Hybrid metaheuristic optimization algorithm for strategic planning of 4D aircraft trajectories at the continent scale

Supatcha Chaimatanan, Daniel Delahaye, Marcel Mongeau

A hybrid metaheuristic optimization algorithm for strategic planning of 4D aircraft trajectories at the continental scale

Supatcha CHAIMATANAN
MAIAA Laboratory, École nationale de l'aviation civile, Toulouse, France
Email: supatcha@recherche.enac.fr
Daniel DELAHAYE
MAIAA Laboratory, École nationale de l'aviation civile, Toulouse, France
Email: daniel@recherche.enac.fr
Marcel MONGEAU
MAIAA Laboratory, École nationale de l'aviation civile, Toulouse, France
Email: mongeau@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. In this work, we present a strategic trajectory planning methodology which aims to minimize interaction between aircraft at the European-continent scale. In addition, we propose a preliminary study that takes into account uncertainties of aircraft positions in the horizontal plane. 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. In addition, we present a computationally-efficient interaction detection method for large trajectory sets. The proposed methodology is successfully implemented and tested on a full-day simulated air traffic over the European airspace, yielding to an interaction-free trajectory plan.

Keywords

Strategic planning, 4D aircraft trajectory, mixed-integer optimization, hybrid-metaheuristic optimization

---

This work has been supported by 4 Dimension Contract-Guidance and Control (4D CO-GC) project, and by the French National Research Agency (ANR) through COSINUS program (project ID4CS n° ANR-09-COSI-005) and through JCJC program (project ATOMIC n° ANR 12-JS02-009-01).

I. Introduction

Air traffic management (ATM) is a system that assists and guides aircraft from a departure aerodrome to a destination aerodrome in order to ensure their safety, while minimizing delays and airspace congestion. According to standard separation requirements [1], aircraft that operate in the Terminal Maneuvering Area[1] (TMA) environment are required to be vertically separated by at least N_v = 1,000 feet (ft), and to be horizontally separated by a minimum of $N_{h_{TMA}}$ = 3 nautical miles (NM). In the en-route[2] environment up to FL[3]410, the minimum horizontal separation is increased to N_h = 5 NM [1].

Aircraft are considered to be *in conflict* when such a *minimum separation* requirement is violated. In other words, each aircraft has a protection zone (or protection volume) defined by a three-dimensional cylinder, as shown in Figure 1, in which other aircraft are not allowed to enter. A conflict situation does not necessary lead to a collision; however, it is a situation that controllers must avoid.

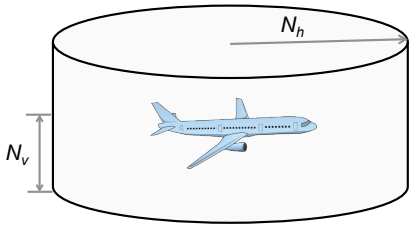

Figure 1: The *separation-norm* cylinder.

To ensure such a separation between aircraft, the airspace is partitioned into different *sectors* and a group of controllers is assigned to each sector. The controllers are responsible for maintaining aircraft at a safe distance from each other by applying *separation rules*. As the air-traffic demand keeps on growing, each controller must handle more and more aircraft.

To cope with increasing air-traffic demand, the ATM paradigm in Europe is being transformed towards a concept of Trajectory Based Operations (TBOs). This is the concern of the Single European Sky ATM Research (SESAR) project, which is a major collaborative project aiming at modernizing the European ATM system. In this new ATM paradigm, an aircraft flying through the airspace will be required to follow a 4D (3 spatial dimension + time) trajectory defined by a sequence of 4D coordinates (x, y, z, t) with high precision [2]. In other words, aircraft will be required to arrive at a specific 3D spatial position at a specific time. In fact, the introduction of this new ATM concept permits to alleviate the controller's tactical workload by shifting it to the strategic phase, so that the separation of all aircraft is ensured. The concept of 4D strategic deconfliction that aims to generate conflict-free trajectories for aircraft from origin to destination airports is introduced in the Innovative Future Air Transport

[1]Terminal maneuvering area (or Terminal Control Area (TCA) in the U.S. and Canada) is the airspace around a major airport which is extending upward from the surface of the Earth up to 10,000 feet above mean sea level.

[2]Aircraft traveling between two airports is considered to be in the en-route airspace once it leaves the TMA zone.

[3]Flight level (FL) is a pressure altitude, expressed in hundreds of feet, e.g. altitude of 41,000 feet is referred to as FL 410.

System (IFATS) project [3] and the 4 Dimension Contract-Guidance and Control (4D CO-GC) project [4]. This 4D trajectory concept will reduce recourse to controllers' intervention during the tactical phase. As a result, more flights can be accommodated by the controllers in a given airspace at a given time.

In this work, we propose a methodology to address such a strategic planning of trajectories at a continent scale. To our knowledge, no other research work addresses globally the 4D strategic deconfliction problem for such a large-scale problem. Instead of trying to ensure aircraft separation by solving each conflict locally, we focus on minimizing the global *interactions* between trajectories. An interaction occurs when two or more trajectories have an effect on each other; for instance, when multiple trajectories occupy the same space at the same period of time. Given an initial set of trajectories, the approach proposed in this paper separates these trajectories by allocating an alternative route and an alternative departure time to each participating flight. This route/departure-time allocation problem is formulated under the form of mixed-integer optimization problem. The objective is to minimize the total number of interactions between trajectories during a full day of traffic over Europe.

In reality, aircraft position may be subjected to uncertainties due to external events such as wind conditions, external temperature, etc. To increase robustness of the trajectories we compute, we also introduce in this paper a preliminary study taking into account uncertainties on aircraft position in the trajectory optimization process. An optimal route and a departure time for each flight are obtained through an iterative process which relies on a hybrid-metaheuristic optimization algorithm. The proposed methodology is implemented and tested on one day of air-traffic data over the European airspace which involves more than 30,000 flights.

The following sections of this paper are organized as follows. Section II reviews previous related works on strategic trajectory planning. Section III introduces our mathematical model. Section IV proposes an efficient method for detecting interactions between aircraft trajectories in a large-scale context. Section V presents a hybrid-metaheuristic optimization algorithm which relies on simulated annealing and on a hill-climbing local-search method, to solve the problem. Finally, numerical results are presented in Section VI.

II. Previous related works and proposal outline

During recent years, numerous research works on the trajectory planning problem based on deterministic and metaheuristic optimization approaches have been conducted. A comparison of the different optimization methods used for air-traffic management is provided in [5]. In the strategic planning framework, aircraft trajectories can be separated in many different ways. One of the most used methods is to modify the departure time of aircraft. This is commonly referred to as *ground delay* or *ground holding*, and examples of related work are [6], [7].

Delaying aircraft on the ground is effective since it reduces fuel consumption due to the extra distance aircraft would otherwise have to fly to avoid congested areas. Furthermore, it avoids aircraft to absorb the requested delay in holding patterns, which also increase fuel consumption. However, with increasing demand, significant delays still have to be assigned to a large number of aircraft in order to meet all airspace sector capacity constraints.

Another idea to separate trajectories is based on speed regulations; it is used for instance in [8] and [9]. Speed regulations introduce additional degree of freedom to the trajectory design. However, it requires numerous extensive and fine-tuned computation, which is not suitable to implement in a large-scale problem.

Other commonly-used strategies to separate aircraft trajectories rely on modifying the shape of trajectories (re-routing), or proposing alternative flight levels, or a combination of any of the above-mentioned methods. The

optimization problem that concentrates on combining ground holding and re-routing is referred to as the *air-traffic flow management rerouting problem*. This problem is shown to be NP hard[4] in [11] where optimal ground-holding times and routes are obtained by solving a 0-1 integer model taking into account airspace sector capacity. To simplify the problem, instead of considering individually each trajectory, several studies focus on separating the flow of trajectories. For instance, [12] addresses large-scale (one day traffic over France) air-traffic flow problems via a flow-based trajectory allocation, where the optimal separated 3D trajectory is obtained using A* algorithm or a global search strategy genetic algorithm (GA).

In [13] and [14], integer optimization approaches are used to allocate ground delays and rerouting options to trajectory flows taking into account airspace sector capacity constraints. In [15], a ground-holding is assigned to each aircraft and an optimal flight level is subsequently allocated to each flow of aircraft using constraint programming.

Despite advantages in terms of reduced computation time, the flow-based trajectory planning cannot separate aircraft that belong to the same flow of trajectories. The authors of [16] introduce a mixed-integer programming model to minimize traveling time, operating/fuel cost, air/sound pollutions subjected to separation and technical constraints. Their decision variables are the arcs and nodes (in a 3D-mesh network), speeds and departure/arrival times for each flight. The problem is solved by an exact deterministic method on instances limited to problem involving 10 flights. Further works that focus on managing each individual flight, but in large problems, can be found for instance in [17] and [18]. In these works, congestion in the airspace sectors is minimized by allocating to each flight optimal departure times and alternative routes (based on route-beacon navigation) using GAs. The results show advantages of using the route/departure-time allocation technique to alleviate airspace congestion and also show that GA is very efficient in solving highly complex problems. Nevertheless, the conflicts between aircraft trajectories are not managed. Moreover, GA is not well adapted for large-scale 4D trajectory planning due to excessive memory requirement intrinsic to population-based optimization algorithms whose performance depends on the population size. The reader interested in a survey on modeling and optimization in air traffic is referred to the recent book [19] where several problems are addressed using GA.

In [20], the authors use a ground-holding method to solve 4D trajectory deconfliction problems. Their objective is to minimize the overall delays while respecting the conflict constraints. The authors rely on pairwise comparisons to detect conflicts between trajectories. The proposed algorithm is able to address a problem involving up to 9,500 flights. However, relying only on the departure-time adjustment, the proposed method must allocate significant delays in order to solve all the conflicts. Moreover, the pairwise conflict detection algorithm is not well adapted for continent-scale problems due to excessive computation time requirements.

Uncertainty of aircraft velocity is taken into account in the 4D trajectory deconfliction problem of [21]. Instead of a point, the aircraft position at any given time is represented by a bounded 3D envelope whose size grows with time. In [22], the authors propose a 4D trajectory deconfliction method using a light-propagation algorithm. Uncertainty of aircraft trajectory in the time domain is considered by modeling the aircraft arrival time to a given point as a time segment. The uncertainty increases the difficulty of the problems, and it reduces the solution space.

[4]According to the famous NP $\neq$ P conjecture, a Non-deterministic Polynomial-time hard (NP hard) problem cannot be solved in polynomial time (with respect to the size of the instance). The reader interested in complexity of decision problems and combinatorial optimization problems is referred for example to [10].

In these works, the authors simplify the problem by solving conflicts iteratively over short moving time windows. This technique is effective for problems involving air traffic over a short period of time (e.g. tactical deconfliction) but cannot be applied in our strategic deconfliction context.

In [23] and [24], preliminary studies on individual 4D trajectory optimization are presented. In these papers, optimal 4D trajectories for individual flights were allocated by solving a combinatorial optimization problem using a non-population based hybrid-metaheuristic optimization method. The numerical results presented in [24] show that an integration of a simple greedy local search into a classical simulated-annealing method can significantly decrease the computation time (10 times less computation time than using simulated annealing alone) for small traffic instances ($\approx$ 4,000 flights). However the discretization of the search domain (possible departure times and alternative trajectories) induces high combinatorics.

In this paper, we contribute to the area of air traffic management in the framework of the future ATM paradigm. More precisely, we propose an alternative, mixed-integer programming formulation of the strategic trajectory planning problem presented in [24]. Due to the complexity of the problem, we concentrate on a single objective function: to minimize the total interaction between trajectories. We improve the hybrid-metaheuristic optimization algorithm proposed in [24] by introducing a new intensification local-search step. Moreover, we describe a computationally efficient hash-table based method, which does not rely on pair-wise comparisons, for detecting interaction between trajectories. We introduce a preliminary step to take into account uncertainty of the aircraft position in the horizontal plane. To the best of our knowledge, no other research work considers solving globally conflicts between aircraft trajectories, taking into account uncertainty, for such a large-scale traffic. Finally, we prove the viability of the overall methodology on large-scale air-traffic data on the European-continent airspace, including the traffic in the terminal maneuvering area (TMA).

III. Mathematical modeling

This section presents the mathematical model used to describe our strategic trajectory-planning methodology. First, the assumptions and simplifications that are made in this work are presented. Then, uncertainty of aircraft position is characterized. Next, a definition of *interaction* between trajectories is given. Then, the route/departure-time allocation techniques adapted for strategic trajectory-planning are described. Finally, a mathematical formulation of the interaction minimization problem is introduced.

In this work, the following assumptions and simplifications are made. The airspace is considered as a Euclidean space. Latitudes and longitudes on the earth surface are transformed into (x, y) coordinates by a Lambert azimuthal projection with the center of projection located at the center of the given airspace. The altitude, in feet, will be represented by the z coordinate. The flight level and altitude profile of the initial discretized 4D trajectory is assumed to be optimal (for the airliner). A given (initial) trajectory is composed of three parts; the *initial en-route segment* is the shortest possible route between the origin and the destination airports (*great circle path*) and the two (extremity) TMA parts of the trajectory.

A. Uncertainties of aircraft position

In reality, aircraft are subjected to unpredicted external events (such as wind, external temperature, etc.) which cause uncertainties on aircraft positions with respect to their planned 4D trajectory. In this preliminary study, we

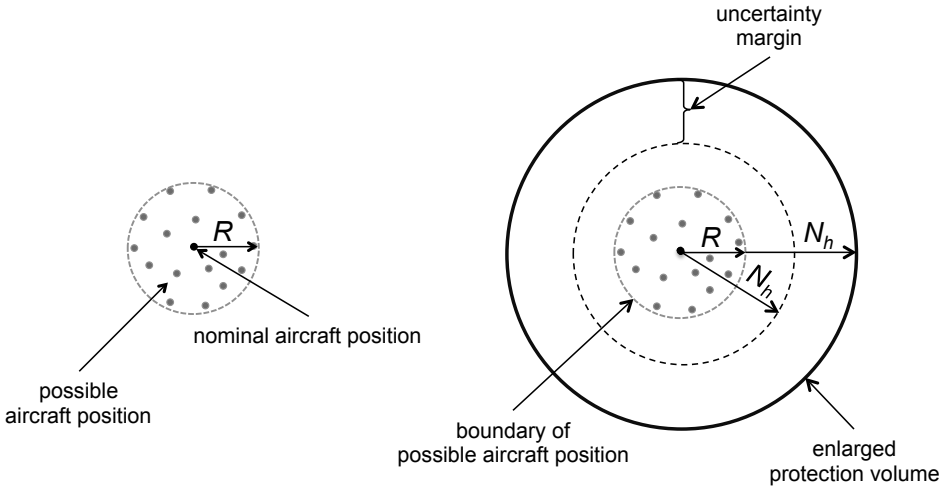

Figure 2: Left: Possible aircraft positions in presence of uncertainties. Right: Enlarged protection volume and uncertainty margin.

consider only uncertainties of aircraft positions in the horizontal plane, assuming that aircraft can attain a given altitude z with high precision. Consider a given set of N discretized 4D trajectories, where each trajectory i is a time sequence of 4D coordinates, $P_{i,k}(x_{i,k}, y_{i,k}, z_{i,k}, t_{i,k})$, specifying that aircraft must arrive at a given point $(x_{i,k}, y_{i,k}, z_{i,k})$ at time $t_{i,k}$, for $k = 1, \ldots, K_i$, and K_i is the number of *sampling points* of trajectory i.

Due to uncertainties, we shall assume that the real (horizontal) position, $(x^r_{i,k}, y^r_{i,k})$, of the aircraft at time $t_{i,k}$ can be in an area defined by a disk of radius R around $(x_{i,k}, y_{i,k})$, as illustrated in Figure 2. Thus, the possible locations of the aircraft at time t are the elements of the set:

$$\{(x^r_{i,k}, y^r_{i,k}) : (x^r_{i,k} - x_{i,k})^2 + (y^r_{i,k} - y_{i,k})^2 \leq R^2\}. \tag{1}$$

To ensure separation of aircraft subjected to such uncertainties, the protection volume has to be enlarged by a radius of R, as illustrated in Figure 2. Thus, the (robust) minimum separation, N^r_h, in the horizontal plane becomes:

$$N^r_h = N_h + R, \tag{2}$$

where N_h is the minimum separation in the case without uncertainty. This enlargement of the protection volume allows an aircraft to follow its 4D trajectory plan with some margin. It can arrive at a given point with some deviation while the minimum interaction condition is still satisfied. This margin will be called the *uncertainty margin* in the remaining of this paper.

In order to simplify the presentation, we shall consider in the remaining subsections of the paper the case without uncertainty ($R = 0$). However, in the numerical results section, experiments are also conducted for the case with uncertainty.

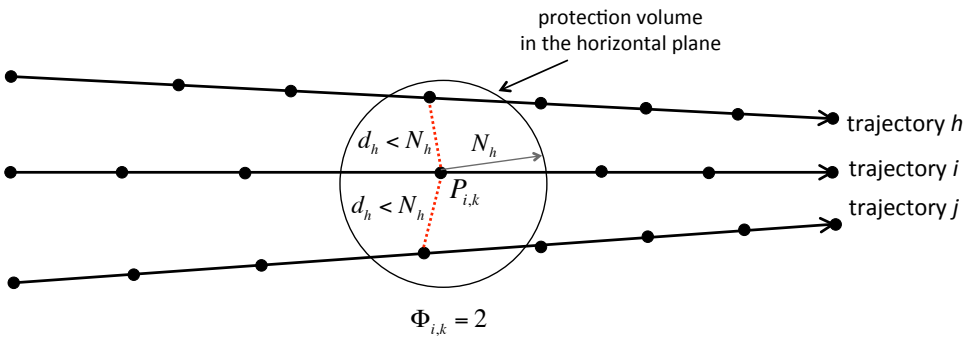

Figure 3: Interactions, $\Phi_{i,k}$, at sampling point $P_{i,k}$ of trajectory i.

B. Interaction between trajectories

Interaction between trajectories is, roughly speaking, a situation which occurs in the planning phase, when more than one trajectory compete for the same space at the same period of time. It is different from the *conflict* situation, which corresponds simply to a violation of the minimum *separation* (i.e., 5 NM horizontally and 1,000 ft vertically). Additional separation conditions, such as time separation, topology of trajectory intersection, distance between trajectories, etc., can also be taken into account in the concept of interaction.

For the sake of simplicity, let us first rely on the validation of the minimum separation for the measurement of interactions. Thus, minimizing interaction between trajectories, in this particular case, boils down to minimizing the number of conflicts between aircraft.

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

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

$$\Phi_i = \sum_{k=1}^{K_i} \Phi_{i,k}. \tag{3}$$

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

$$\Phi_{tot} = \sum_{i=1}^{N} \Phi_i = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \Phi_{i,k}. \tag{4}$$

C. Route/departure-time allocation

The objective of this work is to allocate an alternative trajectory and an alternative departure time for each aircraft in order to minimize the total interaction between trajectories, taking into account uncertainty of aircraft position in the horizontal plane. This paper focuses on the strategic level (planning one day in advance); the interaction reduction problem can therefore be solved simultaneously on both the spatial and the temporal dimensions.

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

Alternative departure time. The departure time of each flight can be shifted by a positive (delay) or a negative (advance) time shift. Let $\delta_i \in \Delta_i$ be a departure time shift attributed to flight i, where Δ_i is a set of acceptable

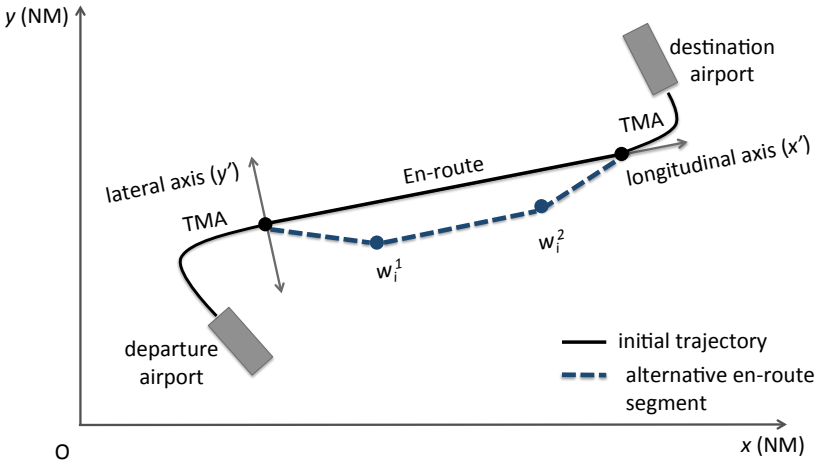

Figure 4: Initial and alternative trajectories with $M = 2$ virtual waypoints.

time shifts for flight i. The departure time t_i of flight i is therefore $t_i = t_{i,0} + \delta_i$, where $t_{i,0}$ is the initially-planned departure time of flight i.

Alternative trajectory design. To respect the given optimal cruise level, altitude profile, and standard departure and arrival procedures, in this work we concentrate on modifying only the *horizontal* profile of the en-route segment of a given 4D trajectory.

In this work, an alternative trajectory is constructed by placing a set of virtual waypoints near the initial en-route segment and then by reconnecting the successive waypoints with straight-line segments.

We call *longitudinal axis* (x') the axis that is tangent to the initial en-route segment, and the *lateral axis* (y') is the axis that is perpendicular to the longitudinal axis. The position of each waypoint will be defined using these relative $x'y'$-reference axes. Let $w_i^m = (w_{ix'}^m, w_{iy'}^m)$ be the m^{th} virtual waypoint of trajectory i, where $w_{ix'}^m$ and $w_{iy'}^m$ are the longitudinal and lateral components of w_i^m respectively. We define, for each flight i, a set of virtual waypoints (optimization variables) used to control the trajectory shape of flight i, denoted w_i can be represented as $w_i = \{w_i^m | w_i^m = (w_{ix'}^m, w_{iy'}^m)\}_{m=1}^M$, where M denotes the number of virtual waypoints that the user is allowed to introduce. In Figure 4, a dashed line illustrates an alternative en-route profile constructed with $M = 2$ virtual waypoints.

Remark that the alternative trajectory will yield an increase in flight duration when compared with the initial trajectory. To compensate this increased flight duration, the altitude profile must be updated to avoid a premature descent. Let T_{ext} be the increased flight duration. In the case of a regional flight whose all flight phases (departure, climb, cruise, descent, and arrival) are executed in the considered airspace sector, the altitude profile is updated by extending the cruise phase (constant-level) at the top of descent (TOD) for a duration T_{ext}, as illustrated in Figure 5.

Besides, for a flight whose origin or destination airport is outside of the current airspace sector, the cruise phase can take place inside or outside the current airspace sector. This yields six possible configurations of the initial

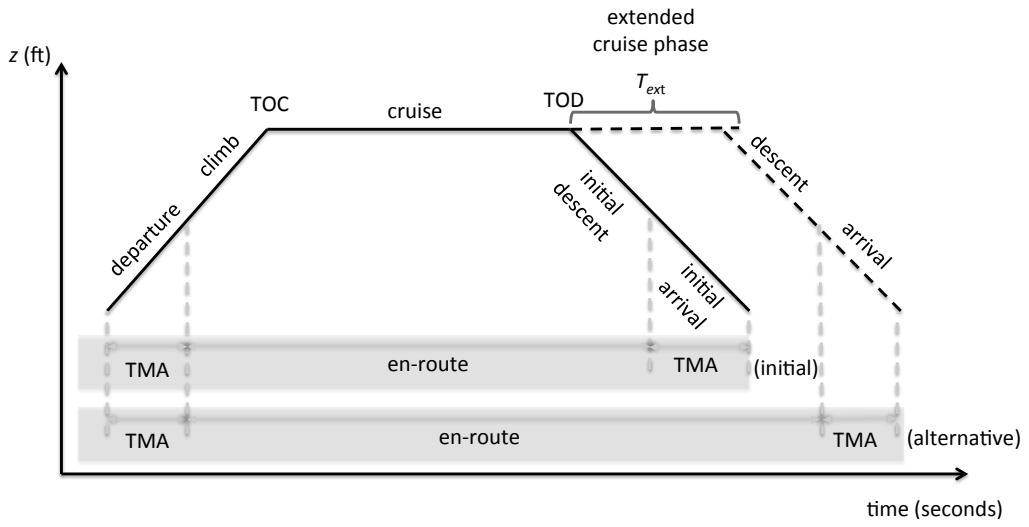

Figure 5: Altitude profile update: extending cruise phase at the top of descent (TOD).

altitude profile, as illustrated in Figure 6. Let z_{max} be the maximum altitude that the flight will attain in the current airspace sector. In this case, the vertical profile can be updated by extending the flight with a constant altitude z_{max} for a duration T_{ext} according to which of the configurations of the initial altitude profile is relevant (the aim is to preserve the given optimal profile and the same climb/descent slopes).

D. Mixed-integer programming formulation

We formulate here the strategic trajectory-planning problem as an optimization problem.

Objective. Find departure time shifts and alternative en-route trajectories that reduce the total interaction between the aircraft trajectories.

Given data. A problem instance is given by:

- A set of initial (nominal) N discretized 4D trajectories;
- The number of allowed virtual waypoints, M;
- The maximum allowed advance departure time shift of each flight i, $\delta_a^i < 0$;
- The maximum allowed delay departure time shift of each flight i, $\delta_d^i > 0$;
- The maximum allowed route length extension coefficient of each flight i, $0 \leq d_i \leq 1$;
- The length of the initial en-route segment of each flight i, $L_{i,0}$.

Decision variables. To separate trajectories in the time domain, a departure-time shift δ_i is associated to each flight i. In addition, a vector, w_i, of virtual waypoint locations, $w_i = (w_i^1, \ldots, w_i^M)$, is associated to each flight to separate trajectories in 3D space. Let us set the compact vector notation $\mathbf{w} = (w_1, \ldots, w_N)$ and $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)$. Therefore, the decision variables of our route/departure-time allocation problem can be represented by $(\mathbf{w}, \boldsymbol{\delta})$.

Constraints. The above optimization variables must satisfy the following constraints:

Allowed departure time shift. Since it is not reasonable to delay or advance departure time of flight for too long, the departure time shift δ_i will be limited to lie in the interval $\Delta_i := [\delta_a^i, \delta_d^i]$. Common practice in airports

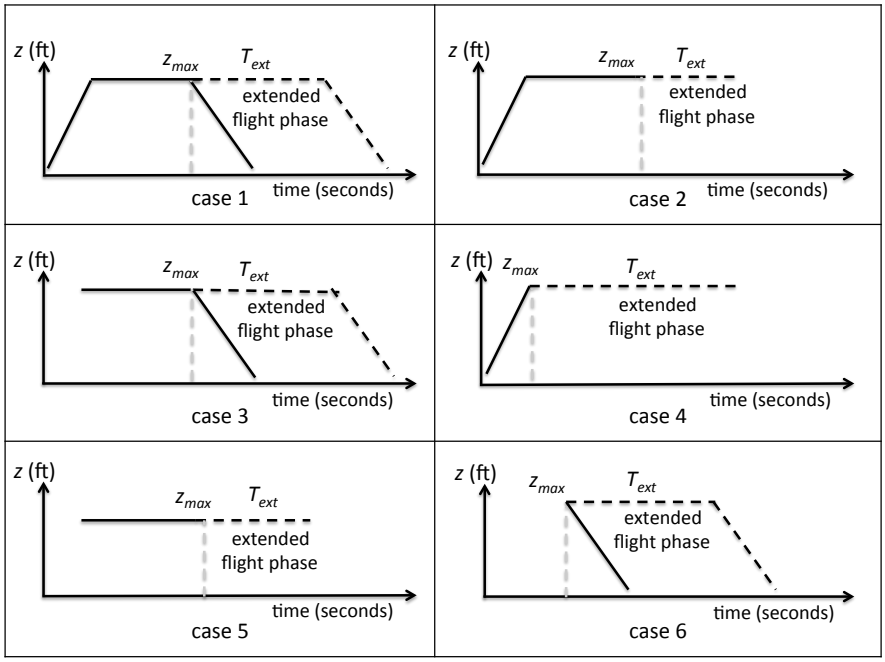

Figure 6: Six possible configurations of the initial altitude profile of flight, whose origin or destination airports are outside of the current airspace sector, and the ways to update them.

conducted us to rely on a discretization of this time interval. Given a time-shift step size δ_s (to be set by the user), this yields $n_a^i := \frac{-\delta_a^i}{\delta_s}$ possible advance slots and $n_d^i := \frac{\delta_d^i}{\delta_s}$ possible delay slots of flight i. Therefore, the set, Δ_i, of all possible departure time shifts of flight i is given by

$$\Delta_i = \{-n_a^i\delta_s, -(n_a^i - 1)\delta_s, \ldots, -\delta_s, 0, \delta_s, \ldots, (n_d^i - 1)\delta_s, n_d^i\delta_s\}. \tag{5}$$

Maximal route length extension. The alternative trajectory induces route length extension which causes an increase of fuel consumption. Therefore, it should be limited so that it is acceptable for the airline. Let d_i be the maximum allowed route length extension coefficient of flight i (to be set by the user). To restrain the route length extension, the alternative en-route profile of flight i must satisfy:

$$L_i(w_i) \leq (1 + d_i)L_{i,0}, \tag{6}$$

where $L_i(w_i)$ is the length of the alternative en-route profile determined by w_i. This constraint can be satisfied by restricting the set of possible waypoint locations (as will be described below).

Allowed waypoint locations. To limit the search space, to prevent undesirable sharp turns, and to restrain the route length extension, we bound the possible location of each virtual waypoint. For simplicity, for each trajectory i and each virtual waypoint m, the lateral component, $w_{iy'}^m$, is restricted to lie in the interval:

$$W_{iy'}^m := [-a_i L_{i,0}, a_i L_{i,0}]. \tag{7}$$

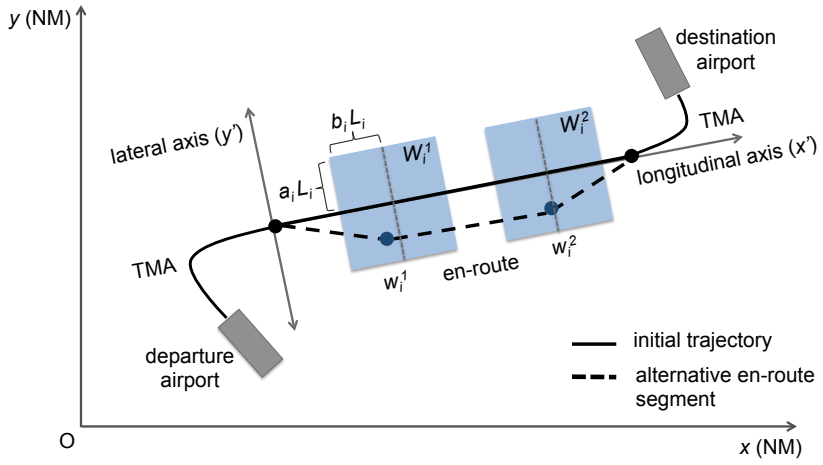

Figure 7: Rectangular-shape possible location of $M = 2$ virtual waypoints.

Similarly, the longitudinal component, $w_{ix'}^m$, is set to be in a range:

$$W_{ix'}^m := \left[\left(\frac{m}{1+M} - b_i \right) L_{i,0}, \left(\frac{m}{1+M} + b_i \right) L_{i,0} \right], \tag{8}$$

where $0 \leq a_i \leq 1$ and $0 \leq b_i \leq 1$ are user-defined coefficients that satisfy (6). This yields a rectangular shape for the possible locations of the virtual waypoint w_i^m, as illustrated in Figure 7. To obtain a regular trajectory, the longitudinal component of two adjacent waypoints must not overlap, i.e.,

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

and hence the user should choose b_i so that $b_i < \frac{1}{2(M+1)}$. Finally, to restrain the maximum route extension, a_i and b_i must be chosen so that:

$$\max\{L_i(w_i) | w_i \in W_{ix'} \times W_{iy'}\} \leq (1 + d_i) L_{i,0}. \tag{9}$$

Objective function. One wishes to determine the values for the optimization variables w_i and δ_i for each flight $i = 1, \ldots, N$ so as to minimize interaction between N flights: $\Phi_{tot}(\mathbf{w}, \boldsymbol{\delta})$. The problem can be represented as follows:

$$\min_{\mathbf{w}, \boldsymbol{\delta}} \Phi(\mathbf{w}, \boldsymbol{\delta})$$

subject to

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

where $\Phi(\mathbf{w}, \boldsymbol{\delta})$ is defined by (4), and Δ_i, $W_{iy'}^m$, and $W_{ix'}^m$ are defined by (5), (7), and (8) respectively. The complexity of the problem (P1) is discussed in [24]. The mathematical model (P1) involves mixed-integer variables introducing high combinatorics to the search space. Since we have $W_{ix'}^m$ and $W_{iy'}^m \subseteq \mathbb{R}^2$, we have $\mathbf{w} \in \mathbb{R}^{2NM}$.

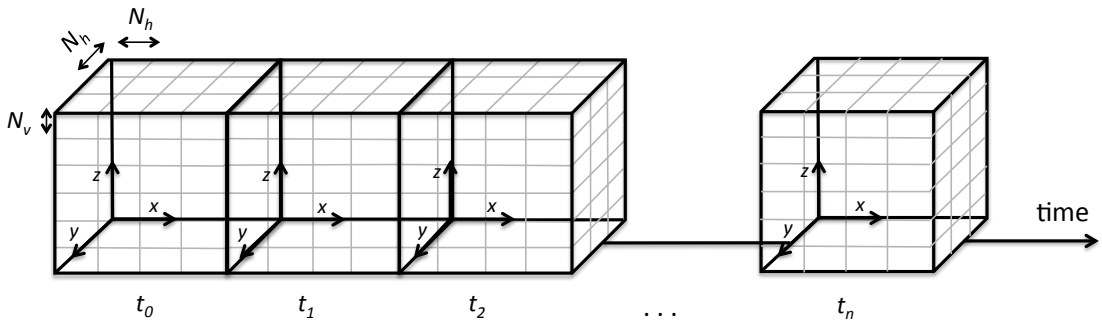

Figure 8: Four-dimension (3D space + time) grid for conflict detection, illustrated as a time series of 3D grids which is sampled with discretization time step $\Delta t = t_n - t_{n-1}$. The size of each cell in the 3D grids is defined by the minimum separation requirements (N_h and N_v).

The discrete variable feasible set has cardinality $|\Delta_i| = \frac{|\delta_a^i| + |\delta_d^i|}{\delta_s} + 1$, where δ_a^i is the maximum allowed advance time shift, δ_d^i is the maximum allowed delay time shift, and δ_s is the time-shift step size. The objective function of problem (P1) is non-separable, because each term $\Phi_{i,k}$ does not depend solely on variables w_i and δ_i; it is also affected by neighboring trajectories. The evaluation of the objective function involves a heavy computational burden in practice, as will be seen in the sequel of the paper, where we consider the continental scale. Besides, the objective function may feature several equivalent optima (multimodal). This interaction minimization problem based on route/departure-time assignment technique is therefore sufficiently difficult to motivate recourse to a stochastic methods for optimization.

IV. Interaction detection method

In order to evaluate the objective function, at a candidate solution, $(\mathbf{w}, \boldsymbol{\delta})$, one needs to compute interaction between the N aircraft trajectories. We present here a method to detect any violation of the protection volume. To avoid the $\frac{N(N-1)}{2}$ time-consuming pair-wise comparisons, which are prohibitive in our continental-scale application context, we propose a grid-based interaction detection scheme which is implemented in a so-called *hash table* as presented in [24].

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

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

An example of conflict detection in the horizontal plane can be seen in Figure 9, where the nine neighboring cells of a given trajectory sampled point are checked, and the distance between the given sampled point and the

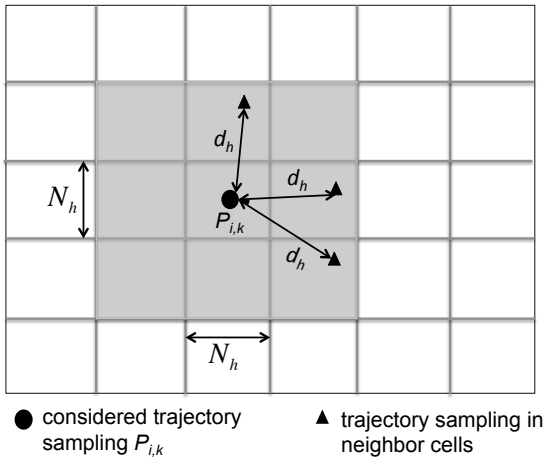

Figure 9: Conflict detection in the horizontal plane: Verifying the nine neighboring cells for separation requirements.

neighbor-cell sampled point (corresponding to another trajectory) are measured. Since the violation of the protection volume can only occur when the points in question are in the same or in adjacent grid cells, the number of points to check is significantly smaller than in a pair-wise comparison method.

In order not to underestimate interaction, trajectories must be discretized with a sufficiently small sampling-time step, Δt, which depends on the maximum possible aircraft horizontal and vertical speeds. As stated in [15], the worst-case scenario for interaction detection in the horizontal plane occurs when two aircraft follow parallel trajectories that are separated by a distance, D, less than or equal to the horizontal separation norm, N_h, at maximum horizontal speed, $V_{h_{max}}$, with heading in opposite directions. Hence, in the horizontal plane, undetected interaction can occur when: $\Delta t > \frac{N_h}{V_{max}} \cos\left(\arcsin\left(\frac{D}{N_h}\right)\right)$. In the vertical plane, the worst-case scenario occurs when one aircraft is climbing at a maximum rate of climb, RoC_{max}, and another is descending at maximum rate of descent, RoD_{max}. Thus, in the vertical plane, in an analogical way as what was done in [15], undetected interaction can occur when: $\Delta t > \frac{N_v}{(RoC_{max}+RoD_{max})}$.

One can therefore simply choose a sufficiently small value of Δt. However, using a small sampling-time step leads to a large number of trajectory samples, which therefore requires more computation time and memory. Instead, we propose an inner-loop algorithm, called `interp`, detecting interaction between two sampling times, t and $t + \Delta t$, by *interpolating* aircraft positions with a sufficiently small step size, t_{interp}. Then, one checks each pair of these interpolated points. The algorithm stops when an interaction is identified or when every pair of the interpolated points has been checked. The inner-loop interpolation algorithm, `interp`, is described in Figure 10.

V. Hybrid-metaheuristic optimization algorithm

The strategic trajectory planning methodology we are presenting relies on the interaction minimization problem introduced in Section III whose objective function values are obtained by black-box simulation through the interaction detection scheme developed in Section IV. In this work, we present a hybrid metaheuristic approach

Algorithm Interp

Require: $P_{i,k}$, $P_{i,k+1}$, $P_{j,l}$, $P_{j,l+1}$

Discretize, using time step t_{interp}, the trajectory segments $[P_{i,k}, P_{i,k+1}]$ and $[P_{j,l}, P_{j,l+1}]$ as $\{P_\alpha\}_{\alpha=1}^{K}$ and $\{Q_\beta\}_{\beta=1}^{K}$ respectively;
Set isInteract = False;
for $k = 0 \rightarrow K$ **do** ▷ for each pair of interpolated points
    Measure distance, d_h and d_v, between P_k and Q_k;
    if $d_h < N_h^r$ and $d_v < N_v$ **then**
        isInteract = True;
        Return isInteract;
        End;
    end if
end for
Return isInteract;
End;

Figure 10: Inner-loop interpolation algorithm for detecting interactions between two sampling time steps, by interpolating aircraft positions with a small time step size, t_{interp}.

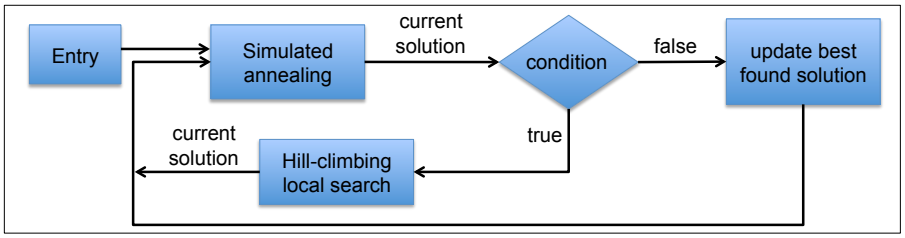

Figure 11: Structure of the proposed hybrid algorithm of simulated annealing and hill-climbing local search methods.

adapted to handle an air-traffic assignment problem at the continent scale involving more than 30,000 trajectories, each of which is described by around 500 sample points. It relies on a classical simulated annealing algorithm and two different hill-climbing local-search modules. The local search allows the system to intensify the search around a potential candidate solution while the simulated annealing allows the system to escape from a local trap and thereby ensuring the exploration of the solution space. The proposed hybrid algorithm combines the SA and the local search algorithm such that the local search is considered as an inner-loop of the SA, which will be performed when a pre-defined condition is satisfied. The structure of the proposed hybrid algorithm of simulated annealing and hill-climbing local search methods is illustrated in Figure 11.

Simulated annealing was introduced by S. Kirkpatrick et al. in 1982 [25]. It is a metaheuristic stochastic method of optimization that is well known for its ability to escape from local minima by allowing occasional moves that deteriorate the value of the objective function, such deteriorating moves being less and less as the number of iterations grows. This algorithm is inspired by an annealing process in metallurgy where material at high-energy state is slowly cooled down according to a pre-described temperature reduction schedule until the material reaches a global-minimum energy state and forms a crystallized solid. Too rapid decrease of the temperature can, however, yield a non-desirable local minimal energy state. In the simulated annealing optimization algorithm, the objective

Algorithm Neighborhood function

Require: P_w, i.

Generate random number, $r :=$ random(0,1);
if $r < P_w$ **then**
    Choose new δ_i from Δ_i;
else
    Choose one virtual waypoint w_i^m to be modified.
    Choose new $w_{ix'}^m$ from $W_{ix'}^m$;
    Choose new $w_{iy'}^m$ from $W_{iy'}^m$;
end if

Figure 12: Neighborhood function.

function value is analogical to the energy of the physical problem while a control parameter, T, that decreases as the number of iteration grows, plays the role of the temperature schedule.

For our problem, the simulated annealing proceeds as follows. First, we evaluate the objective function at the current configuration $(\mathbf{w}, \boldsymbol{\delta})_C$. It is denoted Φ_C. Then a neighboring solution, $(\mathbf{w}, \boldsymbol{\delta})_N$, is generated by randomly choosing one flight to be modified. Then, a new solution for this chosen flight is generated according to a pre-defined neighborhood structure (to be described below). If the neighborhood solution improves the objective function value, then it is accepted. Otherwise, it is accepted with a probability $e^{\frac{-\Delta\Phi}{T}}$, where $\Delta\Phi = \Phi_N - \Phi_C$ is the difference of energy between current state C and new state N (*Metropolis algorithm*)[5]. When the maximum number of iterations, n_T, at a given temperature is reached, the temperature is decreased according to the user-provided pre-defined schedule, and the process is repeated until the pre-defined final temperature, T_{final}, is reached. More detail on simulated annealing can be found, for instance, in [27], [28].

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

Hill-climbing local search modules. The local search modules we use are heuristic methods that accepts a new solution only if it yields a decrease of the objective function. The process repeats until no further improvement can

[5]Metropolis algorithm is a Markov Chain Monte Carlo (MCMC) method that generates a sequence of random samples according to a pre-defined probability distribution [26].

be found or until the maximum number of iterations $n_{T_{LOC}}$ is reached. The two local-search modules correspond to the two following strategies:

- **Intensification of the search on one Particular Trajectory (PT).** Given a flight i, this state-exploitation step focuses on improving the current solution by applying a local change from the neighborhood structure only to flight i (only the decision variables (w_i, δ_i) are affected).
- **Intensification of the search on the Interacting Trajectories (IT).** Given a flight i, this state-exploitation step applies a local change, from the neighborhood structure, to every flight that is currently interacting with flight i. For instance, suppose that trajectory i interacts with trajectory p, q, and r. The changes are sequentially applied to the decision variables (w_p, δ_p), (w_q, δ_q), and (w_r, δ_r).

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

$$P_{SA}(T) = P_{SA,min} + (P_{SA,max} - P_{SA,min}) \cdot \frac{T_0 - T}{T_0}, \tag{10}$$

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

$$P_{Loc}(T) = P_{Loc,min} + (P_{Loc,max} - P_{Loc,min}) \cdot \frac{T_0 - T}{T_0}, \tag{11}$$

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

$$P_{SL}(T) = 1 - (P_{SA}(T) + P_{Loc}(T)). \tag{12}$$

A key factor in tuning this hybrid algorithm is to reach a good trade off between exploration (diversification) and exploitation (intensification) of the solution space, i.e. a compromise between fine convergence towards local minima and the computation time invested in exploring the whole search space.

The proposed hybrid algorithm is detailed in Figure 13 where T_{init} and T_{final} are respectively the initial and the final temperature of the (user-provided) cooling schedule, and n_T is the maximal number of iterations at each temperature step (set by the user).

VI. Numerical Results

The proposed hybrid-metaheuristic algorithm is implemented in Java and simulated on an AMD Opteron 2 GHz processor with 128 GB RAM. The problem instance we must solve is given by a data set consisting of a simulated full day air-traffic over the European airspace on July 1^{st}, 2011, involving $N = 30{,}695$ trajectories. The trajectory set is simulated using Base of Aircraft Data (BADA) aircraft performance model with optimal altitude profiles [29]. The proposed algorithm is first conducted without consideration of uncertainty $(R = 0)$. It is tested with the traffic data corresponding only to the en-route segments. Then, it is tested with the full traffic, taking also into account air-traffic in the terminal maneuvering area (TMA). Finally, computational experiments are conducted with the full traffic, taken into account uncertainty of aircraft position in the horizontal plane.

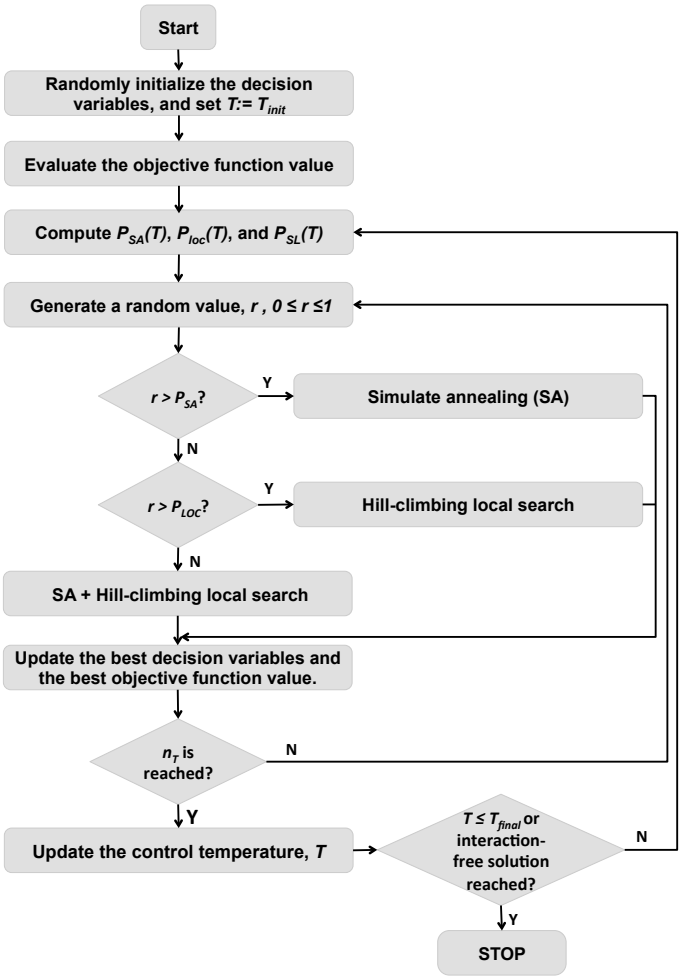

Figure 13: Hybrid algorithm of simulated annealing and hill-climbing local search methods.

The following values were chosen for the user-defined parameter: $M = 2$, $b_i = 0.1$. The maximum route extension can occur when $w_i^1 = ((\frac{1}{3} + b_i)L_{i,0},\ a_i L_{i,0})$ and $w_i^2 = ((\frac{1}{3} - b_i)L_{i,0},\ -a_i L_{i,0})$, and therefore a_i can be straightforwardly deduced from:

$$(1 + d_i) = 2\sqrt{\left(\frac{1}{3} - b_i\right)^2 + (a_i)^2} + \sqrt{\left(\frac{1}{3} + b_i\right)^2 + (2a_i)^2},$$

which yields $a_i = 0.125$. Empirical tests lead us to set $P_{SA,min} = 0.8$, $P_{SA,max} = 0.9$, $P_{Loc,min} = 0.4$, $P_{Loc,max} = 0.6$, $n_T = 3{,}500$ and $n_{T_{Loc}} = 5$. The initial temperature, T_{init}, is calculated using an algorithm proposed in [27]. It is computed by initiating 100 deteriorate disturbances at random; evaluate the average variations ($\Delta\Phi_{avg}$) of the objective function value; then deduce T_{init} from the relation: $e^{\frac{-\Delta\Phi_{avg}}{T_{init}}} = \tau_0$, where τ_0 is the initial rate of accepting degrading solutions whose value depends on the assumed quality of the initial configuration. Empirical test leads us to set $\tau_0 = 0.3$. The temperature is decreased according to a geometrical law $T_{k+1} = 0.99T_k$. The final temperature is set to $T_{final} = \frac{T_{init}}{1{,}000}$, yielding 688 temperature steps.

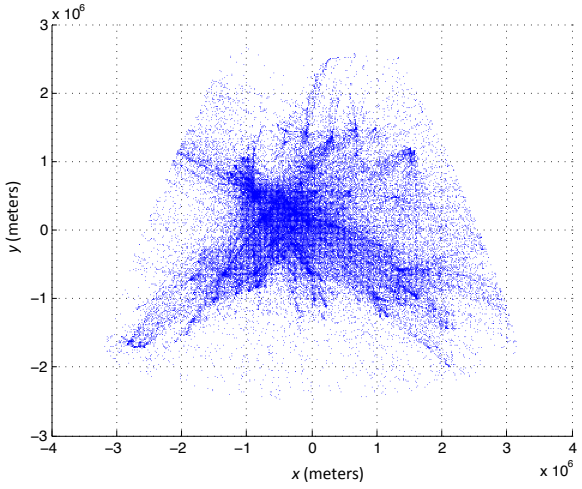

Figure 14: Filtered trajectory set with $\Delta t = 20$ seconds.

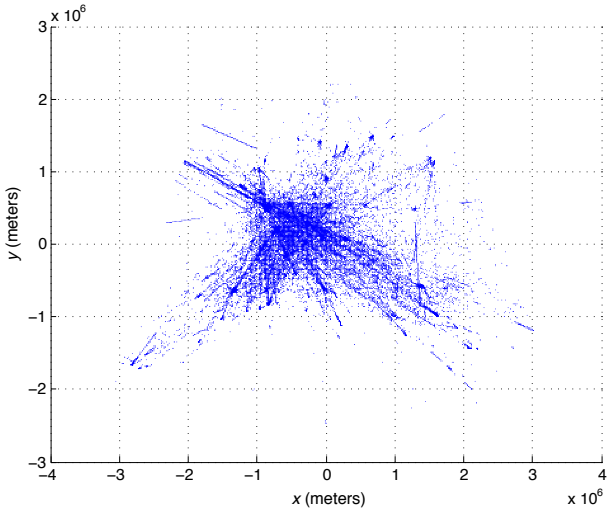

Figure 15: Interaction points of the initial traffic situation.

A. *En-route segment traffic*

Here, the trajectory set is filtered so that only the en-route segments that lie in the European airspace (latitude interval [30.0, 65.0], and longitude interval [-15.0, 40.0]) are considered. The filtered trajectory set consists of 29,843 trajectories. These trajectories are sampled in time using Δt = 20 seconds. Figure 14 shows the sampling points of the initial trajectory set.

The inner-loop interpolation time step, t_{interp}, is set to 5 seconds. The initial trajectory plan features $\Phi_{tot,init}$ = 178,168 interactions which involve 19,447 trajectories. Figure 15 shows the locations where these interactions occur. In order to carry out the simulation, the user-defined input parameters of the optimization algorithm are empirically tuned and set as follows (in coherence with air-traffic controllers' experience) for every trajectory i. The maximum allowed departure time shifts are set to $-\delta_a^i = \delta_d^i = 90$ minutes. The maximum allowed route length extension coefficient, d_i, is set to 12 $\%$.

First, the influences of modifying the trajectory shape (location of virtual waypoints) and the departure time on the resolution of the interaction reduction problem are studied. Simulations with different settings of probability, P_w, to modify the location of virtual waypoints (rather than modifying the departure time) are performed. In each such simulation, we keep the feasible area of the solution space (feasible virtual waypoint location and feasible departure time shift) identical. Each simulation is carried out 10 times. The final total interactions between trajectories $\Phi_{tot,final}$ are compared in Table I. The average number of iterations and the average computation time are compared in Figure 16. The average number of modified routes, and the average number of modified departure times are compared in Figure 17.

The results show that relying on only one degree of freedom ($P_w = 0.0$ and $P_w = 1.0$) is not sufficient to separate all trajectories. Moreover, it requires a large number of objective function evaluations which results in long computation time. Acting only in the temporal space ($P_w = 0.0$) is nevertheless more powerful than acting only in the 3D space ($P_w = 1.0$), yielding better solutions in significantly less number of objective function evaluations

Table I: Impact of the chosen value of probability, P_w, on the final total interaction between trajectories $\Phi_{tot,final}$.

P_w	0.0	0.25	0.50	0.75	1.00
$\Phi_{tot,final}$	32	0	0	0	1984

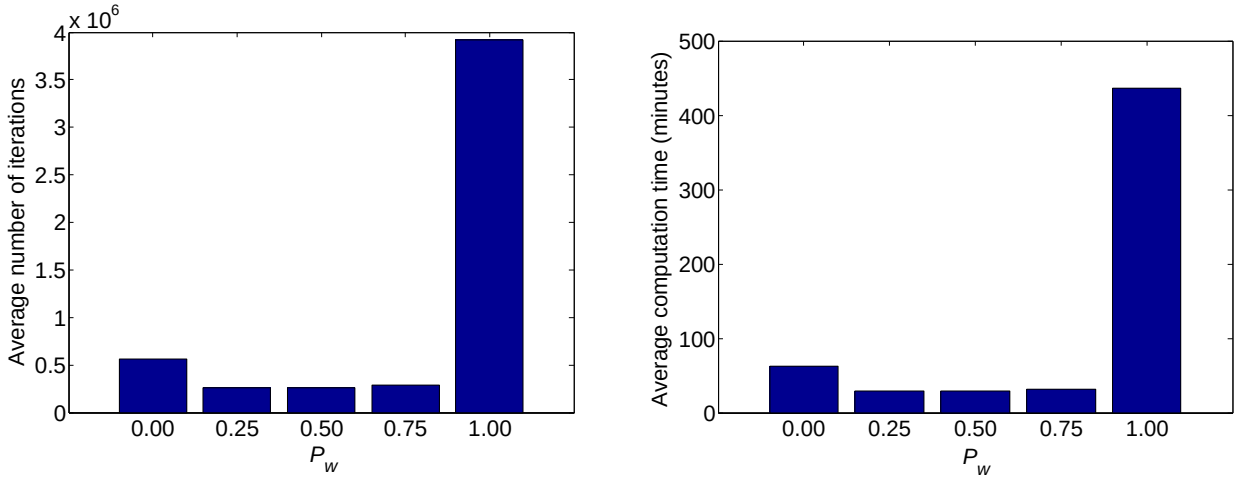

Figure 16: Impact of the chosen value of probability P_w on the average number of iterations (left) and on the average computation time (right).

(and therefore less computation time). However, relying solely on modifying departure time shifts is not sufficient to reach an interaction-free solution ($\Phi_{tot,final} = 0$). Introducing a trade-off probability, P_w, to modify both the trajectory shape and the departure time ($P_w = 0.25, 0.5$ and 0.75) yields an interaction-free (or conflict-free in this particular case) trajectory plan within approximately 30 minutes of computation time. The number of modified trajectories in 3D space and in the temporal space depends on the chosen value for P_w.

The impact of the number of allowed virtual waypoints, M, on the computation time is also investigated. We conducted experiments with $M = 3$ virtual waypoints, using the value P_w = 0.5. The simulations were performed 10 times. Interaction-free solutions were found within an average of 43.17 minutes. Despite the increase of the richness of the solution space, using more virtual waypoints induces a higher combinatorics for the search space, thereby yielding larger computation time. Moreover, the resulting trajectories involve undesirable zig-zags.

Then, the impact of the particular local-search strategies (PT or IT) on the convergence of the hybrid-metaheuristic algorithm is investigated. The simulation was performed 10 times with probability P_w =0.5. The average computation time using a single intensification method (PT or IT), and using both methods sequentially (PT + IT) are shown in Table II. The evolutions of the value of objective function Φ_{tot} at the end of each temperature step are compared in Figure 18.

All local search methods (PT, IT, PT + IT) yield interaction-free solutions ($\Phi_{tot,final}$ =0) within the given maximum number of iterations ($n_T \times 688$). Numerical results show that intensifying the search of solutions on

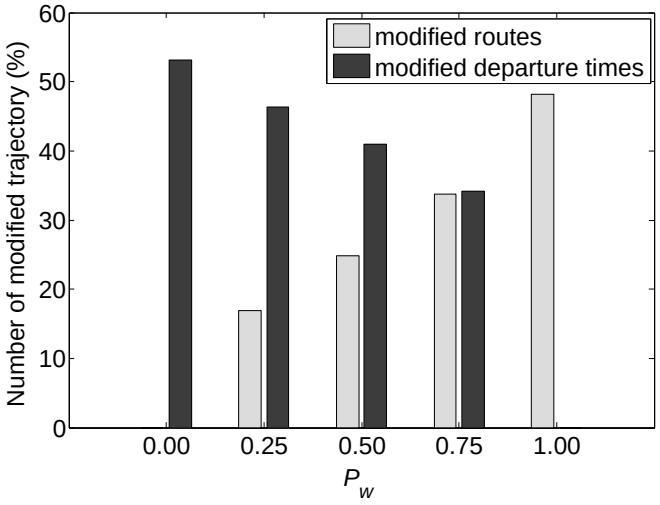

Figure 17: Impact of the chosen value of probability P_w on the number of modified routes and on the number of departure-time shifts.

Table II: Comparing the different intensification methods (all yielding to zero interaction).

Intensification method	Average computation time (minutes)
PT	55.15
IT	52.93
PT + IT	29.47

each individual trajectory followed by intensifying the search on its interacting neighbors (PT + IT) converges significantly faster than performing each intensification method alone.

B. Traffic with TMA

Since it is not possible to modify the shape of the trajectories in the TMA, the TMA-traffic is much more constrained than the en-route traffic. Indeed, the interactions occurring in en-route airspace can be separated in space and in time, while the interactions occurring in the TMA can only be separated by acting in the time domain. In order to take into account the trajectory segments that belong to the TMA, we first set the size of the minimum separation $N_{h_{TMA}}$ to 3 NM. The trajectory set consists of $N = 30,695$ trajectories which yields $\Phi_{tot,initial}$ = 235,632 initial interactions.

The input parameters of the optimization algorithm are, here again, empirically set. Two different values for the maximum departure time shift ($-\delta_a^i$, δ_d^i) and for the maximum route length extension (d_i) are used, and their values are given in Table III. The simulation is carried out 10 times for each of the three cases in Table III. Again, the proposed algorithm is able to find interaction-free solutions ($\Phi_{tot,final}$ = 0) for the given traffic situation involving the high-density traffic occurring in the TMA. The computation time and the number of modified trajectories relevant to each case of the optimization constraints are compared in Figure 19.

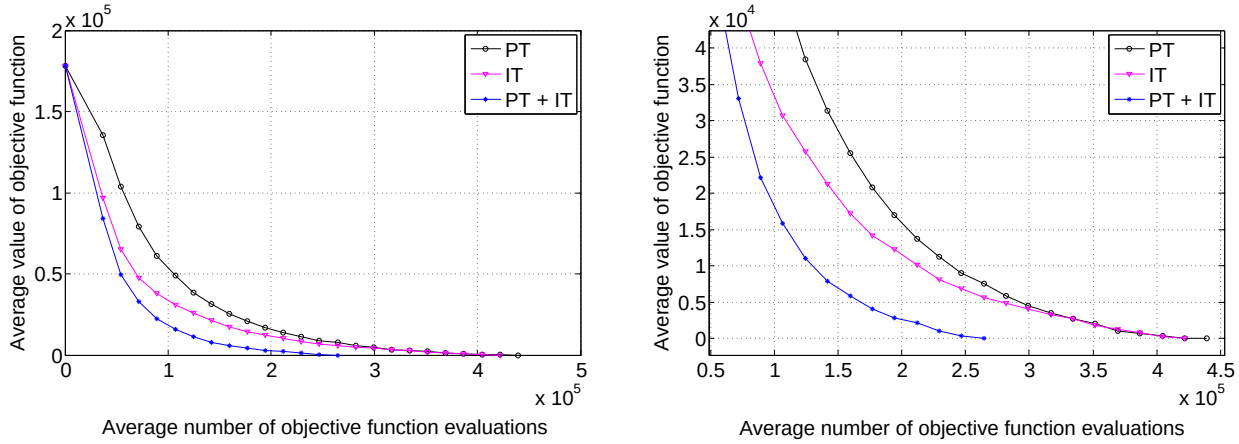

Figure 18: Evolution of the objective function value using the different local search strategies (right: close-up plot near the lowest objective function values).

Table III: Values of the maximum route length extension (d_i) and the maximum departure time shift ($-\delta_a^i$, δ_d^i) used for computational experiments on air traffic with TMA.

Case	d_i	$-\delta_a^i$, δ_d^i (minutes)
1	0.12	60
2	0.25	60
3	0.25	120

Remark that the initial interaction when taking into account the air-traffic in the TMA is significantly higher than when considering only en-route traffic. This is due to the high density of the traffic in the TMA. In addition, our interaction-detection method cannot distinguish aircraft using parallel runways from actual interaction. This leads to some false-positive contributions to the interactions. This larger problem instance is more difficult to solve, and requires longer computation time to converge to interaction-free solutions. Moreover, the algorithm has to modify more trajectories in order to solve all the interactions.

One observes that the computation time required to obtain the interaction-free solution depends on the size of the solution space. As expected, with the same setting d_i = 0.12 and maximum time shift, $-\delta_a^i = \delta_d^i = 60$ minutes (case 1), the algorithm requires significantly more computation time for solving the scenario with TMA traffic than to solve the scenario involving only en-route traffic. However, the required computation time decreases significantly when the solution space is relaxed (i.e., when more candidate solutions are considered; case 2 and case 3).

C. Taking into account uncertainty

In this subsection, we consider the uncertainty of aircraft position in the horizontal plane. Simulations were performed on both the en-route traffic scenario, and on the traffic scenario involving the TMA. The uncertainty margin in en-route is set to $R_{enroute}$ = 3 NM. The uncertainty margin in the TMA is not taken into account

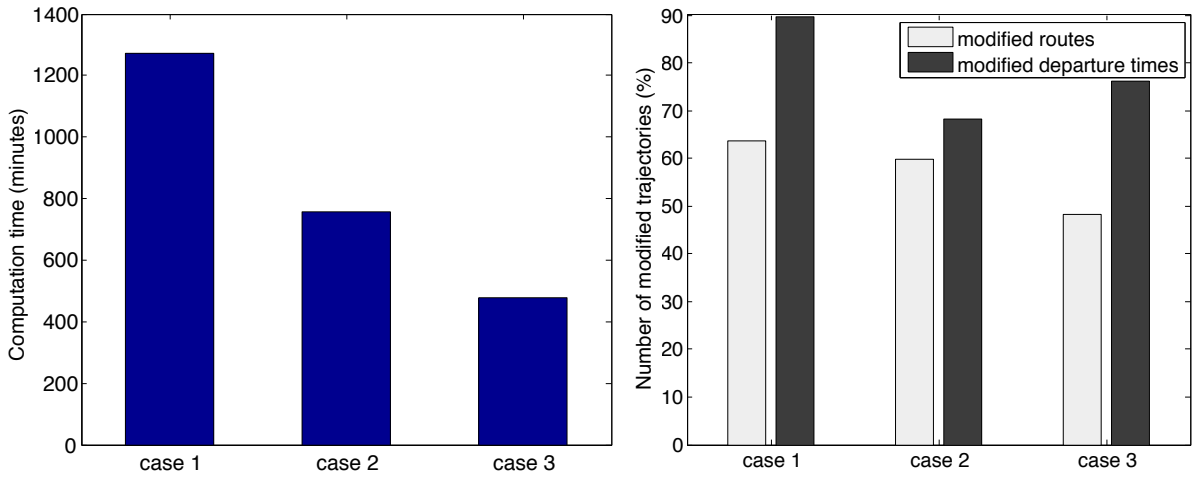

Figure 19: Left: Impact of the size of the solution space on the computation time. Right: Impact of the size of the solution space on the number of modified routes and the number of modified departure times.

($R_{TMA} = 0$), since during this phase of flight, aircraft are usually required to follow a given path with very high precision. The user-defined input parameters of the optimization algorithm are empirically set as follows. The maximum departure time shift is $-\delta_a^i$, δ_d^i = 120 minutes, and the maximum route length extension d_i is 0.25. The number of waypoints, M, and the coefficients a_i and b_i are set as in the previous subsection. The initial total interaction $\Phi_{tot,init}$ of both traffic scenarios with and without consideration of uncertainty are compared in Figure 20 (left). The simulation was carried out 10 times. The proposed algorithm is able to separate all trajectories ($\Phi_{tot,final}$ = 0) for both traffic scenarios, taking into account uncertainties of aircraft positions in the horizontal plane. The computation times to reach the interaction-free solutions for both traffic scenarios with and without consideration of uncertainty are compared in Figure 20 (right).

The required computation time is significantly longer when the uncertainty is considered. It is, however, still viable for a strategic planning phase. This can be improved by introducing more degrees of freedom to the solution space, e.g., alternative flight levels, or speed regulation in the TMA.

VII. Conclusions

We introduced an efficient methodology to address strategic planning of aircraft trajectories in the framework of future trajectory-based ATM operation involving large scale traffic such as that at the European-continent scale. The aim of the proposed method is to minimize interaction between trajectories, so as to minimize the air-traffic controller's workload. The proposed method relies on a route/departure-time allocation technique to modify the initial trajectory plan. The problem was modeled mathematically under the form of an optimization problem aiming at minimizing interaction between trajectories.

In order to measure the interaction between trajectories, we developed a grid-based interaction-detection method. To reduce the number of sampling points needed while minimizing further the computation time, this interaction-detection method interpolates the aircraft position between two suspected sampling points instead of refining the

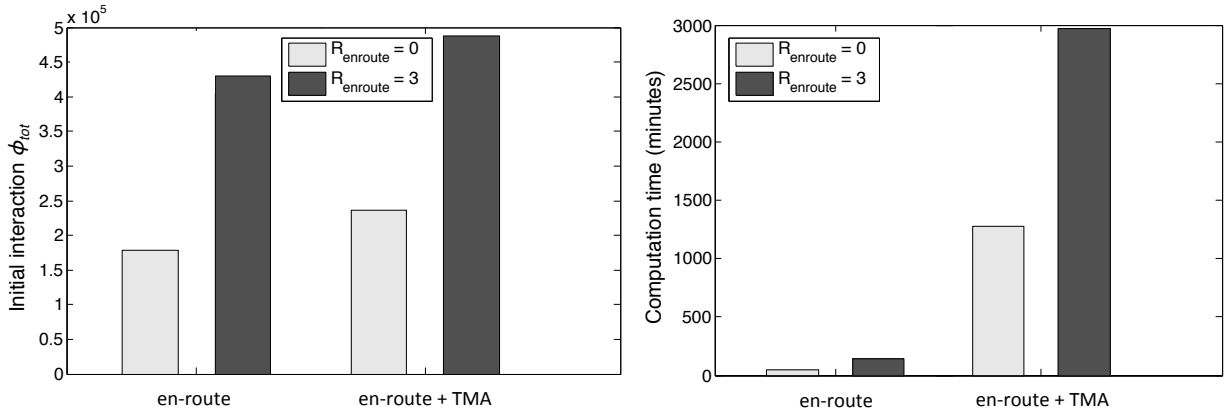

Figure 20: Left: Initial interaction between trajectories with and without consideration of uncertainty. Right: Comparison of computation time with and without consideration of uncertainty.

sampling-time step.

To find an optimal route and a departure time for each flight, we rely on a hybrid-metaheuristic optimization algorithm that combines the advantages of simulated annealing and of hill-climbing local search methods. The simulated annealing ensures diversity of the candidate solutions considered, while the local-search methods intensify the search in promising regions of the feasible domain in order to accelerate convergence.

The proposed algorithm was first tested with en-route air-traffic data over the European airspace. Two different local search strategies were investigated and compared. The first strategy concentrates on improving the current solution by modifying one single trajectory at a time. The second strategy aims at improving solutions by modifying all neighboring flights interacting with a given flight. Numerical results show that it is more efficient to employ both strategies sequentially to converge to interaction-free solutions, since it requires $\approx 40\%$ less computation time than using each strategy separately.

The impact of the number of virtual waypoints on the resolution time was studied. Despite the increasing of the richness of the solution space, using more virtual waypoints induces more combinatorics to the search space leading to a longer computation time. Moreover, the resulting trajectories involve undesirable zig-zags.

The impact of augmenting the number of degrees of freedom in the search space on the quality of the results obtained by our methodology was also studied and discussed. Numerical results show that using departure-time shifts alone is not sufficient to yield zero-interaction solutions. Similarly, modifying only the shape of trajectory alone is not sufficient either. In addition, the latter strategy requires prohibitive computation time. When the modifications of both the departure times and the shape of trajectory are allowed, the richness of the solution space increases. Therefore an optimal (interaction-free) solution can be obtained within significantly less computation time.

The proposed algorithm was then tested with the same large-scale European airspace air-traffic data, but this time taken also into account the air traffic occurring in the terminal control area (TMA). This later traffic scenario involves more number of aircraft trajectories and more initial interactions, which make it more difficult to solve all the interactions. Therefore, the algorithm requires longer computation time to separate all the trajectories. Again,

the proposed methodology proved its ability to find an interaction-free trajectory plan within computational time which is viable for strategic planning.

The effects of the optimization constraints (maximum departure time shifts and maximum route length extension) were also studied. As expected, when relaxing such constraints, the problem can be solved within less computation time.

Finally, uncertainties of aircraft position in the horizontal plane were taken into account. Instead of considering the aircraft position at a given time as a fixed 3D point, in this preliminary study, the aircraft position was modeled as a bounded circular area of radius R. Numerical results show that the proposed algorithm is again able to find an interaction-free trajectory plan for this more demanding optimization problem.

In further research, instead of being content with interaction-free solutions, we shall concentrate on reducing the cost associated to the modifications of the initial trajectory plan, thereby reducing the number of departure time shifts and/or the total trajectory length extension. In addition, we aim at increasing further the number of degree of freedom in the search space: we plan to allow, as new decision variables in our model, speed changes in the TMA, and to introduce alternative vertical profiles. Finally, to increase the robustness of the solutions obtained, we intend to extend the definition of interaction to the temporal space: in addition to minimizing interaction, the strategic trajectory plan will thereby also ensure separation of the trajectories in the temporal space during a sufficiently-long interval of time.

REFERENCES

[1] M. Nolan, *Fundamentals of Air Traffic Control*, 5th ed. Belmont, CA: Wadsworth, 2011.

[2] "SESAR Definition Phase - Deliverable D3, The ATM Target Concept," SESAR Consortium, Tech. Rep. DLM-0612-001-02-00a, 2007.

[3] 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.

[4] 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. [Online]. Available: http://www.4dcogc-project.org

[5] J. Lohn and J. Rios, "A comparison of optimization approaches for nationwide traffic flow management," in *AIAA Guidance, Navigation and Control Conference*, Chicago, 2009.

[6] A. Odoni, "The flow management problem in air traffic control," in *Odoni, A.R., Bianco, L., Szego, G. (eds.), Flow Control of Congested Networks*. Springer Berlin Heidelberg, 1987, ch. 17, pp. 269–288.

[7] M. Terrab and A. Odoni, "Strategic flow management for air traffic control," *Operations Research*, vol. 41, no. 1, pp. 138–152, 1993. [Online]. Available: http://www.jstor.org/stale/171949

[8] S. Cafieri and N. Durand, "Aircraft deconfliction with speed regulation: new models from mixed-integer optimization," *Journal of Global Optimization*, vol. 58, no. 4, pp. 613–629, 2014. [Online]. Available: http://link.springer.com/article/10.1007%2Fs10898-013-0070-1

[9] S. Constans, B. Fontaine, and R. Fondacci, "Minimizing potential conflict quantity with speed control," in *4th Eurocontrol Innovative Research Workshop And Exhibition*, Bretigny-sur-Orge, 2005, pp. 265–274.

[10] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.

[11] 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. [Online]. Available: http://dx.doi.org/10.1287/opre.46.3.406

[12] D. Gianazza and N. Durand, "Separating air traffic flows by allocating 3d-trajectories," in *23rd Digital Avionics Systems Conference.*, vol. 1, 2004, pp. 2.D.4–21–13.

[13] D. Bertsimas, G. Lulli, and A. Odoni, "The air traffic flow management problem: An integer optimization approach," in *Integer Programming and Combinatorial Optimization*, ser. Lecture Notes in Computer Science. Springer, 2008, vol. 5035, ch. 3, pp. 34–46. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-68891-4_3

[14] ——, "An integer optimization approach to large-scale air traffic flow management," *Operations Research*, vol. 59, no. 2, pp. 211–227, 2011.

[15] N. Barnier and C. Allignol, "Combining flight level allocation with ground holding to optimize 4D-deconfliction," in *9th USA/Europe Air Traffic Management Research and Development Seminar*. Berlin, Germany: ATM Seminar, 2011. [Online]. Available: http://pom.tls.cena.fr/papers/notice.html?todo=abstract&type=articles&file=atm2011_barnierallignol.txt

[16] A. Akgunduz, B. Jaumard, and G. Moeini, "Non-time indexed modeling for en-route flight planning with speed-fuel consumption trade-off," in *2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, Toulouse, 2013.

[17] S. Oussedik and D. Delahaye, "Reduction of air traffic congestion by genetic algorithms," in *PPSN V Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 1998, vol. 1498, ch. chapter 84, pp. 855–864. [Online]. Available: http://dx.doi.org/10.1007/BFb0056927

[18] S. Oussedik, "Application de l'evolution artificielle aux problèmes de congestion du traffic aérien," Ph.D. dissertation, École Polytechnique, Palaiseau, France, 2000.

[19] D. Delahaye and S. Puechmorel, *Modeling and Optimization of Air Traffic*. Hoboken, NJ: Wiley-ISTE, 2013.

[20] N. Barnier and C. Allignol, "4D - trajectory deconfliction through departure time adjustment," in 8^{th} *USA/Europe Air Traffic Management Research and Development Seminar (ATM 2009)*, Napa, 2009.

[21] N. Durand and J. Gotteland, "Genetic algorithms applied to air traffic management," in *Metaheuristics for Hard Optimization*. Berlin, Germany: Springer Berlin Heidelberg, 2006, ch. chapter 10, pp. 277–306. [Online]. Available: http://dx.doi.org/10.1007/3-540-30966-7_10

[22] N. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau, "A light-propagation model for aircraft trajectory planning," *Journal of Global Optimization*, vol. 56, no. 3, pp. 873–895, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10898-012-9896-1

[23] S. Chaimatanan, D. Delahaye, and M. Mongeau, "A methodology for strategic planning of aircraft trajectories using simulated annealing," in *1st International Conference on Interdisciplinary Science for Air traffic Management*, Daytona Beach, 2012. [Online]. Available: http://hal-enac.archives-ouvertes.fr/hal-00912772

[24] ——, "Strategic deconfliction of aircraft trajectories," in *2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, Toulouse, France, 2013. [Online]. Available: http://hal-enac.archives-ouvertes.fr/hal-00868450

[25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[26] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machine," *Journal of Chemical Physics*, vol. 21, pp. 1087–1091, 1953.

[27] J. Dréo, A. Petrowski, P. Siarry, and E. Taillard, *Metaheuristics for Hard Optimization Methods and Case Studies*. Berlin, Germany: Springer, 2006.

[28] E. Aarts and J. Lenstra, *Local search in combinatorial optimization*. New York: Wiley, 1997.

[29] G. Morani, V. D. Vito, F. Corraro, N. Grevtsov, and A. Dymchenko, "Automatic Guidance through 4D Waypoints with time and spatial margins," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, Boston, 2013.