HAL
archives-ouvertes.fr

Large Scale 4D Trajectory Planning

Arianit Islami, Supatcha Chaimatanan, Daniel Delahaye

▶ **To cite this version:**

Arianit Islami, Supatcha Chaimatanan, Daniel Delahaye. Large Scale 4D Trajectory Planning. Air Traffic Management and Systems – II , 420, Springer, pp 27-47, 2016, Lecture Notes in Electrical Engineering, 978-4-431-56423-2. <10.1007/978-4-431-56423-2_2>. <hal-01343591>

HAL Id: hal-01343591

https://hal-enac.archives-ouvertes.fr/hal-01343591

Submitted on 8 Jul 2016

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

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

Chapter 1
Large Scale 4D Trajectory Planning

Arianit Islami, Supatcha Chaimatanan, Daniel Delahaye

Abstract To sustain the continuously increasing air traffic demand, the future air traffic management system will rely on a so-called trajectory based operations concept that will increase air traffic capacity by reducing the controllers workload. This will be achieved by transferring tactical conflict detection and resolution tasks to the strategic planning phase. In this future air traffic management paradigm context, this paper presents a methodology to address such trajectory planning at nation-wide and continent scale. The proposed methodology aims at minimizing the global interaction between aircraft trajectories by allocating alternative departure times, alternative horizontal flight paths, and alternative flight levels to the trajectories involved in the interaction. To improve robustness of the strategic trajectory planning, uncertainty of aircraft position and aircraft arrival time to any given position on the trajectory are considered. This paper presents a mathematical formulation of this strategic trajectory planning problem leading to a mixed-integer optimization problem, whose objective function relies on the new concept of interaction between trajectories. A computationally efficient algorithm to compute interaction between trajectories for large-scale applications is pre-

Arianit Islami, Daniel Delahaye
Applied Mathematics Laboratory
French Civil Aviation University
7 Avenue Edourd Belin
31055, Toulouse, France.
e-mail: arianit.islami@caa-ks.org,
e-mail: delahaye@recherche.enac.fr

Supatcha Chaimatanan
Geo-informatics and space
development agency
120 The government complex,
Chaeng Wattana, Lak Si,
Bangkok 10210 Thailand
e-mail: supatcha@gistda.or.th*

sented and implemented. Resolution method based on hybrid-metaheuristic algorithm have been developed to solve the above large-scale optimization problems. Finally, the overall methodology is implemented and tested with real air traffic data taking into account uncertainty over the French and the European airspace, involving more than 30,000 trajectories. Conflict-free and robust 4D trajectory planning are produced within computational time acceptable for the operation context, which shows the viability of the approach.

1.1 INTRODUCTION

Air traffic regulations impose that aircraft must always be separated by some prescribed distance, noted N_v for the vertical separation and N_h for the horizontal separation. Aircraft are considered to be in *conflict* when these *minimum separation* requirements are violated. As the global air traffic demand keeps on increasing, congestion problem becomes more and more critical.One of the key solutions is to balance the air traffic demand and the overall capacity of the Air Traffic Management (ATM) system. In order to cope with the increasing demand, the future ATM system will rely on the trajectory based operations concept. In this concept, aircraft will be required to follow a negotiated *conflict-free* trajectory, accurately defined in 4 dimensions (3 spatial dimensions and time) in order to reduce the need of controller's intervention during the tactical phase. In this perspective, the key factor to improve the ATM capacity is an efficient strategic 4D trajectory planning methodology to compute a *conflict-free 4D trajectory* for each aircraft.

In this work, we propose a methodology to address such a strategic planning of trajectories at national and continent scale. The goal of the proposed method is to separate a given set of aircraft trajectories in both the three dimensional space and in the time domain by allocating an alternative flight plan (route, departure time, and flight level) to each flight.

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

In real-life situations, aircraft may not be able to follow precisely the assigned 4D trajectory due to external events, such as passenger delays, wind conditions, etc. Besides, aircraft may not be able to fly at their optimal speed profile in order to satisfy the hard constraints imposed on the 4D trajectory. To improve robustness of the deconflicted trajectories and to relax the 4D trajectory constraints, uncertainties of aircraft position and arrival time will

also be taken into account in the strategic trajectory planning process presented in this paper.

The following section of this paper is organized as follows. Section 1.2 reviews previous related works on aircraft trajectory deconfliction. Section 1.3 describes uncertainty model, explains the concept of interaction between trajectories, and presents the trajectory planning problem in mathematical framework. Section 1.4 proposes an efficient method for detecting interactions between aircraft trajectories in a large-scale context. Section 1.5 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 and discussed in Section 1.6.

1.2 PREVIOUS RELATED WORKS

During recent years, there are many research works in the literature that address the trajectory deconfliction problem considering large-scale air traffic. Aircraft trajectory deconfliction problem that relies on genetic algorithm to solve en-route conflicts between trajectories, taking into account uncertainties of aircraft velocity, is considered in [12]. The authors propose two conflict-resolution maneuvers: modifying the heading, and modifying the flight level. The solutions are provided by GA. It is able to solve all conflicts involving 7,540 flights considering different levels of uncertainties within reasonable computation time.

In [2], the authors consider a 4D trajectory deconfliction problem using a ground holding method. Potential conflicts between trajectories are detected by pairwise comparison. However, in presence of take-off time uncertainties, the proposed method must allocate significant delays in order to solve all the conflicts. To increase the degrees of freedom, the same authors introduce an option to allocate alternative flight levels in [3]. The results show advantages of using the flight level allocation technique in terms of reduced delay, in presence of departure time uncertainties. In [1], a flight-level allocation technique is used to address 4D trajectory deconfliction at the European continent scale. However, the proposed method yields residual potential conflicts.

Another idea to separate trajectories is based on speed regulations; it is used, for instance in [8]. In these works, conflict detection and resolution are performed at two layers with different sampling periods and time windows. Speed regulations introduce additional degree of freedom to the trajectory design. However, it requires numerous extensive and fine-tuned computations, which are not suitable to implement in a large-scale problem.

In [9, 10], a Light Propagation Algorithm (LPA) is introduced to solve potential conflicts between 4D trajectories, to avoid congested, and bad-weather areas. The optimal trajectory that solve conflicts are provided by a Branch-and-Bound (B&B) algorithm.

In [10], to improve robustness of aircraft trajectories, uncertainty is modeled as a time segment. The uncertainty increases the difficulty of the problem and reduces the solution space, so that the LPA can remove only 88% of the conflicts. The remaining conflicts are solved by imposing time constraints called Required Time of Arrival (RTA).

A methodology to optimize and deconflict aircraft trajectories in the horizontal plane, in en-route environment, and in real time is proposed by the author of [13]. In this work, aircraft trajectories are deconflicted and optimized in the time scale of thirty minutes into the future by sequentially computing optimal-wind and conflict-free trajectories for each aircraft, considering previously-planned trajectories as obstacles.

However, none of the proposed methodologies is able to solve globally the trajectory deconfliction problem due to its size and complexity. Most of the algorithms proposed in the literature rely on the moving time window strategy to reduce the size of the problem. This strategy is effective for conflict detection and resolution in tactical phases. However, when high-density traffic is involved, it tends to fail to solve all conflicts.

In this work, we put forward the works presented in [5, 6, 4, 7]. The proposed 4D trajectory planning methodology aims at solving conflict between all involving trajectories simultaneously at strategic level. In these works, optimal 4D trajectories for individual flights were allocated by solving a combinatorial optimization problem using a non-population based hybrid-metaheuristic optimization method.

1.3 MATHEMATICAL MODELING

This section presents the mathematical model used to describe our strategic trajectory planning methodology. First, uncertainty of aircraft positions and arrival times based on two different models are characterized. Then, a definition of *interaction* between trajectories is given. Next, the route / departure-time / flight level allocation techniques adapted for strategic trajectory-planning are described. Finally, a mathematical formulation of the interaction minimization problem is presented.

1.3.1 Uncertainties

Consider a given set of N trajectories, where each trajectory, i, is defined by a time sequence of 4D coordinates, $P_{i,k}(x_{i,k}, y_{i,k}, z_{i,k}, t_{i,k})$, for $k = 1, \ldots, K_i$, where K_i is the total number of *sampling points* of trajectory i, for $i = 1, \ldots, N$. Each trajectory is sampled with a (given) constant sampling time,

Δt. These coordinates specify that aircraft i must arrive at a given point $(x_{i,k}, y_{i,k}, z_{i,k})$ at time $t_{i,k}$.

However, in reality, aircraft are subjected to unpredicted external events, which cause uncertainties on aircraft position and arrival time with respect to their planned 4D trajectory. In order to consider such uncertainties, we rely on the concept of robust optimization, using two different models of the uncertainty sets.

1.3.1.1 Deterministic model

For simplicity, we define x_P, y_P, z_P, t_P as the 4D coordinate of any given point $P_{i,k}$. Consider an initial 4D trajectory planning specifying that an aircraft must arrive at a given horizontal coordinate (x_P, y_P) at time t_P. Due to uncertainties, we shall assume that the *real* horizontal position, (x_P^r, y_P^r), of the aircraft at time t_P can be in an area defined by a disk of radius R_h (defined by the user) around (x_P, y_P), as illustrated in Fig. 1.1. In other words, the possible locations of the aircraft at time t_P are the elements of the set: $\{(x_P^r, y_P^r) : (x_P^r - x_P)^2 + (y_P^r - y_P)^2 \leq R_h^2\}$.

To ensure horizontal separation of aircraft subjected to such uncertainties, the protection volume has to be enlarged by a radius of R_h as illustrated in Fig. 1.1. Thus, the *robust minimum separation in the horizontal plane*, N_h^r, is defined as: $N_h^r := N_h + R_h$, where N_h is the minimum horizontal separation of the case without uncertainty.

In the vertical dimension, we shall assume that during such a non-level flight phase, the real altitude, denoted z_P^r, of the aircraft at a given time t_P lies in a bounded interval defined by an uncertainty radius R_v (set by the user) which reduces strongly when the aircraft reaches its requested flight level. In other words, the possible altitudes of the aircraft during non-level flight phase at time t_P are the elements of the set: $\{z_P^r : z_P - R_v \leq z_P^r \leq z_P + R_v\}$.

To ensure vertical separation of aircraft subjected to such uncertainties, the vertical separation requirement has to be enlarged by R_v as illustrated in Fig. 1.1. Thus, the *robust minimum separation in the vertical dimension*, noted N_v^r, is defined as: $N_v^r := N_v + R_v$, where N_v is the minimum vertical separation of the case without uncertainty.

In addition to the uncertainty in the 3D space domain (see Fig. 1.1), aircraft may be subject to uncertainty so that it arrives at a given position with a time error. Let t_ϵ be the *maximum time error* (defined by the user). For simplicity, to implement the interaction detection scheme, we shall assume that t_ϵ is chosen so that it is a multiple of the discretization time step Δt. The real arrival time, noted t_P^r, of aircraft at the same trajectory point therefore lies in the time interval: $[t_P - t_\epsilon, t_P + t_\epsilon]$, where t_P is the assigned arrival time to point $P_{i,k}$.

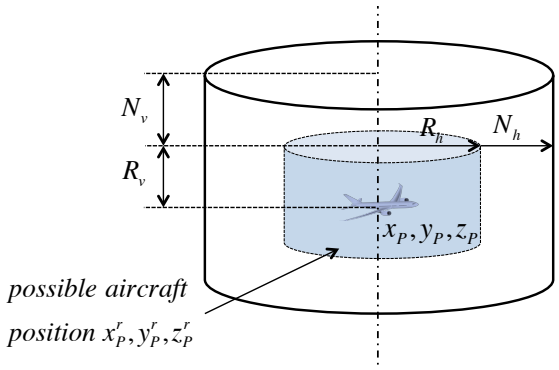

Fig. 1.1 Possible aircraft position in the 3D space domain in presence of deterministic uncertainty.

1.3.1.2 Probabilistic model

The worst-case-oriented uncertainty model presented above considers that every possible cases in the given uncertainty set are equally likely. However, some events corresponding to the points in the uncertainty set have very low probability to occur. Trying to immune the solution against such events could yield unnecessarily costly solutions, and can be interpreted as too conservative for a situation involving high levels of uncertainty as it is the case in strategic planning.

As an aircraft is able to follow a given flight profile with very high accuracy thanks to the flight management system (FMS).We shall consider that the residual uncertainty of aircraft position is more likely to occur in the time domain.

Using the maximum time error, t_ϵ (set by the user), the *predicted* arrival time of an aircraft at a position $P_{i,k}$ under uncertainty lies in the interval: $[t_P - t_\epsilon, t_P + t_\epsilon]$, where t_P is the assigned arrival time to point $P_{i,k}$.

For the purpose of interaction computation, which will be explained in the following subsection, we assume here that the predicted aircraft arrival time can be modeled as a random variable with the following triangular distribution defined over the interval $[t_P - t_\epsilon, t_P + t_\epsilon]$. Given the lower limit $t_P - t_\epsilon$, the upper limit $t_P + t_\epsilon$, the *predicted* arrival time, to the position $P_{i,k}$ is given by the probability density function:

$$\mathcal{T}_{P_{i,k},t_\epsilon}(t) = \begin{cases} 0 & \text{for } t < t_P - t_\epsilon, \\ \frac{(t - t_P + t_\epsilon)}{t_\epsilon^2} & \text{for } t_P - t_\epsilon \leq t \leq t_P, \\ \frac{(-t_P + t_\epsilon - t)}{t_\epsilon^2} & \text{for } t_P < t \leq t_P + t_\epsilon, \\ 0 & \text{for } t_P + t_\epsilon < t. \end{cases} \tag{1.1}$$

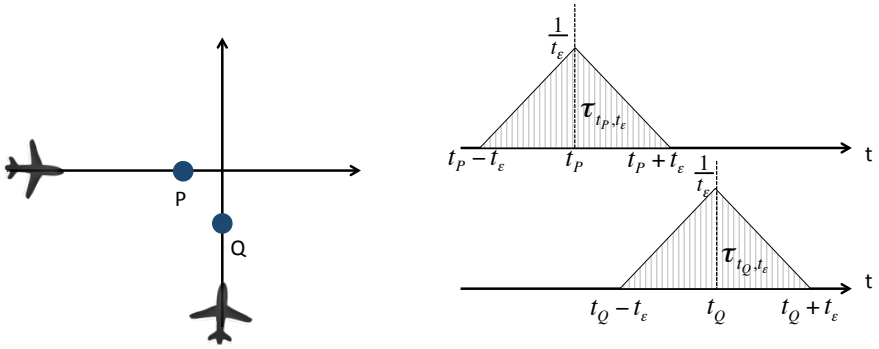

Fig. 1.2 Uncertainty of aircraft arrival time, defined by triangular distribution over given time intervals (left: view in the space domain; right: view in the time domain).

where $\mathcal{T}_{P_{i,k},t_\epsilon}(t)$ denotes the triangular distribution. Fig. 1.2 illustrates the uncertainty of arrival time of two aircraft A and B to the trajectory sample points P and Q respectively. The time uncertainties are defined by a triangular distribution function over the time interval $[t_P - t_\epsilon, t_P + t_\epsilon]$ and $[t_Q - t_\epsilon, t_Q + t_\epsilon]$ respectively.

1.3.2 Interaction between trajectories

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

To explain the process to determine the interaction between aircraft trajectories, let us first consider two trajectories A and B, and let P and Q be any pair of sample points on the trajectories A and B respectively. To consider the above-mentioned deterministic uncertainty models, we must check whether the minimum separations, N_h^r and N_v^r is satisfied, between every possible pair of points such as P and Q (pair-wise comparisons). A potential conflict between trajectories A and B, taking into account uncertainties, can occur when the three following conditions are satisfied for a certain pair of sample points, P and Q, from each trajectory:

- $d_h := \sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2} < N_h^r$.
- $d_v := |z_P - z_Q| < N_v^r$.

- $[t_P - t_\epsilon, t_P + t_\epsilon] \cap [t_Q - t_\epsilon, t_Q + t_\epsilon] \neq \emptyset$,
  i.e. $|t_P - t_Q| \leq 2t_\epsilon$.

When the above conditions are satisfied, we say that *point P is in conflict with point Q taking into account the deterministic-type uncertainty.*

Let us define further

$$\mathcal{C}^D(P,Q) = \begin{cases} 1 & \text{if point } P \text{ is in conflict with point } Q \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

With the above definitions, the *interactions at point $P_{i,k}$*, denoted $\Phi^D_{i,k}$, may be defined as the total number of times the protection volume around point $P_{i,k}$ taking into account the deterministic-type uncertainty is violated. Therefore, $\Phi^D_{i,k}$ is given by

$$\Phi^D_{i,k} = \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{l=1}^{K_j} \mathcal{C}^D(P_{i,k}, P_{j,l}). \tag{1.3}$$

where K_i is the number of sampled points of trajectory i.

Finally, the *robust total interaction between trajectories*, that we are minimizing, is:

$$\Phi^D_{tot} = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \Phi^D_{i,k}, \tag{1.4}$$

where N is the total number of trajectories.

To explain the process to compute the total robust interaction between trajectories based on probabilistic-type uncertainty, let us consider the trajectories A and B given in Fig. 1.2. Let P and Q be any trajectory sample points on trajectories A and B respectively. The predicted arrival time of aircraft A to the given point P, and the predicted arrival time of aircraft B to the given point Q are given by $\mathcal{T}_{t_P,t_\epsilon}(t)$ and $\mathcal{T}_{t_Q,t_\epsilon}(t)$ respectively.

Again, a potential conflict between trajectories A and B occurs when there exists a pair of points, P and Q, from each trajectory such that the three following conditions are satisfied:

- $d_h < N^r_h$;
- $d_v < N^r_v$;
- and $[t_P - t_\epsilon, t_P + t_\epsilon] \cap [t_Q - t_\epsilon, t_Q + t_\epsilon] \neq \emptyset$.

The probabilistic interaction, denoted $\mathcal{P}_{t_\epsilon}(P,Q)$, associated to the trajectory sample points P and Q is formally defined as follows:

$$\mathcal{P}_{t_\epsilon}(P,Q) := \int_{I_{PQt_\epsilon}} \mathcal{T}_{t_P,t_\epsilon}(t)\mathcal{T}_{t_Q,t_\epsilon}(t)dt, \tag{1.5}$$

where I_{PQt_ϵ} denotes the time interval $[t_P - t_\epsilon, t_P + t_\epsilon] \cap [t_Q - t_\epsilon, t_Q + t_\epsilon]$. Remark that when this intersection is the empty set, the integral in (1.5) reduced to zero.

With the above definition, we define a *robust interaction at a point* $P_{i,k}$ based on the probabilistic-type uncertainty, denoted $\Phi^P_{i,k}$, to be the sum of all the probabilistic interaction associated to point P.

Hence, we have

$$\Phi^P_{i,k} := \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{l=1}^{K_j} \mathcal{P}_{t_\epsilon}(P_{i,k}, P_{j,l}), \tag{1.6}$$

where K_j are the number of sampling points for trajectory j, and where $\mathcal{P}_{t_\epsilon}(P_A, P_B)$ is the probabilistic interaction associated to the sample points P_A and P_B of trajectory A and B respectively.

Therefore, the *total interaction between trajectories*, based on probabilistic-type uncertainty, denoted Φ^P_{tot}, for a whole N-aircraft traffic situation is simply defined as:

$$\Phi^P_{tot} = \sum_{i=1}^{N} \Phi^P_i = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \Phi^P_{i,k}. \tag{1.7}$$

1.3.3 Rout/Departure-Time/Flight level Allocation

The objective of this work is to allocate an alternative trajectory, an alternative departure time, and alternative flight level for each aircraft in order to minimize the total interaction between trajectories, taking into account uncertainty of aircraft position and time.

Given data. A problem instance is given by:

- A set of initial N discretized 4D trajectories;
- The discretization time step, Δt;
- The number of allowed virtual waypoints, M;
- The maximum allowed advance departure time shift of each flight i, $\delta^i_a < 0$;
- The departure time shift step size, δ_s;
- The maximum allowed delay departure time shift of each flight i, $\delta^i_d > 0$;
- The maximum allowed flight level shift of each flight i, $l_{i,max}$;
- The maximum allowed route length extension coefficient of each flight i, $0 \leq d_i \leq 1$;
- The length of the initial en-route segment of each flight i, $L_{i,0}$.

The alternative departure time, the alternative route, and the alternative flight level 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

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

$$\begin{aligned}\Delta_i :=\{&-N_a^i.\delta_s, -(N_a^i-1).\delta_s, \ldots,\\ &-\delta_s, 0, \delta_s, \ldots, (N_d^i-1).\delta_s, N_d^i.\delta_s\}.\end{aligned} \tag{1.8}$$

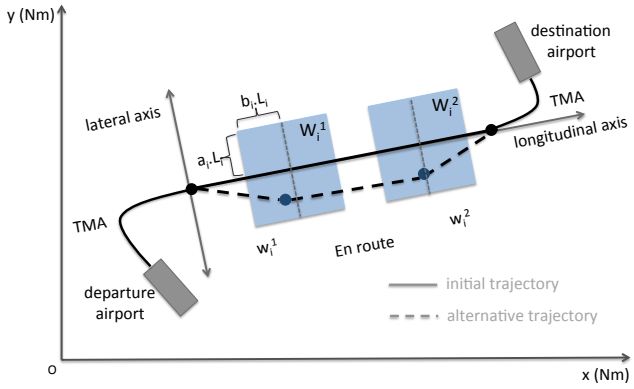

Fig. 1.3 Initial and alternative trajectories with rectangular-shape possible location of $M = 2$ virtual waypoints.

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

$$w_i = \{w_i^m | w_i^m = (w_{ix'}^m, w_{iy'}^m)\}_{m=1}^M, \tag{1.9}$$

near the initial en-route segment and then by reconnecting the successive waypoints with straight-line segments as illustrated in Fig. 1.3. To limit the route length extension, the alternative en-route profile of flight i must satisfy:

$$L_i(w_i) \leq (1 + d_i), \tag{1.10}$$

where $L_i(w_i)$ is the length of the alternative en-route profile determined by w_i. Fig. 1.3 illustrated initial and alternative trajectories, constructed with $M = 2$ waypoints, where the location of each waypoint is constrained to be in a rectangular-shape possible location. Let $W_{ix'}^m$ be a set of all possible normalized longitudinal locations of the m^{th} virtual waypoint on trajectory

i. For each trajectory i, the normalized longitudinal component, $w^m_{ix'}$, is set to lie in the interval:

$$W^m_{ix'} := \left[\left(\frac{m}{1+M} - b_i\right), \left(\frac{m}{1+M} + b_i\right)\right], \tag{1.11}$$

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

$$\left(\frac{m}{1+M} + b_i\right) < \left(\frac{m+1}{1+M} - b_i\right) \tag{1.12}$$

and hence the user should choose b_i so that

$$b_i < \frac{1}{2(M+1)}. \tag{1.13}$$

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

$$W^m_{iy'} := [-a_i, a_i], \tag{1.14}$$

where $0 \leq a_i \leq 1$ is a (user-defined) model parameter that defines the range of possible normalize lateral location of the m^{th} virtual waypoint on trajectory i, chosen a priori so as to satisfy (1.10). More detail about the method to modify the trajectory is presented in [7].

Alternative flight level. Another variable to modify the trajectory of each flight i is a flight-level shift $l_i \in \mathbb{Z}$. Therefore, the flight level, FL_i, of flight i is given by: $FL_i = FL_{i,0} + l_i$, where $FL_{i,0}$ is the initially-planned flight level of flight i. Fig. 1.4 shows a trajectory with two alternative flight levels. In order to limit the change of flight levels,the set, ΔFL_i, of all possible flight-level shifts for flight i is set to:

$$\Delta FL_i := [FL_{i,0} - l_{i,max}, \ldots, 0, \ldots, FL_{i,0} + l_{i,max}], \tag{1.15}$$

where $l_{i,max}$ is the (user-provided) maximum flight level shifts allowed to be allocated to flight i.

Let us set the compact vector notation: $\boldsymbol{\delta} := (\delta_1, \delta_2, \ldots, \delta_N)$, $\mathbf{w} := (w_1, w_2, \ldots, w_N)$, and $\mathbf{l} := (l_1, l_2, \ldots, l_N)$. We shall denote by u_i the components of u. It is a vector whose components are related to the modification of the i^{th} trajectory, thereby our decision variable is:

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

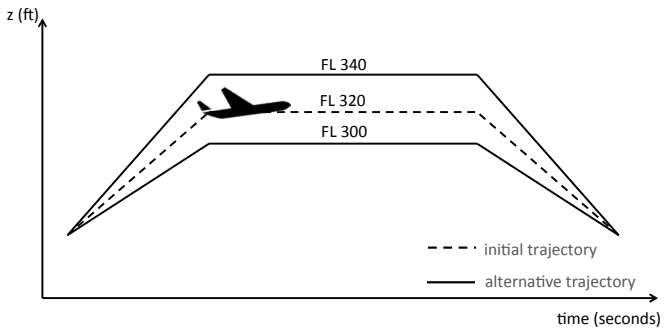

Fig. 1.4 Two alternative vertical profiles for a trajectory (two alternative flight levels).

The strategic trajectory planning problem under uncertainty can be represented by an interaction minimization problem formulated as a mixed-integer optimization problem as follows:

$$\begin{aligned} &\min_{u} \Phi_{tot}(u) \\ &\text{subject to} \\ &\quad \delta_i \in \Delta_i, \quad i = 1, 2, \ldots, N \\ &\quad l_i \in \Delta FL_i, \quad i = 1, 2, \ldots, N \\ &\quad w_i^m \in W_{ix'}^m \times W_{iy'}^m, \quad m = 1, 2, \ldots, M, \\ &\qquad\qquad\qquad\qquad\qquad\quad i = 1, 2, \ldots, N, \end{aligned} \tag{1.16}$$

where $\Phi_{tot}(u)$ is defined by (1.4) or (1.7) according to uncertainty model under consideration, and Δ_i, $W_{ix'}^m$, and $W_{iy'}^m$, ΔFL_i are defined by (1.8), (1.11), (1.14), and (1.15) respectively.

1.4 INTERACTION DETECTION

In order to evaluate the objective function, we rely on a grid-based interaction detection scheme which is implemented in a so-called *hash table* as presented in [4, 6, 7].

First, the airspace is discretized using a 4D grid (3D space + time), as illustrated in Fig. 1.5. The size of each cell in the x, y, z and t direction is defined by the minimum separation requirement, N_h^r, N_v^r and the discretization time step, Δt. To detect conflicts, the idea is to store the N trajectories in each corresponding cell in the 4D grid. Then, for each trajectory i, and for each cell (I_x, I_y, I_z, I_t) corresponding to each sampling point $P_{i,k} := (x_P, y_P, z_P, t_P)$,

we simply need to check all the surrounding (adjacent) cells in the x, y, and z directions corresponding to the time period $[t_p - 2t\epsilon, t_P + 2t\epsilon]$. If one of these surrounding cells is occupied by another aircraft, for instance j, we then note $j \in (I_x, I_y, I_z, I_t)$, and then the horizontal distance, d_h, and the vertical distance, d_v, between point $P_{i,k}$ and the sample point corresponding to aircraft j are computed.

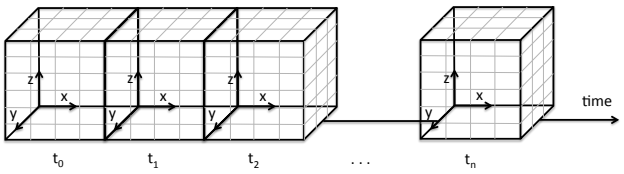

Fig. 1.5 Four-dimension (3D space - time) grid for conflict detection.

A violation of protection the volume is identified when both $d_h < N_h^r$ and $d_v < N_v^r$. When a violation of protection volume is identified, the interaction is computed using (1.3) or (1.6) depending on the type of uncertainty model considered. 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, one can simply choose a sufficiently small value of Δt. However, using small sampling-time step leads to large computation time and memory. Instead, we propose an inner-loop algorithm, 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 have been checked. More details of this interaction detection algorithm is presented in [7].

1.5 RESOLUTION ALGORITHM

To solve the strategic trajectory planning problem, we rely on a hybrid metaheuristics approach adapted to handle an air-traffic assignment problem at the continent scale. The proposed hybrid algorithm combines the Simulated Annealing (SA) and the Local Search (LS) 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.

1.5.1 Simulated annealing

Simulated annealing was separately introduced by S. Kirkpatrick et al. in 1982 [14] and by V. Černý in 1985 [17]. It is inspired by the annealing process in metallurgy where the state of a material can be modified by controlling the cooling temperature.

In the simulated annealing optimization algorithm, the objective function to be minimized is analogical to the energy of the physical problem, while the values of the decision variables of the problem are analogical to the coordinates of the material's particles. A control parameter, T, that decreases as the number of iterations grows, plays the role of the temperature schedule, and a number of iterations, N_I, at each temperature step plays the role of the time duration the material is kept at each temperature stage.

To simulate this evolution of the physical system towards a thermal equilibrium, the *Metropolis algorithm* [15] is used. For a given temperature, T, starting from a current configuration, the state space of the simulated system is subjected to a transformation (e.g. apply a local change to one decision variable). If this transformation improves the objective-function value, then it is accepted. Otherwise, it is accepted with a probability

$$P_{accept} := e^{\frac{\Delta E}{T}}, \tag{1.17}$$

where ΔE is the degradation of the objective-function value (negative for minimization). Repeating this process until the equilibrium is reached, the temperature is decreased according to a pre-defined cooling schedule. As the temperature decreases, the probability, P_{accept}, to accept a degrading solution becomes smaller and smaller. Therefore, the system will eventually converge to the nearest local optimum which will expectantly be close to a global optimum. We refer the reader interested by simulated annealing algorithm to the following books [11, 16].

For our problem, the simulated annealing proceeds as presented in [4]. In order to implement the simulated annealing algorithm to the strategic planning of 4D trajectories, we first define the following parameters.

1. **Neighborhood function.** To generate a neighborhood solution, first a flight i is randomly chosen. In order not to modify excessively the trajectories that are not involved in any interaction, we set a user-defined threshold value of interaction, denoted Φ_τ, such that the trajectory of a randomly chosen flight i will be modified only if

$$\Phi_i(u) \geq \Phi_\tau. \tag{1.18}$$

   Otherwise, another trajectory will be randomly chosen until condition (1.18) is satisfied. This process ensures that changes will be first applied on trajectories involved in congestion area.

Then, for a chosen flight, i, we introduce a user-defined parameter, $P_w \leq 1$ to control the probability of modifying the value of the i^{th} trajectory waypoint location vector, w_i, and a user-defined parameter $P_l \leq 1$ to control the probability of modifying the value of the flight level shift ΔFL_i. The probability to modify rather the departure time is therefore $1-P_w-P_l$. These parameters, P_w and P_l, allows the user to set his/her preference on the way to deconflict trajectories. The neighborhood function we use in this paper is summarized in Algorithm 1.

Algorithm 1 Neighborhood function

Require: probabilities P_w, P_l, trajectory i.

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

2. **Initial temperature and initial acceptance probabilities.** To determine the initial temperature and initial acceptance probability, we rely on a practical recommendations given in [11]. They are computed by first generating 100 deteriorating transformations (neighborhood solutions) at random; then by evaluating the average variations, ΔE_{avg}, of the objective-function value. The initial temperature, T_0, is then deduced from the relation:
$$\tau_0 = e^{\frac{\Delta E_{avg}}{T_0}},$$
where τ_0 is the initial rate of accepting degrading solutions that will be empirically set.
3. **Cooling schedule.** The cooling schedule plays an essential role to guide the system towards a good optimum. If the temperature is decreased slowly, the system is more likely to converge to a better solution, but it will require more computation time. On the other hand, decreasing too rapidly the temperature tends to yield undesirable local optima. For simplicity, we will decrease the temperature, T, following the geometrical law, therefore
$$T_i = \beta.T_{i-1},$$
where $0 \leq \beta \leq 1$ where the constant β will be experimentally tuned.

4. **Equilibrium state.** In order to reach an equilibrium, a sufficient number of iterations, denoted N_I, or moves, have to be performed at each temperature step. For simplicity, the value of N_I will be defined as constant, and will be experimentally set.
5. **Termination criterion.** Theoretically, it is suggested that the SA algorithm stops when the temperature reaches zero. However, this stopping criterion is not utilized in practice, since when the temperature is near zero, the probability of acceptance becomes negligible. In our case, the simulated annealing algorithm will terminate when the final temperature, T_f, reaches the value $C.T_0$, where $0 \leq C \leq 1$ is a user-defined coefficient.

1.5.2 Hill-climbing local search

Hill-climbing is a local search algorithm that only moves to a new solution only if it yields a decrease of the objective function. The process repeats until no further improvement can be found or until the maximum number of iterations $n_{T_{LOC}}$ is reached. In this work, we rely on two local-search modules that correspond to the two following strategies:

- Intensification 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.
- Intensification 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.

1.5.3 Hybrid algorithm

In order to improve efficiency of the optimization algorithm in terms of computation time, we propose to combine the SA and the hill climbing local search. The algorithms are combined in a self-contained manner, such that each algorithm is executed sequentially. The order of execution is controlled by pre-defined parameters, that controls the probabilities to carry out each method. The probability to carry out simulated-annealing step, P_{SA}, is:

$$P_{SA}(T) = P_{SA,min} + (P_{SA,max} - P_{SA,min}) \cdot \frac{T_0 - T}{T_0}, \tag{1.19}$$

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:

$$P_{Loc}(T) = P_{Loc,min} + (P_{Loc,max} - P_{Loc,min}) \cdot \frac{T_0 - T}{T_0}, \tag{1.20}$$

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:

$$P_{SL}(T) = 1 - (P_{SA}(T) + P_{Loc}(T)). \tag{1.21}$$

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.

1.6 NUMERICAL RESULTS

The proposed hybrid SA / LS algorithm is implemented in Java and run on an AMD Opteron 2 GHz processor with 128 Gb RAM. It is tested with two different uncertainty models, using real air traffic data at nation-wide and continent scale.

1.6.1 Deterministic uncertainty model

First, the proposed algorithm is tested on national-size and continent-size air traffic, considering deterministic uncertainties model.

National-size air traffic data First, we test the proposed methodology on the full-day national-size en-route air traffic over the French airspace involving 8,836 trajectories. Simulations are performed with different values for the parameters R_h, R_v, and t_ϵ, defining the size of the uncertainty sets. The parameter values chosen to specify the optimization problem are given in Table 1.1. The parameter values that specify the resolution algorithm are given in Table 1.2. The initial and final total interaction between trajectories, the computation time, and the number of iterations performed to solve the problems considering different levels of uncertainty are reported in Table 1.3 (the vertical uncertainty radius, R_v, is used only when aircraft are climbing and descending).

The size of the uncertainty set affects the resolution time and the final total interaction between trajectories. When increasing the time uncertainty, the initial interaction increases significantly (cases 1, 3, 4 and 5), and the algorithm requires more computation time to converge. The algorithm reaches an interaction-free solution for the case 2. It solves up to 99.7% of the initial interactions in the remaining cases (1, 3, 4, and 5), within computation times

Table 1.1 Chosen (user-defined) parameter values specifying the robust optimization problem for the national-size air traffic.

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

Table 1.2 Empirically-set (user-defined) parameter values of the resolution methodology to solve the national-size air traffic.

Parameter	Value
Number of iterations at each temperature step, N_I	200
Initial rate of accepting degrading solutions, τ_0	0.3
Geometrical temperature reduction coefficient, β	0.99
Final temperature, T_f	$(1/500).T_0$
Inner-loop interpolation sampling time step, t_{interp}	5 seconds
Probability to modify horizontal flight profile, P_w	1/3
Probability to modify flight level, P_l	1/3
Threshold value, Φ_τ	0.5 Φ_{avg}

that are still compatible in a strategic planning context (the worst run, case 5, involving less than 38 hours of CPU time).

Continent-size traffic data Then, the hybrid algorithm is tested on an air traffic data, involving en-route air traffic over the European airspace. The data set is a full day of air-traffic over the European airspace on July 1, 2011. It consists of 30,695 trajectories simulated with optimal vertical profiles and with direct routes. The user-defined parameter values specifying the optimization problem are the same as those given in Table 1.1. The maximum allowed flight level shifts, $l_{i,max}$ is set to 0 due to the lack of data. The parameter values of the hybrid-metaheuristic algorithm are the same as those given in Table 1.2, with $N_I = 4,000$.

The initial and final total interaction between trajectories, and the computation time to solve the problem considering different levels of uncertainty are reported in Table 1.4. Although the trajectories can be separated only by modifying the horizontal flight profile and the departure time of each flight, the resolution algorithm finds an interaction-free solution, taking into account uncertainty of aircraft positions, for problem instance in case 2. When time

Table 1.3 Initial and final total interaction between trajectories for the national-size air traffic, considering different dimensions for the deterministic uncertainty set.

case	uncertainty set dimensions	initial Φ^D_{tot}	final Φ^D_{tot}	solved interactions	CPU time (minutes)	no. of iterations
1	$R_h = 0$ NM. $R_v = 0$ feet. $t_\epsilon = 180$ seconds.	2,282,436	5,934	99.7%	1,093.8	1,083,215
2	$R_h = 1$ NM. $R_v = 100$ feet. $t_\epsilon = 60$ seconds.	765,448	0	100.0%	101.1	97,400
3	$R_h = 1$ NM. $R_v = 100$ feet. $t_\epsilon = 120$ seconds.	1,425,384	4,314	99.7%	1,809.0	1,791,000
4	$R_h = 1$ NM. $R_v = 100$ feet. $t_\epsilon = 240$ seconds.	2,821,706	37,290	98.7 %	2,213.3	2,191,970
5	$R_h = 2$ NM. $R_v = 100$ feet. $t_\epsilon = 240$ seconds.	5,000,430	110,021	97.9%	2,289.8	2,266,956

Table 1.4 Initial and final total interaction between trajectories for the continent-scale air traffic with different dimensions for the deterministic uncertainty set.

case	uncertainty set dimensions	initial Φ^D_{tot}	final Φ^D_{tot}	solved interactions	CPU time (minutes)	no. of iterations
6	$R_h = 3$ NM. $R_v = 200$ feet. $t_\epsilon = 60$ s.	5,142,632	634,474	87.7 %	2,756.2	2,728,776
7	$R_h = 3$ NM. $R_v = 200$ feet. $t_\epsilon = 0$ s.	430,234	0	100.0 %	347.6	345,528

uncertainty is considered (case 1), there remains less than 15% of the initial interaction between trajectories.

1.6.2 Probabilistic uncertainty model

Then, the proposed robust strategic 4D trajectory planning methodology is tested based on the probabilistic type uncertainty model. The parame-

Table 1.5 Numerical results for the national-size air traffic considering four different levels of aircraft maximum time uncertainty (1 to 4 minutes) based on probabilistic uncertainty model.

t_ϵ (seconds)	initial Φ^P_{tot}	final Φ^P_{tot}	solved interactions	CPU time (minutes)	No. of iterations
60	217,441.37	0.0	100.0 %	116.07	114,970
90	274,953.55	0.0	100.0 %	175.4	173,736
120	383,967.60	915.04	99.8 %	586.3	1,031,730
240	718,374.42	1,547.13	99.8 %	1,052.4	1,041984

ters of the hybrid simulated-annealing / local-search are the same as those presented in Table 1.2. Again, the proposed algorithm is tested with the national-size air traffic over the French airspace. Assuming that aircraft is able to follow a given trajectory with high precision in the 3D space domain ($R_h = 0NM, R_v = 0feet$), the simulations are performed considering successively aircraft maximum time uncertainty, t_ϵ, of 1 up to 4 minutes, respectively. The initial and final interaction between trajectories and the required computation time are reported in Table 1.5. Remark that the initial total interactions between trajectories are significantly smaller than those of the worst-case-oriented approach. This is not surprising, since in the later (deterministic) case one counts one interaction when in the former (probabilistic) case there is even only a tiny positive probability of conflict.

The proposed strategic trajectory planning methodology is able to find interaction-free trajectory planning for all cases. When considering higher level of time uncertainty (4 minutes), the solution space becomes more constrained and therefore the algorithm requires more computation time to converge

Now we test the algorithm with the continent-size air traffic considering en-route air traffic.The parameter values that specify the problem under consideration are, here again, the same as those given in Table 1.1. The parameters of the hybrid SA / hill climbing are the same as those given in Table 1.2, with the number of iterations at each temperature step, N_I empirically set to 2,000, more than for the above, smaller, national-size instance (N_I =200).

The initial and final interactions between trajectories, and computation time to solve the problem are reported in Table 1.6. Recall again that, as in the case without uncertainty, alternative flight levels for this continent-size instances are not available. Therefore, due to this lack of data, these problem instances can be separated only by modifying the horizontal flight profile and by modifying the departure time of aircraft. Nevertheless, there still remains less than 7% of the initial interactions taking into account the probabilistic type time uncertainty.

Table 1.6 Numerical results for the continent-size instances, considering two different levels of time uncertainty based on probabilistic uncertainty model.

t_ϵ (seconds)	initial Φ^P_{tot}	final Φ^P_{tot}	solved interaction	CPU time (minutes)	no. of iterations
60	529,555.5	12,550.0	97.6 %	1,341.7	1,328,152
120	1,079,738.4	40,706.2	96.2 %	2,254.2	2,231,881

1.7 CONCLUSIONS

In this paper, we have presented a methodology to solve 4D trajectory planning problem considering uncertainty of aircraft position and arrival time at strategic planning level. First, the uncertainties have been modeled with deterministic sets. The algorithm was tested on national-size and continent-size air traffic. To avoid being too conservative, probabilistic-type uncertainty sets were then considered.

The level of uncertainty to be considered is a trade-off between the desired robustness of the solution obtained and the associated trajectory modifications costs, to be decided by the user. Considering too important uncertainty in strategic planning will, indeed, results in a lost of capacity, since large portions of airspace have to be cleared for a given aircraft for a long period of time. Instead, the user can consider lower uncertainty levels, and iteratively solve the remaining interactions during pre-tactical and tactical phases.

References

1. C. Allignol, N. Barnier, and A. Gondran. Optimized vertical separation in Europe. In *DASC 2012, the 31st IEEE/AIAA Digital Avionics Systems Conference*, pages 4B3–1–4B3–10, October 2012.
2. N. Barnier and C. Allignol. 4D - trajectory deconfliction through departure time adjustment. In *ATM 2009, the 8th USA/Europe Air Traffic Management Research and Development Seminar*, Napa (California), 2009.
3. N. Barnier and C. Allignol. Combining flight level allocation with ground holding to optimize 4D-deconfliction. In *ATM 2011, the 9th USA/Europe Air Traffic Management Research and Development Seminar*, Berlin (Germany), 2011.
4. S. Chaimatanan. *Planification Stratégique de Trajectoires D'avions*. 2014.
5. S. Chaimatanan, D. Delahaye, and M. Mongeau. A methodology for strategic planning of aircraft trajectories using simulated annealing. In *ISIATM 2012, the 1st International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, Florida, 2012. http://hal-enac.archives-ouvertes.fr/hal-00912772/PDF/Chaimatanan/ISIATM2012.pdf.
6. S. Chaimatanan, D. Delahaye, and M. Mongeau. Strategic deconfliction of aircraft trajectories. In *ISIATM 2013, the 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, Toulouse (France), 2013. http://hal-enac.archives-ouvertes.fr/hal-00868450/PDF/isiatm2013/submission/123.pdf.
7. S. Chaimatanan, D. Delahaye, and M. Mongeau. A hybrid metaheuristic optimization algorithm for strategic planning of 4d aircraft trajectories at the continental scale. *Computational Intelligence Magazine, IEEE*, 9(4):46–61, Nov 2014.
8. S. Constans, B. Fontaine, and R. Fondacci. Minimizing potential conflict quantity with speed control. In *The 4th Eurocontrol Innovative Research Workshop And Exhibition*, pages 265–274, Bretigny-sur-Orge (France), 2005.
9. N. Dougui, D. Delahaye, and M. Mongeau. A new method for generating optimal conflict free 4D trajectory. In *The 4th International conference on research in air transportation*, pages 185–191, Budapest (Hungary), 2010.
10. N. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau. A light-propagation model for aircraft trajectory planning. *Journal of Global Optimization*, 56(3):873–895, 2013.
11. J. Dreo, A. Petrowski, P. Siarry, and E. Taillard. *Metaheuristics for hard optimization*. Springer, 2006.
12. N. Durand and J. B. Gotteland. Genetic algorithms applied to air traffic management. In *Metaheuristics for Hard Optimization*, pages 277–306. Springer, 2006.
13. M. R. Jardin. *Towards Real Time En Route Air Traffic Control Optimization*. PhD thesis, Standford University, 2003.
14. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
15. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
16. E.-G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley, 2009.
17. V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.