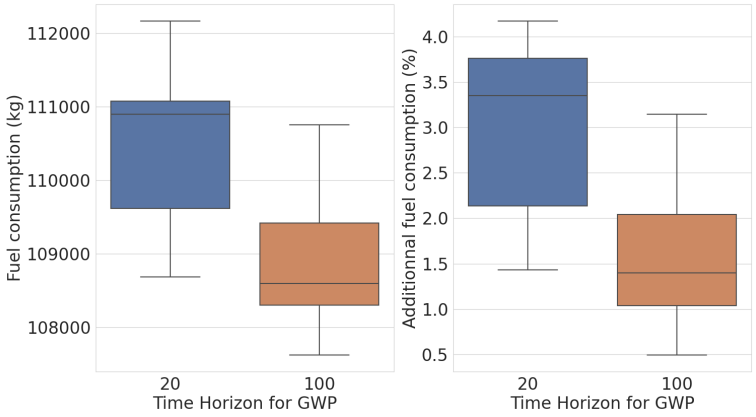


Figure 10: Distribution of the value of fuel consumption for the Paris-Miami tests, depending on the time horizon chosen for the GWP metric. The left side shows the value of fuel consumption, while the right side shows the additional fuel consumption compared to the case where no contrails are considered (only fuel consumption is minimized).

high. To solve this issue, the problem can be more tightly constrained to avoid these cases, even at the expense of the overall environmental impact (depending on the metric chosen).

Fig. 13 illustrates that the number of flight level changes is relatively low. Moreover, in current traffic situations, leveling off is quite uncommon. Our results show a small number of levels off, which is a new behavior to be adopted for contrail avoidance. The proposed method allows for trajectories that propose delayed climbs or longest stays at lower levels, thereby reducing the number of level changes. It is important to note that no additional constraints were added to obtain this.

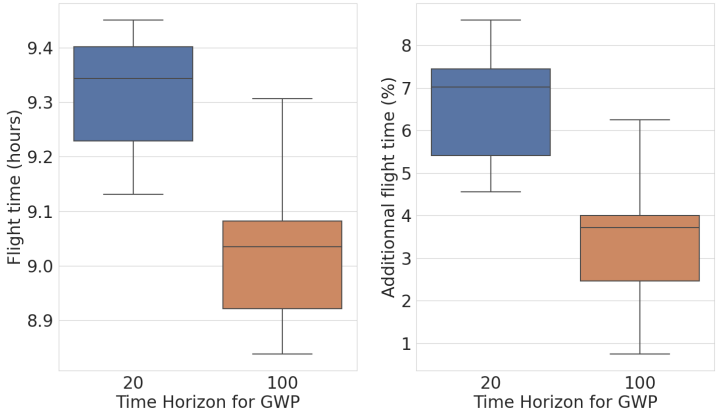


Figure 11: Distribution of the value of flight time for the Paris-Miami tests, depending on the time horizon chosen for the GWP metric. The left-hand side shows the value of the flight time, while the right-hand side shows the additional flight time compared to the case where no contrails are considered (only fuel consumption is minimized).

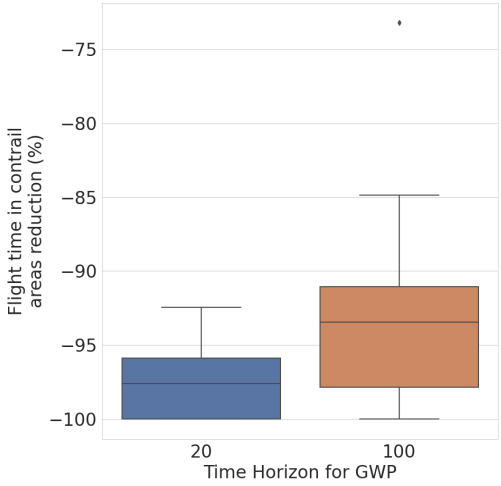


Figure 12: Distribution of the reduction of flight time in contrail areas for the Paris-Miami tests, depending on the time horizon chosen for the GWP metric, compared to the case where no contrails are considered (only fuel consumption is minimized).

Pareto analysis

To better understand and analyze the results, we propose a weighting approach for the two objectives, fuel and flight time in contrail areas. The cost of an edge then becomes:

$$c_{i,j} = f_{i,j}\,(1+\alpha\,\lambda_{i,j}), \tag{12}$$

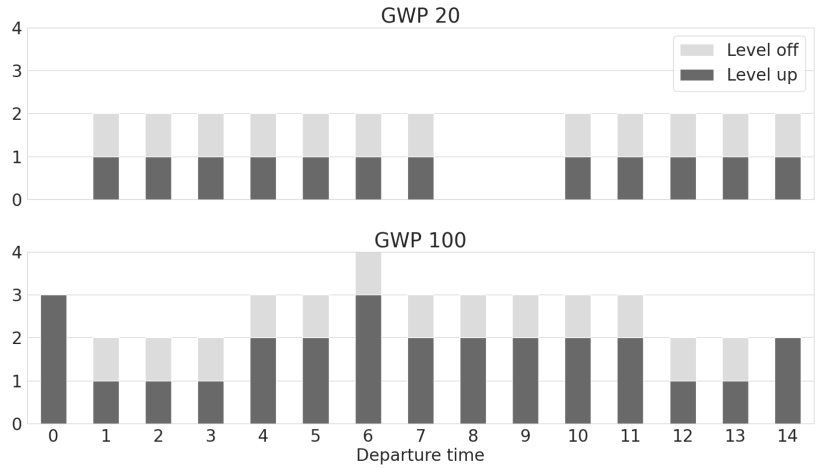


Figure 13: Numbers of levels off (light grey) and levels up (dark grey) for each departure time considered in tests for Paris-Miami flights with GWP 20 (top figure) and GWP 100 (bottom figure).

where $\alpha \geq 0$ is a user-defined parameter. Numerical experiments have been conducted on the particular case of a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC, as proposed for the first results presented in this Section. Fig. 14 displays the reduction of flight time in contrail areas in function of the additional fuel consumption (compared to the case where only fuel is minimized). Figs. 15 and 16 show the flight time in contrail areas reduction and the additional fuel consumption α value dependency respectively. Finally, Fig. 17 shows the number of flight level changes in function of α.

Several observations can be made from these results. First, Fig. 14 shows that the first few percent of contrails are easy to eliminate, *i.e.*, with a relatively small additional CO_2 budget, a significant proportion of contrails can be avoided. Moreover, from the point of view of the trade-off between the CO_2 emitted and the contrails created, it may not be worthwhile to completely eliminate contrails at the cost of much higher consumption with limited marginal gain in contrails. Steps appear in Fig. 14, as well as in Figs. 15 and 16. There are two main reasons for this. First, the algorithm used is based on sampling, and even if it is asymptotically optimal, the tree is built from a sampling of a finite number of points. As a result, approximations necessarily occur. In addition, there may be breaks in the choice of whether or not to avoid a zone. Even if the aim is to minimize the time flown through a region of contrails, the algorithm may also behave in a binary way when avoiding certain zones (especially the smallest ones), so there may be breaks depending on the value of α. There is a break in the number of flight level changes as a function of alpha, as shown in Fig. 17. This is due to the fact that, above a certain value of α, the relative cost of contrails becomes so high that it becomes more attractive not to climb at a certain point in the flight. A last remark is that even if all contrails are eliminated in the solution, fuel consumption may still increase (see Fig. 16). This is due to the heuristic we have chosen for managing time dependencies. A trajectory can be computed to avoid contrails that will not be encountered following a data update. This is the case when the relative cost of contrails becomes too high. As the computation time is low, several tests with different weightings of the contrails can be carried out to limit this problem.

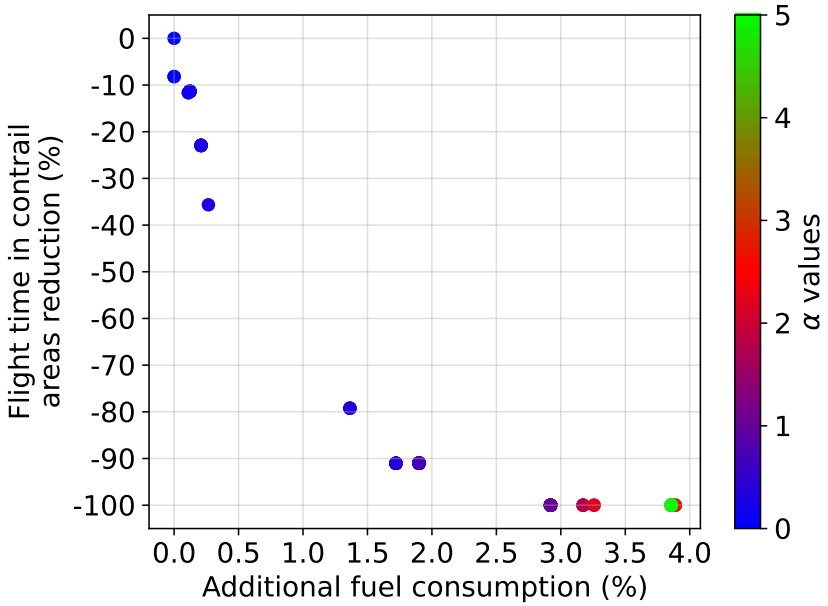


Figure 14: Flight time in contrail areas reduction in function of additional fuel consumption when using user-defined penalty for contrails, for a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC.

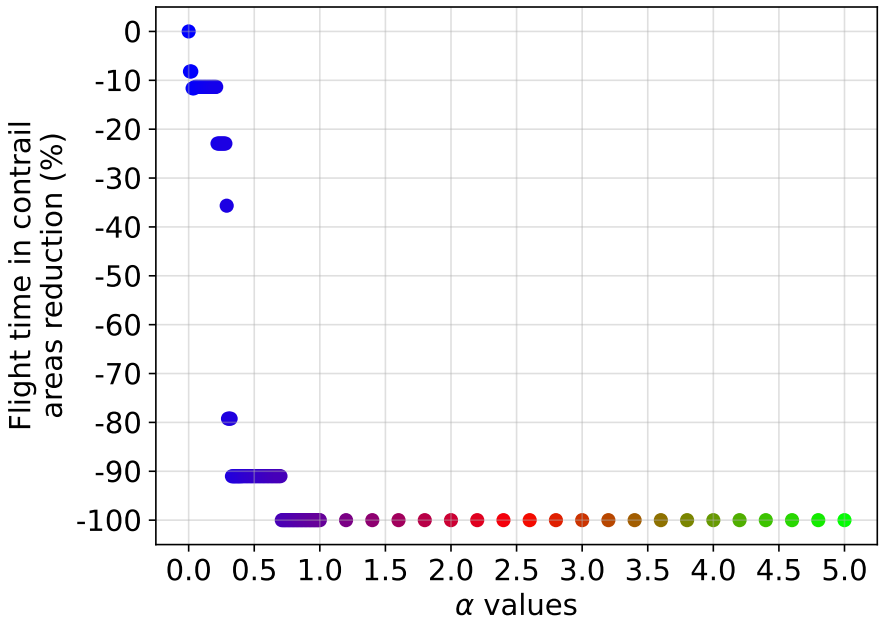


Figure 15: Flight time in contrail areas reduction in function of the value of the user-defined penalty for contrails (α), for a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC.

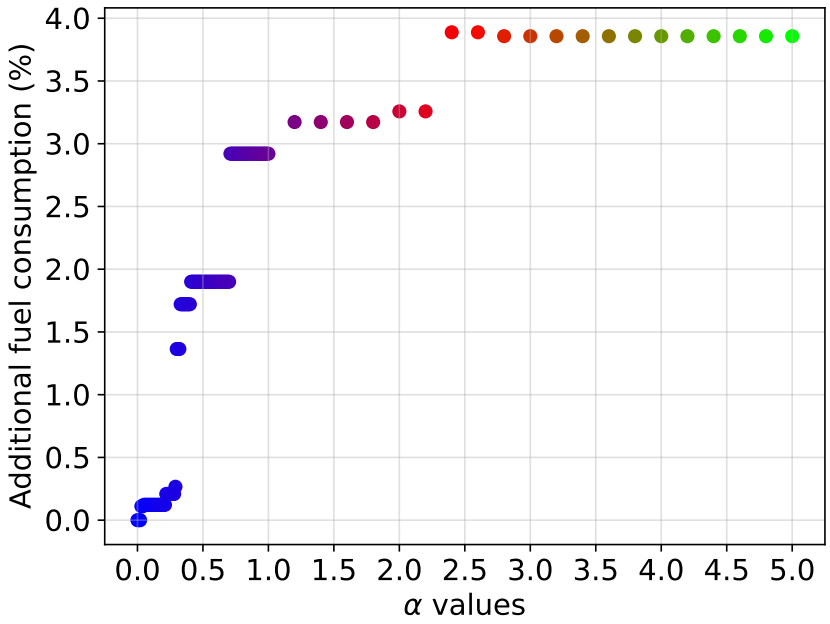


Figure 16: Additional fuel consumption in function of the value of the user-defined penalty for contrails (α), for a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC.

4.3 Paris - Denver case study

The second study case is a flight from Paris to Denver. It has been tested with different departure times, performing computations with start times between 0:00 and 13:00 UTC, on the same day with the previously mentioned parameters for the aircraft and altitude consideration. Different results are obtained, depending on the start time, the associated weather conditions, and the time horizon for the GWP calculations (20 or 100 years). Fig. 18 shows the distribution of the total objective function for each time horizon. Figs. 19 and 20 show the distribution of the fuel consumption and flight time respectively for each time horizon. Fig. 22 shows the number of level-offs and level-ups for each departure time for both time horizons.

Based on Figs. 19 and 20 the same conclusion as in the Paris-Miami tests can be drawn about the impact of time horizon for GWP. However, the additional fuel consumption is higher than in the Paris-Miami tests. This is due to a worse contrail situation compared to the Paris-Miami flight. Once again, constraints can be added to avoid too high additional fuel consumption, but at the cost of higher overall climate impact. Another action lever could be to accept to climb or descend by 1000 ft instead of the current 2000 ft step or to be less constrained on the vertical profile according to flight mechanics constraints. Indeed, Fig. 22 shows that in many cases, the flight level does not change and this may indicate that the vertical profile is too constrained for altitude to provide sufficient leverage to limit contrails. However, current operational practices consider flight-level parity constraints based on the trajectory direction to reduce traffic complexity.

The proposed results were obtained with an average computation time of 5.2 minutes for Paris-Miami and 5.9 minutes for Paris-Denver. These computation times are relatively small, considering that the method takes into account both temporal and altitude updates. Moreover, these computa-

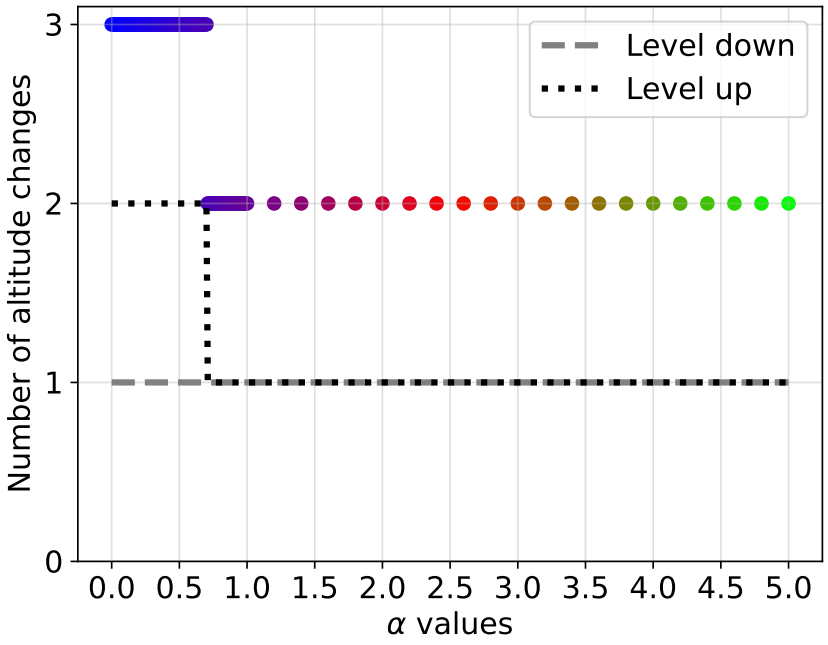


Figure 17: Number of flight level changes (up and down) in function of the value of the user-defined penalty for contrails (α), for a Paris-Miami flight on 1st August 2023, with departure time 04:00 UTC.

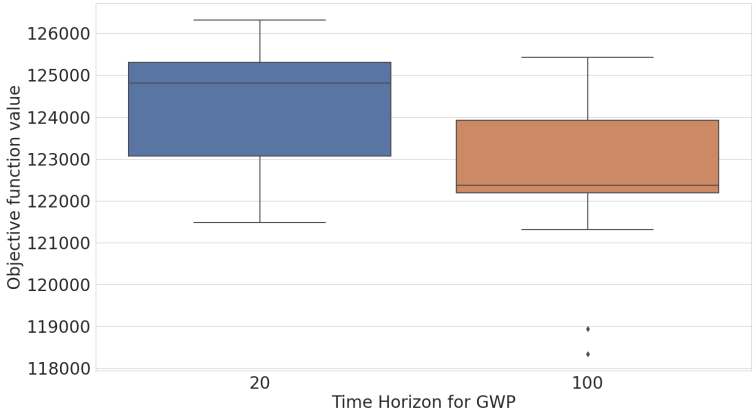


Figure 18: Distribution of the value of the objective function in the Paris-Denver tests, depending on the time horizon chosen for the GWP metric.

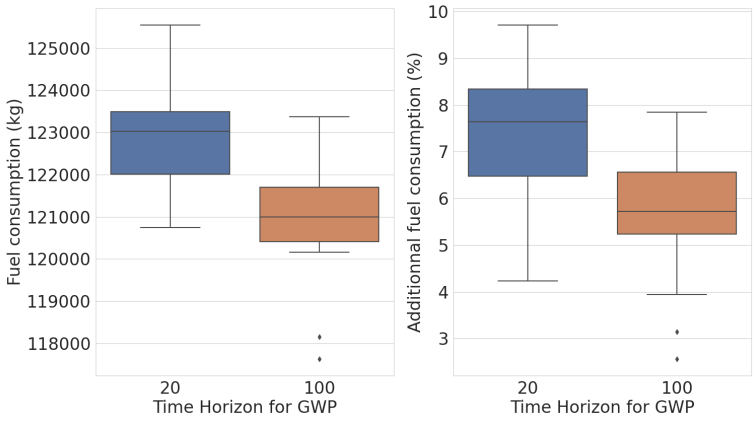


Figure 19: Distribution of the value of fuel consumption for the Paris-Denver tests, depending on the time horizon chosen for the GWP metric. The left side shows the value of fuel consumption, while the right side shows the additional fuel consumption compared to the case where no contrails are considered (only fuel consumption is minimized).

tional performances can be improved by parallelization if necessary. The proposed method has been tested on arbitrary aircraft performance data, but they are completely adaptable, according to the user's needs. These data, e.g. aircraft mass or rate of climb, are sensitive and therefore generally not public. Moreover, they change during the flight, and the proposed method can also allow the computations to be updated during the flight as the flight data is updated.

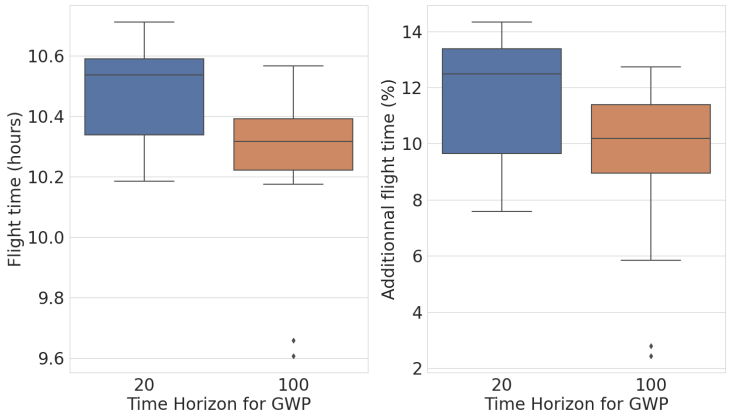


Figure 20: Distribution of the value of flight time for the Paris-Denver tests, depending on the time horizon chosen for the GWP metric. The left side shows the value of the flight time, while the right side shows the additional flight time compared to the case where no contrails are considered (only fuel consumption is minimized).

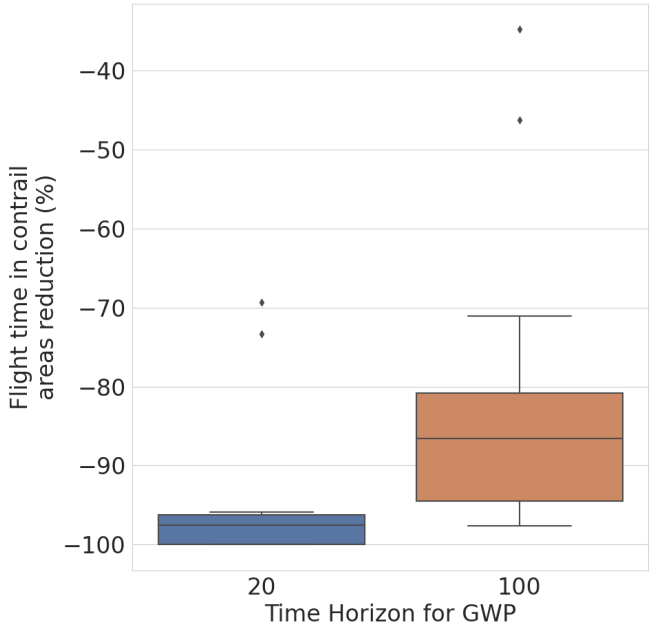


Figure 21: Distribution of the reduction of flight time in contrail areas for the Paris-Denver tests, depending on the time horizon chosen for the GWP metric, compared to the case where no contrails are considered (only fuel consumption is minimized).

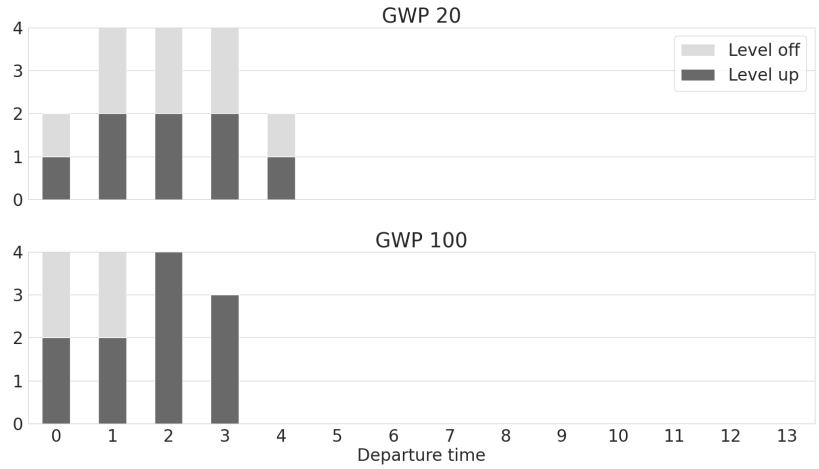


Figure 22: Numbers of levels off (light grey) and levels up (dark grey) for every time departures considered in tests on Paris-Denver flights with GWP 20 (top figure) and GWP 100 (bottom figure).

5 Conclusion

This paper proposes a robotics-based method for generating an airliner cruise trajectory, to minimize the environmental impact of an aircraft trajectory. The Fast Marching Tree (FMT*) was first adapted to the great circle distance, as it was originally made for Euclidean spaces and for considering wind fields [10]. The algorithm was originally designed to avoid hard obstacles, so it was adapted to contrails which are soft obstacles [11], only in 2D. Finally, this paper proposes a method derived from the FMT* algorithm to consider time-varying weather data by regular updates. Moreover, since altitude change can be an efficient action lever for contrail avoidance, this method proposes to add it as a decision variable. These two extensions have been implemented by taking into account operational constraints. For the temporal updating, the method is adapted by using fixed step updates synchronized with the weather updates, thus allowing to reduction of the uncertainties on this data. For taking into account altitude, we restrict the search space to an altitude range compatible with the performance of the aircraft. In addition, it is suitable to update the trajectory during the flight in case of changes in the aircraft performance, which again allows the trajectory to be readjusted as the uncertainties are reduced. Finally, the aircraft data used are standard, but they can be adapted to suit the user's needs, for example, an airline can adapt the data to its strategy and departure conditions.

There are several directions to extend this work. First, there are more complex time and space-dependent metrics associated with contrails, such as aCCFs [31]. The method could be adapted to these metrics. Second, the data used are meteorological and therefore inherently uncertain. This type of uncertainty could be incorporated, for example, to obtain robust trajectories. Finally, the computation time could be improved by parallelization. Indeed, at each iteration, the computations performed on each flight level are independent and can therefore be performed on different computing nodes.

Acknowledgment

The authors thank DGAC (French civil aviation authority) for prompting and funding this work, and more specifically its DTA and DSNA services.

References

[1] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[2] P. Pharpatara, B. Hérissé, and Y. Bestaoui, "3-D Trajectory planning of aerial vehicles using RRT*," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1116–1123, May 2017.

[3] A. Fallast and B. Messnarz, "Automated trajectory generation and airport selection for an emergency landing procedure of a cs23 aircraft," *CEAS Aeronautical Journal*, vol. 8, pp. 481–492, 2017.

[4] R. Sáez, H. Khaledian, X. Prats, A. Guitart, D. Delahaye, and E. Feron, "A fast and flexible emergency trajectory generator enhancing emergency geometric planning with aircraft dynamics," in *Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)*, 2021.

[5] A. Guitart, D. Delahaye, and E. Feron, "An accelerated dual fast marching tree applied to emergency geometric trajectory generation," *Aerospace*, vol. 9, no. 4, p. 180, 2022.

[6] L. Bonin, D. Delahaye, A. Guitart, E. Feron, and X. Prats, "Optimal path planning for soaring flight," in *Conference on Guidance Navigation and control (CEAS EuroGNC 2022)*, 2022.

[7] A. Guitart, D. Delahaye, F. M. Camino, and E. Feron, "Collaborative generation of local conflict free trajectories with weather hazards avoidance," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[8] J.-C. Lebegue, A. Guitart, C. Demouge, D. Delahaye, J. Hoekstra, and E. Feron, "Aircraft cruise alternative trajectories generation: a mixed rrg-clustering approach," in *International Conference on Intelligent Transport Systems*. Springer, 2023, pp. 34–50.

[9] R. Sáez García, D. Toratani, R. Mori, and X. Prats Menéndez, "Generation of rnp approach flight procedures with an rrt* path-planning algorithm," in *DASC 42nd Digital Avionics Systems Conference: Barcelona, Spain, October 1-5, 2023: conference proceedings*, 2023.

[10] C. Demouge, A. Guitart, and D. Delahaye, "Fast marching tree applied to geodesic trajectories in presence of uncertain wind: a day of flights in europe study," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*. IEEE, 2023, pp. 1–10.

[11] C. Demouge, A. Guitart, G. Dannet, and D. Delahaye, "An adapted fast marching tree for contrails mitigation: A short-and long-range flight study," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 4901–4906.

[12] D. Lee, D. Fahey, A. Skowron, M. Allen, U. Burkhardt, Q. Chen, S. Doherty, S. Freeman, P. Forster, J. Fuglestvedt, A. Gettelman, R. De León, L. Lim, M. Lund, R. Millar, B. Owen, J. Penner, G. Pitari, M. Prather, R. Sausen, and L. Wilcox, "The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018," *Atmospheric Environment*, vol. 244, p. 117834, 2021.

[13] J. Fuglestvedt, K. Shine, T. Berntsen, J. Cook, D. Lee, A. Stenke, R. Skeie, G. Velders, and I. Waitz, "Transport impacts on atmosphere and climate: Metrics," *Atmospheric Environment*, vol. 44, no. 37, pp. 4648–4677, 2010.

[14] S. Matthes, V. Grewe, K. Dahlmann, C. Frömming, E. Irvine, L. Lim, F. Linke, B. Lührs, B. Owen, K. Shine, S. Stromatas, H. Yamashita, and F. Yin, "A concept for multi-criteria environmental assessment of aircraft trajectories," *Aerospace*, vol. 4, no. 3, p. 42, 2017.

[15] S. Matthes, L. Lim, U. Burkhardt, K. Dahlmann, S. Dietmüller, V. Grewe, A. S. Haslerud, J. Hendricks, B. Owen, G. Pitari, M. Righi, and A. Skowron, "Mitigation of non-CO_2 aviation's climate impact by changing cruise altitudes," *Aerospace*, vol. 8, no. 2, 2021.

[16] H. K. Ng, B. Sridhar, S. Grabbe, and N. Chen, "Cross-polar aircraft trajectory optimization and the potential climate impact," in *IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, pp. 3D4–1.

[17] S. E. Campbell, M. B. Bragg, and N. A. Neogi, "Fuel-optimal trajectory generation for persistent contrail mitigation," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1741–1750, 2013.

[18] F. Yin, V. Grewe, C. Frömming, and H. Yamashita, "Impact on flight trajectory characteristics when avoiding the formation of persistent contrails for transatlantic flights," *Transportation Research Part D: Transport and Environment*, vol. 65, pp. 466–484, 2018.

[19] J. Rosenow, S. Förster, M. Lindner, and H. Fricke, "Multicriteria-optimized trajectories impacting today's air traffic density, efficiency, and environmental compatibility," *Journal of Air Transportation*, vol. 27, no. 1, pp. 8–15, 2019.

[20] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[21] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT* : Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[22] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms," in *Robotics Science and Systems VI*, 2010.

[23] J. Gammell, S. Srinivasa, and T. Barfoot, "Bit* : Batch informed trees for optimal sampling-based planning via dynamic programming on implicit random geometric graphs," in *2015 IEEE International Conference on Robotics and Automation*, 2015.

[24] S. M. LaValle, J. H. Yakey, and L. E. Kavraki, "A probabilistic roadmap approach for systems with closed kinematic chains," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 3. IEEE, 1999, pp. 1671–1676.

[25] S. M. LaValle, J. J. Kuffner, B. Donald *et al.*, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and computational robotics: new directions*, vol. 5, pp. 293–308, 2001.

[26] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast Marching Tree: a fast marching sampling-based method for optimal motion planning in many dimensions," *arXiv:1306.3532 [cs]*, 2015. [Online]. Available: http://arxiv.org/abs/1306.3532

[27] B. Graver, D. Rutherford, and S. Zheng, "CO_2 emissions from commercial aviation, 2013, 2018, and 2019." [Online]. Available: https://theicct.org/wp-content/uploads/2021/06/CO2-commercial-aviation-oct2020.pdf

[28] V. Williams, R. B. Noland, and R. Toumi, "Reducing the climate change impacts of aviation by restricting cruise altitudes," *Transportation Research Part D: Transport and Environment*, vol. 7, no. 6, pp. 451–464, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920902000135

[29] "Flightradar24: Live Flight Tracker - Real-Time Flight Tracker Map." [Online]. Available: https://www.flightradar24.com

[30] Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., and Thépaut, J-N. ERA5 hourly data on pressure levels from 1940 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS). [Online]. Available: https://doi.org/10.24381/cds.bd0915c6

[31] S. Dietmüller, S. Matthes, K. Dahlmann, H. Yamashita, A. Simorgh, M. Soler, F. Linke, B. Lührs, M. M. Meuser, C. Weder, V. Grewe, F. Yin, and F. Castino, "A python library for computing individual and merged non-CO_2 algorithmic climate change functions: Climaccf v1.0," *Geoscientific Model Development*, vol. 16, no. 15, pp. 4405–4425, 2023. [Online]. Available: https://gmd.copernicus.org/articles/16/4405/2023/

[32] (2024) Airbus A-388 | SKYbrary aviation safety. [Online]. Available: https://skybrary.aero/aircraft/a388

[33] J. Sun, J. M. Hoekstra, and J. Ellerbroek, "Openap: An open-source aircraft performance model for air transportation studies and simulations," *Aerospace*, vol. 7, no. 8, p. 104, 2020. [Online]. Available: https://doi.org/10.3390/aerospace7080104