A Simulated Annealing Approach to 3D Strategic Aircraft Deconfliction based on En-Route Speed Changes under Wind and Temperature Uncertainties

Valentin Courchelle, Manuel Soler, Daniel González-Arribas, Daniel Delahaye

Courchelle, Valentin; Soler, Manuel; González-Arribas, Daniel; Delahay, Daniel. (2019). A simulated annealing approach to 3D strategic aircraft deconfliction based on en-route speed changes under wind and temperature uncertainties. *Transportation Research Part C: Emerging Technologies*, v. 103, pp.: 194-210.

DOI: https://doi.org/10.1016/j.trc.2019.03.024

Strategic Aircraft Deconfliction under Wind Uncertainties: A Simulated Annealing Metaheuristic Approach based on En-Route Speed Changes

Valentin Courchelle, Manuel Soler, Daniel González-Arribas, and Daniel Delahaye[1]

[a] *ENAC -Ecole Nationale de l'Aviation Civile-Toulouse, France. email: valentin.courchelle@gmail.com*
[b] *Department of Bioengineering and Aerospace Engineering, Universidad Carlos III, Madrid, Spain. e-mail: masolera@ing.uc3m.es*
[c] *ENAC -Ecole Nationale de l'Aviation Civile-Toulouse, France. email: delahaye@recherche.enac.fr*
[d] *Department of Bioengineering and Aerospace Engineering, Universidad Carlos III, Madrid, Spain. e-mail: masolera@ing.uc3m.es*

Abstract

We tackle the problem of minimizing the number of aircraft potential conflicts via speed regulations and taking into account uncertainties on aircraft position due to wind. The resolution is done at a strategic level, before any of the aircraft has departed. Owing to the complexity of this kind of optimisation problem, a simulated annealing metaheuristic approach is employed. A scenario with four hours of traffic overflying the Spanish (structured, continental) airspace has been selected. Inputted traffic provides routes, speed profiles (considered to be constant), and altitude profiles as in their flight plans. Probabilistic weather forecasts from an Ensemble Prediction System are employed. Solutions provide constant speed profiles that slightly differ from those in the flight plans. It is shown that the number of conflicts can be significantly reduced by slightly modifying flight plan speeds while not altering the routes, nor the altitudes, both selected by the airspace user. The impact of this resolution strategy in flight efficiency is also analyzed.

Keywords: ATM, Strategic Deconfliction, Speed control, Wind uncertainties, Simulated annealing

1. Introduction

Increasing air traffic is expected in the coming decades. An Air Traffic Management (ATM) paradigm shift is required to cope with it, aimed at increasing its capacity, enhancing its safety, improving its cost-effectiveness, and reducing the aviation related environmental impact. These objectives are being pursued

in different regions of the world, fostered by R&D programmes such as NextGen[1] in the US, SESAR[2] in Europe, or CARATS[3] in Japan.

Many different operational procedures and tools are envisioned within these programs, including the development of advanced algorithms and tools capable of anticipating conflict detection and resolution, which could lighten future air traffic controller (ATC) workload, and indirectly increase ATM's capacity. In particular, the development of computer aided conflict resolution tools at different look-ahead times (from tactical to strategic timescales) is aligned with the goals and technological solutions pursued within SESAR. It is however paramount to cope with uncertainty, which becomes more important as the look-ahead resolution time increases.

A large number of strategies have been proposed for the so-called conflict detection and resolution problem; refer for instance to Martín-Campo (2010) for a review. According to the studied time horizon, conflict detection & resolution algorithms can be classified into tactical (real time algorithms, usually considering only a sector) and strategic (planning level algorithms at network scales). These methods can be further classified as stochastic or deterministic, depending on whether system dynamics are modelled taking uncertainties into consideration or not.

The tactical, deterministic conflict detection and resolution problem (also referred to as collision avoidance problem) has been extensively treated in the literature. Solving approaches to the problem include, among others: analytical methods (e.g., Bicchi and Pallottino (2000)); methods based on operations research techniques, i.e., employing mixed-integer and mixed-integer nonlinear optimisation models (e.g., Pallottino et al. (2002); Alonso-Ayuso et al. (2011); Cafieri and Durand (2014)); or metaheuristics such as genetic or ant colony algorithms, (e.g., Durand et al. (1996); Durand and Alliot (2009)). While analytical approaches provide solution manoeuvres based on continuous curves; the latter two approches provide discrete advisory manoeuvres based on either speed changes, vectoring (heading changes), FL changes, or any possible combination of them. As for the tactical, stochastic conflict detection and resolution problem, different probabilistic approaches have been also explored, e.g., Maria Prandini and Sastry (2000); Lecchini et al. (2006); Liu and Hwang (2011, 2014); Matsuno et al. (2015); Hernández et al. (2017). The latter is the only one to use Ensemble Probabilistic Forecast (EPS) to model wind uncertainties as we do herein.

The nature of the tactical conflict detection and resolution methods restricts the problem to a limited number of aircraft, ranging from two (in a two aircraft encounter) to values of 40 to 60 at the maximum (considering the whole sector), which constitutes a challenging combinatorial problem. The strategic (in this context, before departure) conflict resolution would consider macroscale traffic

[1] https://www.faa.gov/nextgen/
[2] http://www.sesarju.eu/
[3] Collaborative Actions for Renovation of Air Traffic Systems

and would typically seek to deviate as little as possible aircraft from the original aircraft flight plan, minimising the impact of the separation maneuvers on the flight efficiency. It is thus required to consider a large-scale airspace and deal with thousands of flights. In such scenarios, the problem becomes cumbersome and would become very difficult to solve using classical optimisation techniques. Previous work on strategic, deterministic conflict detection and resolution includes, for instance, Vela et al. (2009), which uses FL assignment and speed control, or Ruiz et al. (2014), who analyse real air traffic on a day in the European airspace and resolves conflicts by heading changes.

Nevertheless, the consideration of uncertainties in strategic deconfliction remains an understudied problem. Existing work includes Chaimatanan et al. (2014) and Rodionova et al. (2016), considering uncertainties in the European airspace and in the North Atlantic oceanic airspace, respectively. Both resolved conflicts following ground delay strategies and route modifications. On the contrary, and to the best of the authors' understanding, macroscale traffic deconfliction under uncertainties through speed changes has not been addressed in the literature.

Thus, the main contribution of this paper is to propose a strategic deconfliction algorithm through speed regulations that extends our previous work in Courchelle et al. (2017) by: incorporating temperature uncertainty in the calculation of Mach number; extending the modeling to three-dimensional motion; and quantifying the impact of the propose speed regulation in terms of flight time and fuel. Thus, we are herein considering both wind and temperature to be the only sources of uncertainty (though we note that other sources of uncertainty could be incorporated in the model in a similar fashion). Wind uncertainty affects ground speed and temperature the Mach number, turning into uncertainty in aircraft position. We propose a metaheuristic solving approach based on simulated annealing.

A macroscale traffic application is presented as illustration. The data set originates from Eurocontrol's DDR2 database and corresponds to all air traffic overflying Spanish airspace the 26th of July in 2016 between 12. am and 4. pm., which is composed by a total of 1060 flights. Information retrieved includes route (waypoints and airways), cruising Mach number, altitude profile (including cruising flight level), and aircraft type. Wind and temperature uncertainties are retrieved from MétéoFrance PEARP ensemble probabilistic forecasts (EPS). The number of conflicts is minimized by small speed deviations from the flight plan speed, while respecting airline preferences (as filled in the flight plan) on route and vertical profile. The optimization problem is solved using a metahueristic approach based on Simulating Annealing. The resolution advisories reduce the number of potential conflicts at the strategic level, in this context before all aircraft's departure. Its impact on traffic performance is also measured analyzing the deviations in flight times and fuel consumptions.

The rest of this manuscript is organised as follows: Section 2 introduces the employed probabilistic wind forecast and associated uncertainties. Section 3 describes the mathematical modelling. Section 4 describes the simulated annealing algorithm to solve the problem. Section 6 presents the numerical results for the

chosen scenario. Finally, some conclusions and future directions of research are drawn in Section 7.

2. Atmospheric uncertainty

Uncertainty of wind fields and temperature will be drawn from Ensemble Prediction Systems (EPS). Ensemble forecasting is a prediction technique that generates a representative sample of the possible future states of the atmosphere. An ensemble forecast is a collection of typically 10 to 50 weather forecasts (referred to as members) with a common valid time, which can be obtained using different Numerical Weather Prediction (NWP) models with varying initial conditions or model parameters. The spread of solutions can be used as a measure of uncertainty. In this paper we focus on the output data of the global ensemble forecast system MétéoFrance PEARP EPS. Data can be accessed (among other EPS forecasts) at the TIGGE dataset, hosted by the European Center for Medium-Range Weather Forecasts (ECMWF).

2.1. MétéoFrance PEARP EPS

The MétéoFrance PEARP (Prévision d'Ensemble ARPège) is the probabilistic form of the MétéoFrance global numerical weather prediction model ARPEGE. It represents uncertainty in the initial conditions by creating a set of 34 forecasts starting from slightly different states that are closed, but not identical, to our best estimate of the initial state of the atmosphere (the control). All in all, The EPS probabilistic forecast is based on 35 integrations (or members) with approximately 10-km resolution in France (60-km at the antipodes) performing forecasts up to 4.5 days with 90 vertical levels.

3. Mathematical modelling

3.1. Set definition

Formulation of the model requires the definition of the following sets:

$\mathcal{F}$:= set of flights
F^+ := set of fictitious aircraft *ahead*
F^- := set of fictitious aircraft *behind*
$\mathcal{E}$:= set of ensemble members
$\mathcal{W}$:= set of waypoints
$\mathcal{W}^f$:= set of waypoints overflown by flight $f \in \mathcal{F}$
$\mathcal{A}$:= set of airways
$\mathcal{A}^f$:= set of airways overflown by flight $f \in \mathcal{F}$
$\mathcal{R}$:= set of routes (being r^f the route of $f \in \mathcal{F}$)
$\mathcal{M}$:= set of Mach numbers (being M^f the Mach number of $f \in \mathcal{F}$)
$\mathcal{FL}$:= set of Flight Levels (being h^f the FL of $f \in \mathcal{F}$)
$\mathcal{L}$:= set of links
$\mathcal{V}$:= set of vertexes
$\mathcal{C}_l^0$:= set of potential link conflicts
$\mathcal{C}_v^1$:= set of potential vertex conflicts of the first kind
$\mathcal{C}_v^2$:= set of potential vertex conflicts of the second kind
$\mathcal{C}_v^3$:= set of potential vertex conflicts of the third kind
$\mathcal{C}_{h,l}^4$:= set of potential vertical conflicts on link l
$\mathcal{C}_{h,v}^5$:= set of potential vertical conflicts on vertex v

3.2. Problem modelling

Let us consider a set of flights $\mathcal{F}$ scheduled to overly a given portion of airspace in a time interval. The flights take place in a structured airspace that can be modeled as a graph $\mathcal{G}(\mathcal{W}, \mathcal{A})$, where vertexes $\mathcal{W}$ represent waypoints and arcs $\mathcal{A}$ represent airways connecting them. Let us also consider a set of routes $\mathcal{R}$, a set of Mach speeds $\mathcal{M}$, and a set of flight levels $\mathcal{FL}$. For each flight $f \in \mathcal{F}$, there is a given route $r^f \in \mathcal{R}$, a nominal speed profile $M^f \in \mathcal{M}$, and an altitude profile $h^f \in \mathcal{FL}$ as if they were calculated in the flight plan. Thus, all three are inputs to the model.

Each route r^f can be modeled as a sequence of $n+1$ waypoints and a sequence of n airways that connect them, i.e.,

$$r^f = \left((w_0^f, \ldots, w_n^f), (a_0^f, \ldots, a_{n-1}^f) \right), w_i^f \in \mathcal{W}, a_i^f \in \mathcal{A}$$

where airway a_i connects waypoint w_i to w_{i+1}. We assume r^f is fixed, and thus will not be subject to modifications in the resolution procedure.

Similarly, altitude profile h^f can be modeled as a sequence of altitude values along the route r^f, i.e., $h^f = (h_1^f, \ldots, h_n^f)$, with $h_i^f \in \mathcal{FL}$. We assume h^f to be also an invariant, and thus will not be subject to modifications in the resolution procedure.

The speed profile $M^f \in \mathcal{M}$ is given as a Mach number and its assumed to be constant during the flight. Nevertheless, the Mach number will be subject to adjustments, i.e., we will define it as

$$M^f = M_0^f + \Delta M^f,$$

being M_0^f the nominal cruising Mach number (retrieved as input from the Flight Plan) and ΔM^f the speed adjustments that act as decision variables in the optimization problem (please refer to Section 3.5).

All in all, we will provide a solution that respects the original route and altitude profile stated in the flight plans, however providing speed profiles that (though constant) slightly differ from those in the flight plans. It is this compliant with current operational practices.

3.2.1. Ground speed

Let us now consider the horizontal motion of an aircraft with respect to Earth, governed by its ground speed. True airspeed $\vec{v}_{TAS}$, ground speed $\vec{v}_g$, and wind speed $\vec{w}$ are related by the vector composition:

$$\vec{v}_g = \vec{v}_{TAS} + \vec{w}, \tag{1}$$

Let $\hat{u}_\psi$ be a unit vector parallel to the ground speed vector (i.e. a vector that follows the track, and thus can be determined from the route and the location within it). Let the along-track and cross-track components of the wind be defined as:

$$w_{at} = \vec{w} \cdot \hat{u}_\psi, \tag{2}$$

$$w_{ct} = ||\vec{w} \times \hat{u}_\psi||, \tag{3}$$

where $||\cdot||$ denotes the standard Euclidean norm of a vector. If we denote $v_{TAS} = ||\vec{v}_{TAS}||$, then the ground speed can be determined by the following equation, which follows from Equation (1):

$$v_g = ||\vec{v}_g|| = \sqrt{v_{TAS}^2 - w_{ct}^2} + w_{at}. \tag{4}$$

Therefore, for any flight f given its route r^f and its airspeed $\vec{v}_a^f$, the wind speed univocally determines the groundspeed at each longitude/latitude point (λ, ϕ) in the route, i.e., $v_g^f(\vec{v}_a(\lambda, \phi), \vec{w}((\lambda, \phi))$. For the sake of clarity, we will omit these functional dependences in the sequel.

Notice also that for convenience, we can manipulate Mach speed (M) and readily transform it into True Airspeed (v_a) using the speed of sound (v_{sound}).

The formula that relates these variables reads as follows:

$$v_a = M \cdot v_{sound}, \tag{5}$$

where

$$v_{sound} = \sqrt{\gamma R T},$$

being γ the adiabatic constant of gases, R the gas constant for the air, and T the absolute temperature. For the air $\gamma = 1.4$ and $R = 287.05 J/kgK$. The absolute Temperature will be extracted from atmospheric forecasts and considered to be uncertain.

3.2.2. Wind and temperature uncertainty modelling

In order to include wind and temperature uncertainties into the model, we use the Météo France PEARP ensemble forecast as described in Section 2. Let us denote the EPS forecast $\mathcal{E}$. For each ensemble member $e \in \mathcal{E} = \{1, \ldots, n_{memb}\}$, we get n_{memb} wind and temperature forecast values, i.e., $\vec{w}^e$ and T^e, at each of the grid points (lat, long, pressure level) provided in the EPS forecast. For a given flight f, and applying bilinear interpolation of data to the waypoints/airways in $\mathcal{G}(\mathcal{W}, \mathcal{A})$, one can easily obtain the n_{memb} possible realizations of ground speed in its route $v_g^{f,e}$. We will employ them to evaluate and eventually solve potential conflicts.

We will represent all points in time as their difference, in seconds, with respect to an arbitrary reference time, common to all trajectories. For flight $f \in \mathcal{F}$, we consider departure time $t_0^f \in \mathbb{R}$ and flight plan route $r^f \in \mathcal{R}$ and compute for each ensemble member $e \in \mathcal{E}$ the arrival time, denoted as the final waypoint's (w_n^f) flyover time:

$$t_n^{f,e} \in \mathbb{R}; \quad \forall f \in \mathcal{F},\ e \in \mathcal{E}, \tag{6}$$

and index n denoting the final waypoint in the route r^f.

Different statistical metrics can be obtained, e.g., the mean time and the range of times, out of this set of possible arrival times $\{t_n^{f,e}\}_{e \in \mathcal{E}}$ for flight f. Figure 1(a) illustrates this idea. We define the mean arrival time as:

$$\bar{t}_n^f = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} t_n^{f,e}. \tag{7}$$

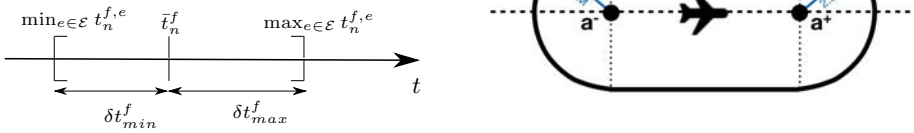
We define the maximum error on arrival times as follows:

$$\Delta t_n^f = \max(\delta t_{min}^f, \delta t_{max}^f), \quad f \in \mathcal{F}, \tag{8}$$

where

$$\delta t_{max}^f = \max_{e \in \mathcal{E}} t_n^{f,e} - \bar{t}_n^f, \quad \text{and}$$

$$\delta t_{min}^f = \bar{t}_n^f - \min_{e \in \mathcal{E}} t_n^{f,e}.$$



(a) Aircraft's a time arrival error. (b) Protected area around aircraft a.

Figure 1: Aircraft's time uncertainty and protected area.

We will now consider as the uncertain arrival time the interval $[\bar{t}_n^f - \Delta t_n^f, \bar{t}_n^f + \Delta t_n^f]$. In order to obtain an interval of uncertainty at any other arbitrary point in the trajectory, we assume that the uncertainty grows linearly from zero at departure to its maximum value at arrival:

$$\Delta t^f(t) = \frac{t - t_0^f}{\bar{t}_n^f - t_0^f} \Delta t_n^f, \qquad \forall f \in \mathcal{F}, \forall t \in [t_0^f, t_n^f]. \tag{9}$$

Two fictitious positions can be constructed by adding and substracting $\Delta t^f(t)$; they bound a segment of possible positions of aircraft f. We define two sets containing these imaginary flights: $\mathcal{F}^+$ (ahead) and $\mathcal{F}^-$ (behind). Therefore, $f^+ \in \mathcal{F}^+$ denotes a flight that is identical to f, but its flyover times are shifted such that $t_i^{f^+} = \bar{t}_i^f - \Delta t^f(\bar{t}_i^f)$; analogously, flyover times for $f^- \in \mathcal{F}^-$ are $t_i^{f^-} = \bar{t}_i^f + \Delta t^f(\bar{t}_i^f)$.

3.3. Horizontal conflict evaluation

We require a separation of 5 NM between aircraft in the horizontal plane. We define a protected area composed by the area within a 2.5 NM radius of any potential position of the plane within the uncertainty interval (see Figure 1(b)). We say that two aircraft at the same flight level are in conflict whenever their protected areas intersect. In order to evaluate conflicts we define a numerical grid on $\mathcal{G}(\mathcal{W}, \mathcal{A})$: each waypoint has an associated grid node (termed herein vertex); airways' tracks are discretized with equidistant grid nodes (termed links).

Let us consider a pair of aircraft $f_1, f_2 \in \mathcal{F}$. Let us also consider the set of link and vertex grid points $\mathcal{L}$ and $\mathcal{V}$, respectively. Two types of conflicts can be distinguished within $r^{f_1} \in \mathcal{R}$ and $r^{f_2} \in \mathcal{R}$:

- The first type takes place when (f_1, f_2) share the same airway $a_i^{f_1} \equiv a_j^{f_2}$ in their corresponding routes (with i, j denoting indexes for their sequence of arcs). In other words, both aircraft are following the same airway one ahead of the other. This conflict configuration are evaluated at the *links*, $l \in \mathcal{L}$. When the conflict exists under this configuration, we denominate it a *link conflict*.

- The second type occurs when (f_1, f_2) share the same waypoint $w_i^{f_1} \equiv w_j^{f_2}$ in their corresponding routes (with i, j denoting indexes for their sequence of waypoints). In other words, both routes are scheduled to overfly the same waypoint, flying however two different airways (converging at that waypoint). This conflict configuration are evaluated at the *vertexes*, $v \in \mathcal{V}$. This conflict configuration is denominated *vertex conflict*.

3.3.1. Link conflict:

Consider the two aircraft f_1 and f_2 flying on link l such that f_1 is ahead of f_2. In other words, two aircraft that follow the same airway, one ahead of the other. Let $v_g^{f_1,e}(l)$ and $v_g^{f_2,e}(l)$ be the ground speeds of aircraft f_1 and f_2 on link l and ensemble member e, respectively. Let us additionally consider node $f_1^- \in \mathcal{F}^-$, where the conflict would be evaluated.

Note that the time

$$t_l^{f_1^-} + \frac{S_0}{v_g^{f_1,e}(l)}$$

corresponds to the latest time of arrival of aircraft f_1 at node l plus the time to fly the minimum distance $S_0 = 5NM$ at the given ground speed. Then the time interval

$$\left[t_l^{f_1^-}, t_l^{f_1^-} + \frac{S_0}{v_g^{f_1,e}(l)}\right]$$

represents the minimum separation buffer (at aircraft's f_1 ground speed).

Let us consider that ground speed to be the minimum one (across all members), i.e., let us denote $v_g^{f_1}(l) = \min_{e \in E} v_g^{f_1,e}(l)$. Then, if the condition

$$t_l^{f_2^+} - \left(t_l^{f_1^-} + \frac{S_0}{v_g^{f_1}(l)}\right) < 0 \tag{10}$$

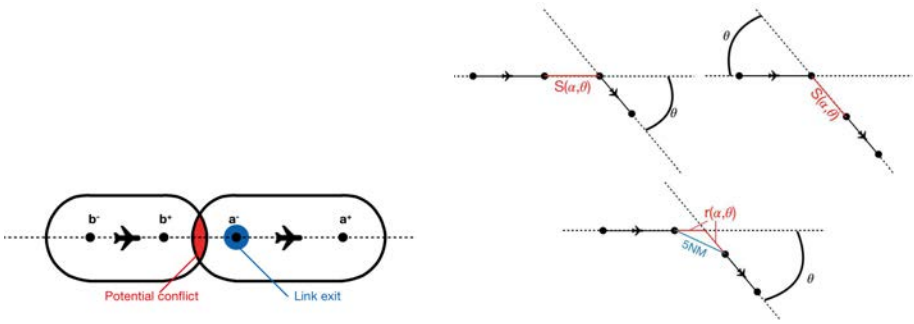
holds, there would be a conflict. In other words, if the earliest time of arrival of aircraft f_2 at link l, i.e., $t_l^{f_2^+}$, overlaps with the safety buffer around the latest time of arrival of aircraft f_1 at link l, then there is a conflict.

A similar rationale holds *mutatis mutandis* for the conflict analysis made over node $b^+ \in B^+$. In this situation, if

$$\left(t_l^{f_2^+} - \frac{S_0}{v_g^{f_2}(l)}\right) - t_l^{f_1^-} < 0 \tag{11}$$

holds, then there would be a conflict. See Figure 2(a).

To evaluate all conflicts that occur on link l, let us first define the indicator functions $\mathbb{1}_{Eq.(10)}$ and $\mathbb{1}_{Eq.(11)}$, which take value one if equations (10) and (11) are satisfied, respectively, and zero otherwise. Then, let us define the following



(a) Link conflict. (b) Vertex conflict (3 configs).

Figure 2: Link and Vertex conflicts.

function:

$$\phi_{\mathcal{L}}(l) = -\sum_{(f_1,f_2)\in\mathcal{C}_l^0}\left(t_l^{f_2^+} - (t_l^{f_1^-} + \frac{S_0}{v_g^{f_1}(l)})\right)\cdot \mathbb{1}_{Eq.(10)} - \sum_{(f_1,f_2)\in\mathcal{C}_l^0}\left(\left(t_l^{f_2^+} - \frac{S_0}{v_g^{f_2}(l)}\right) - t_l^{f_1^-}\right)\cdot \mathbb{1}_{Eq.(11)};\ \ \forall l\in\mathcal{L} \quad (12)$$

where $\mathcal{L}$ is the set of links and $\mathcal{C}_l^0$ is the set of potential conflicts, i.e., the set of aircraft pairs (f_1, f_2) involved into a conflict at the entry link l, with the pair ordered such that f_1 flies ahead of f_2. Notice that Eq. (12) counts the conflict twice, looking at the potential soonest time of arrival of aircraft f_2 and the potential latest time of arrival of aircraft f_1. This is necessary in order to account for all possible relative ground speeds of (f_1, f_2) (f_1 flying faster or slower than f_2), even though in some situations it results in counting the conflict twice. By construction, this function is positive: the *indicator* functions indicate whether there is conflict or not, whereas the remaining term weights the severity of the conflict.

3.3.2. Vertex conflict:

Consider again two aircraft f_1 and f_2 flying now on two different airways that converge in a waypoint. Assume they fly sufficiently close to the waypoint. Should a loss of separation minima occur in that situation, we say there is a vertex conflict. Three different cases need to be modelled to cover all configurations. These are illustrated in Figure 2(b).

For the first two configurations -upper sketches in Figure 2(b)- two aircraft f_1 and f_2 are considered such that in the vicinity of vertex $v \in \mathcal{V}$ they follow different links (different airways) and that one is sufficiently ahead of the other. Thus, their tracks differ with an angle $\theta_v^{(f_1,f_2)}$, which in turn has an impact on the minimum required distance between f_1^- and f_2^+ (related to their difference in

velocities). The required distance is $S(\alpha^{(f_1,f_2)}, \theta_v^{(f_1,f_2)}) > 5NM$, where $\alpha^{(f_1,f_2)}$ is the ratio between the ground speeds (f_1 over f_2).

Following the proof in Rey et al. (2010):

$$S(\alpha^{(f_1,f_2)}, \theta_v^{(f_1,f_2)}) = S_0 \times \frac{\sqrt{\alpha^{(f_1,f_2)} \cdot \alpha^{(f_1,f_2)} - 2 \cdot \alpha^{(f_1,f_2)} \cdot \cos(\theta_v^{(f_1,f_2)}) + 1}}{|\sin(\theta_v^{(f_1,f_2)})|}.$$

As for the third configuration -lower sketch in Figure 2(b)-, the separation distance is set to be 5NM between f_1^- and f_2^+ when they are both at distance $S'(\alpha^{(f_1,f_2)}, \theta_v^{(f_1,f_2)})$ from vertex v. Then this distance is:

$$S'(\alpha^{(f_1,f_2)}, \theta_v^{(f_1,f_2)}) = S_0 \times \frac{\sqrt{\alpha^{(f_1,f_2)} \cdot \alpha^{(f_1,f_2)} - 2 \cdot \alpha^{(f_1,f_2)} \cdot \cos(\theta_v^{(f_1,f_2)}) + 1}}{2 \cdot \cos(\frac{\theta_v^{(f_1,f_2)}}{2})}.$$

As it is done for a link conflict, we focus on the overlapping of the specific intervals. If an overlap is revealed, aircraft f_1 and f_2 are in conflict. This is materialized should the following equations be fulfilled:

$$\left(t_n^{f_2^+} - \frac{S(\alpha^{(f_1,f_2)}, \theta_v^{(f_1,f_2)})}{v_g^{f_2}(v)}\right) - t_n^{f_1^-} < 0; \tag{13}$$

$$t_n^{f_2^+} - \left(t_n^{f_1^-} + \frac{S(\alpha^{(f_1,f_2)}, \theta_v^{(f_1,f_2)})}{v_g^{f_1}(v)}\right) < 0; \tag{14}$$

$$\left(t_n^{f_2^+} - \frac{S'(\alpha^{(f_1,f_2)}, \theta_v^{(f_1,f_2)})}{v_g^{f_2}(v)}\right) - \left(t_n^{f_1^-} + \frac{S'(\alpha^{(f_1,f_2)}, \theta_v^{(f_1,f_2)})}{v_g^{f_1}(v)}\right) < 0. \tag{15}$$

Consequently, let us define the following three indicator functions $\mathbb{1}_{Eq.(13)}$, $\mathbb{1}_{Eq.(14)}$, and $\mathbb{1}_{Eq.(15)}$, which take value one if the corresponding conditions fulfilled, and zero otherwise. To evaluate all conflicts which occur on a vertex v we define the following function:

$$\phi_{\mathcal{V}}(v) = \underbrace{-\sum_{(f_1,f_2)\in\mathcal{C}_v^1}\left(\left(t_n^{f_2^+} - \frac{S(\alpha^{(f_1,f_2)},\theta_v^{(f_1,f_2)})}{v_g^{f_2}(v)}\right) - t_n^{f_1^-}\right)\cdot \mathbb{1}_{Eq.(13)}}_{\text{conflicts config. 1}}$$
$$\underbrace{-\sum_{(f_1,f_2)\in\mathcal{C}_v^2}\left(t_n^{f_2^+} - \left(t_n^{f_1^-} + \frac{S(\alpha^{(f_1,f_2)},\theta_v^{(f_1,f_2)})}{v_g^{f_1}(v)}\right)\right)\cdot \mathbb{1}_{Eq.(14)}}_{\text{conflicts config. 2}}$$
$$\underbrace{-\sum_{(f_1,f_2)\in\mathcal{C}_v^3}\left(\left(t_n^{f_2^+} - \frac{S'(\alpha^{(f_1,f_2)},\theta_v^{(f_1,f_2)})}{v_g^{f_2}(v)}\right) - \left(t_n^{f_1^-} + \frac{S'(\alpha^{(f_1,f_2)},\theta_v^{(f_1,f_2)})}{v_g^{f_1}(v)}\right)\right)\cdot \mathbb{1}_{Eq.(15)}}_{\text{conflicts config. 3}}, \quad \forall v \in \mathcal{V} \tag{16}$$

where $\mathcal{V}$ is the set of vertexes and $\mathcal{C}_v^1$ is the set of aircraft pairs (f_1, f_2) involved into a conflict at the entry vertex v (and configuration 1), where f_1 reaches n before f_2. $\mathcal{C}_v^2$ and $\mathcal{C}_v^3$ denote the set of aircraft pairs (f_1, f_2) in conflict detected in the second and third configuration, respectively. By construction, this function is positive. Similarly, the *indicator* functions indicate whether there is conflict or not, whereas the remaining term weights the severity of the conflict.

3.4. Vertical conflict evaluation

Let us again consider a pair of aircraft $f_1, f_2 \in \mathcal{F}$. Each flight follows a given route $r^{f_1}, r^{f_2} \in \mathcal{R}$, a given speed profile $M^{f_1}, M^{f_2} \in \mathcal{M}$, and a given altitude profile $h^{f_1}, h^{f_2} \in \mathcal{FL}$. In other words, every flight plan provides the expected altitude along the route.

We evaluate vertical conflict (also refereed to as vertical loss of separation) in nodes and links, $v, l \in \mathcal{V}, \mathcal{L}$ at different flight levels. Recall that the structure of links and nodes is based on that structured airspace modeled as a Graph $\mathcal{G}(\mathcal{W}, \mathcal{A})$, expanded now into the vertical dimension. In oder words, we create copies of the link/node structure for each flight level. We assume vertical winds are neglectable (this hypothesis is consistent with reality, with vertical winds being typically two/three orders of magnitude lower than horizontal winds), thus altitude kinematics being deterministic. Nonetheless, horizontal wind uncertainty affects the vertical motion, e.g., introducing uncertainty in the top of climb/top of descent.

Then, in order to consider the vertical dimension, we add an additional inequality to the conditions for existence of a horizontal conflict. A conflict would exist between aircraft f_1 and f_2 at a vertex/link v, l, if and only if the difference between their altitudes is lower than a minimum required $H_0 = 1000$ ft., i.e.:

$$|h(t_l^{f_1-}) - h(t_l^{f_2+})| \leq H_0, \qquad \forall l \in \mathcal{L}. \tag{17}$$
$$|h(t_v^{f_1-}) - h(t_v^{f_2+})| \leq H_0, \qquad \forall v \in \mathcal{V}. \tag{18}$$

Notice that, even though the vertical motion is considered to be deterministic, the vertical separation is still coupled with the uncertainty associated to the ground speed in the horizontal motion. Notice also that the vertical conflict is only evaluated between aircraft flying the same link/node, however at different altitudes.

Let us denote:

$$|h(t_l^{f_1-}) - h(t_l^{f_2+})| = \Delta h_l \quad \forall l \in \mathcal{L}, (f_1, f_2) \in \mathcal{C}_{h,l}^4.$$

$$|h(t_v^{f_1-}) - h(t_v^{f_2+})| = \Delta h_v \quad \forall v \in \mathcal{V}, (f_1, f_2) \in \mathcal{C}_{h,v}^5.$$

and define the following vertical conflict indicator functions:

$$1_{\Delta h_l \leq H_0}(l) = \mathcal{H}\left(H_0 - \Delta h_l\right), \quad \forall l \in \mathcal{L}, (f_1, f_2) \in \mathcal{C}_{h,l}^4. \tag{19}$$

$$1_{\Delta h_v \leq H_0}(v) = \mathcal{H}\left(H_0 - \Delta h_v\right), \quad \forall v \in \mathcal{V}, (f_1, f_2) \in \mathcal{C}_{h,v}^5. \tag{20}$$

where $\mathcal{C}_{h,l}^4$ and $\mathcal{C}_{h,v}^5$ are the set of aircraft pairs (f_1, f_2) involved into a vertical loss of separation at the entry link/vertex l, v (f_1 reaches v, n before f_2), and in Equation (19) $\mathcal{H}$ denotes the Heaviside function, i.e.:

$$\mathcal{H}(H_0 - \Delta h_l) = \begin{cases} 0 & \text{if } H_0 - \Delta h_l \leq 0 \\ 1 & \text{if } H_0 - \Delta h_l > 0 \end{cases} \tag{21}$$

Notice that the Heaviside function takes value zero when the aircraft are separated more than H_0 and one when their are in vertical conflict. The reasoning holds *mutatis mutandis* for Equation (20).

Finally, let us define the vertical conflict count function as:

$$\phi_H = \phi_H(l) + \phi_H(v), \tag{22}$$

where:

$$\phi_H(l) = \sum_{(f_1,f_2)\in\mathcal{C}_{h,l}^4} \mathbb{1}_{\Delta h_l \leq H_0} \cdot \frac{1}{\Delta h_l}, \quad \forall l \in \mathcal{L}, \tag{23}$$

$$\phi_H(v) = \sum_{(f_1,f_2)\in\mathcal{C}_{h,v}^5} \mathbb{1}_{\Delta h_l \leq H_0} \cdot \frac{1}{\Delta h_v}, \quad \forall v \in \mathcal{V}, \tag{24}$$

where the indicator functions provide information on whether there is a vertical loss of separation or not and the factor $1/\Delta h_l$ $(1/\Delta h_v)$[4] weights the severity of the conflict, i.e., the closer the altitudes the higher the factor. Notice that both $\phi_H(l)$ and $\phi_H(v)$ are by construction nonnegative.

[4]In order to avoid the singularity, a slack parameter should be added.

3.5. Mathematical model set-up

3.5.1. Decision variables definition

Decision variables in the problem are the set of Mach advisories ΔM^{f_i}, $i = 1 \dots n_f \in \mathbb{Z}^{n_f}$ with dimension $n_f := |\mathcal{F}|$ equal to the number of flights considered. Each ΔM^{f_i} corresponds to a variation in Mach number applied to flight f_i. These velocity variations are integers, corresponding to increments of 0.01 Mach. They can be positive or negative in order to increase or decrease the cruising Mach (with respect to the nominal one provided in the flight plans).

Thus, variables to be considered in the problem are as follows:

$$\Delta M^{f_i} \in \mathbb{Z}, \qquad \forall f_i \in \mathcal{F}. \tag{25}$$

By construction, Mach number, which can be transformed into true are speed (affected by temperatures) and wind speeds determine the ground speeds needed in the conflict evaluation functions.

3.5.2. Constraints

In order to prevent too strong speed variations, we impose boundaries to the variables. A reasonable interval in which Mach variations should be located is $\pm 0.04M$. This is somehow consistent with the "subliminal control" concept envisioned for tactical deconfliction Cafieri and Durand (2014).

Let then the variable boundaries be:

$$-0.04 \leq \Delta M^{f_i} \leq 0.04, \qquad \forall f_i \in \mathcal{F}. \tag{26}$$

Recall that we can manipulate Mach speed (the input to the problem) and readily transform it into True Airspeed using the speed of sound (see Equation 5).

3.5.3. Objective function

The aim is to minimise conflicts with the least impact on aircraft performance. We define the function which evaluates conflicts corresponding to the possible values of decision variables ΔM^{f_i}, $i = 1..n_f \in \mathbb{Z}^{n_f}$ as follows:

$$\begin{aligned}\Phi(\Delta M^{f_i}) = &\underbrace{\sum_{l \in \mathcal{L}} \phi_{\mathcal{L}}(l) \cdot (1 - \mathcal{H}(\Delta h_l))}_{\text{Link conflicts}} + \underbrace{\sum_{v \in \mathcal{V}} \phi_{\mathcal{V}}(v) \cdot (1 - \mathcal{H}(\Delta h_v))}_{\text{Vertex conflicts}} \\ &+ \underbrace{\sum_{l \in \mathcal{L}} \phi_{\mathcal{L}}(l) \cdot \phi_H(l) \cdot \mathcal{H}(\Delta h_l)}_{\text{Vertical link conflicts}} + \underbrace{\sum_{v \in \mathcal{V}} \phi_{\mathcal{V}}(l) \cdot \phi_H(v) \cdot \mathcal{H}(\Delta h_v)}_{\text{Vertical vertex conflicts}}\end{aligned} \tag{27}$$

Notice that $\mathcal{H}(\Delta h_l)$ (idem for the vertexes) is a Heaviside function that takes value zero if altitudes are the same and one elsewhere. Thus, the interpretation of Equation (27) is as follows: horizontal conflicts (both link and vertex ones) are evaluated (are different than zero and positive) only if altitudes at links/nodes

coincide, i.e., both aircraft f_1 and f_2 fly at the same flight level; otherwise the horizontal conflict count is set to zero. The opposite, the vertical conflict count functions are evaluated (are different than zero and positive) if altitudes are different (and less than H_0 according to Eq. (23)-(24)) and the horizontal separation is violated; otherwise their value is zero.

So the objective function is:

$$min\ J = \underbrace{k \cdot n_f \cdot \Phi(\Delta M^{f_i})}_{\text{weighted conflict count}} + \underbrace{\sum_{i=1}^{n_f} \mid \Delta M^{f_i} \mid}_{\text{speed changes}} \tag{28}$$

where k is a parameter used to weight the minimisation of conflicts w.r.t speed changes.

3.5.4. Problem Statement

All in all, the problem is stated as follows:

$$min\ J = k \cdot n_f \cdot \Phi(\Delta M^{f_i}) + \sum_{i=1}^{n_f} \mid \Delta M^{f_i} \mid .$$

Subject to:

$$-0.04 \leq \Delta M^{f_i} \leq 0.04, \qquad \forall f_i \in \mathcal{F}.$$

where:

$$\Phi(\Delta V_a^f) \leftarrow (27)$$

$$\forall l \in \mathcal{L},\ \phi_{\mathcal{L}}(l) \leftarrow (10) - (11)$$

$$\forall v \in \mathcal{V},\quad \phi_{\mathcal{V}}(v) \leftarrow (16)$$

$$\forall l, v \in \mathcal{L}, \mathcal{V},\quad \phi_H \leftarrow (22)$$

$$\forall f \in \mathcal{F},\ \Delta M^{f_i} \in \mathbb{Z}$$

3.6. Complexity

For a given flight plan, we can compute the associated time windows (with the uncertainty margins) for any given point in the route. Potential conflicts between two aircraft will be then detected. The relationship *is in conflict with*, or *is in potential conflict with* defines an equivalence relation coined *cluster*. As described in Durand and Gotteland (2003), *if we restrict ourselves to the horizontal plane with n airplanes, we can find the presence of* $\frac{n(n-1)}{2}$ *potential conflicts*. It can be shown Durand (1996) that the set of possible solutions contains $2^{\frac{n(n-1)}{2}}$ connected components, which implies that it requires as many executions of the search algorithm for a local search optimisation. Thus, for a cluster with 6 aircraft, this represents 32,768 related components. The presence of as many components without knowing which one contains the optimal

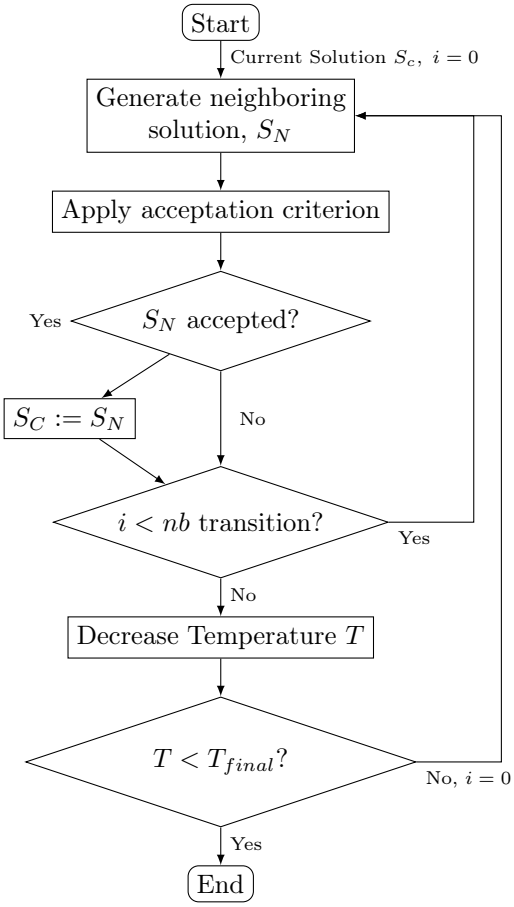


Figure 3: Simulated annealing algorithm.

solution make the problem highly combinatorial. That is the reason behind conflict resolution problems being hard optimisation problems, which motivates the employment of metaheuristic algorithms.

4. Simulated Annealing

4.1. Overview

Simulated annealing (SA) is a metaheuristic optimization method inspired by the annealing process in metallurgy, e.g., refer to Kirkpatrick et al. (1983). The annealing process is composed by a "heating" and a "cooling" process, which bring certain physical systems from an initial disordered random state to a minimum-energy one (for metallic alloys, it corresponds to crystallizations with desirable mechanical properties). Simulated annealing for mathematical

Algorithm 1 Neighbourhood function

Require: the flight conflict count set conflictCount to record the sum of number of conflicts for a subset of aircraft

procedure GENERATENEIGHBOUR
  Generate a random number p between 0 and 1;
  Calculate the total number of conflicts, sumConf in the flight set
  if sumConf > 0 **then**
    target $\leftarrow$ sumConf $\times\ p$;
    $sum \leftarrow 0$;
    while $sum < target$ **do**;
      $i \leftarrow$ iStart ▷ iStart is the beginning index of flight set
      $sum \leftarrow sum +$ conflictCount$[i]$;
      $i \leftarrow i + 1$;
    end while
  else
    $i \leftarrow$random number between iStart and jEnd; ▷ jEnd is the ending index of active flight set
  end if
  Save the current decision variables;
  Change the decision variable of flight i i.e. the speed change;
  Update the flight set information;
end procedure

optimization uses analogous processes to bring an objective function to a local minimum. A global parameter, the temperature T, regulates these two processes. In each iteration, the SA algorithm compares a generated neighbouring state solution (S_N) to its current one (S_C). The evolution from one solution to the following is done in a probabilistic fashion. When T is high, deteriorated solutions (with high energy) are more likely to be accepted. As T decreases, better solutions (in terms of optimality) are obtained. In the end, a state considered to be good enough (a local optima) is reached and accepted. Please, refer to Figure 3 for a diagrammatic representation.

SA is well known for its ability to avoid the stuck on local minima by allowing random neighbourhood changes. Moreover, it can be easily adapted to various kinds of problems with continuous or discrete space states. In the remaining of the Section we provide insight on the steps followed to adapt the SA algorithm to the particularities of the problem herein considered.

4.2. Neighbourhood function

A *neighbourhood function* is used to generate a local change from the actual solution ($S_{C,i}$) within each iteration i. Two criteria are considered in its design: low computational times on the one hand; the generation of local solution changes so as to avoid this search to resemble a pure random search on the other. The neighbourhood generation function is described in Algorithm 1.

The fact that the neighbourhood choice is based on the conflict number count increases the likelihood that a flight involving many conflicts would be regulated. Moreover, this function may preserve weak (suboptimal) solutions, which in turn may include some components that could be useful later in the annealing process.

4.3. Initial temperature and acceptance probabilities

The temperature parameter, T_i -at iteration i of the SA algorithm-, is used to determine the acceptance of a solution's degradation. The acceptance probability is given by the following formula:

$$P(Accept\ S_N|S_C) = \begin{cases} 1 & if\ S_N >> S_C \\ \exp\left(-\frac{|f(S_N)-f(S_C)|}{T_i}\right) \end{cases}$$

where S_C is the current solution, S_N the solution generated by the neighbourhood function and $S_N >> S_C$ can be interpreted as "S_N is better than S_C". Should T_i be high, then all neighbouring solutions would have a similar probability to be accepted and thus large degradation in solution's optimality is expected. In the limit, as T_i approaches infinity, all neighbours are systematically accepted. On the contrary, should T_i be low, worse solutions are far less likely to be accepted. The slower the rate of temperature decrease, the better the chances of finding an optimal solution, but the larger the total number of SA iterations (thereby increasing the computational time). In order to determine the initial temperature ($T_0 := T_{init}$), we choose a temperature that brings an acceptance rate of 80%, i.e., $P(Accept\ S_N|S_C) = 0.8$. This evaluating method is described by the *HeatUpLoop* procedure of Algorithm 2.

4.4. Cooling loop

In order to decrease the temperature, we use a geometric law that has been extensively used in SA applications:

$$T_{i+1} = T_i \times \alpha, \quad 0 < \alpha < 1, \quad i = 0 \dots N_{trans}.$$

Note that for both the heat up and cooling processes we also set a parameter termed number of transitions N_{trans}. It reflects the number of iterations allowed within both iterative processes.

At each iteration i, the temperature is multiplied by a coefficient α. The choice of α is delicate. Should α be too large, the temperature would decrease very slowly and the convergence would be slow as well. On the other hand, if the value α is too small, the temperature would decrease too fast and thus the algorithm might be quickly blocked in a local optimum. That's why this parameter has to be adapted to the problem as a trade off between computational efficiency and optimality gap. The cooling process is described by the CoolingLoop procedure of Algorithm 2.

Algorithm 2 Simulated Annealing

Require: initial temperature T, number of transitions $nbTransitions$

procedure HEATUPLOOP
  while $\chi_0 < 0.8$ **do** ▷ the accepted rate is 0.8
    acceptCount $\leftarrow 0$
    $T \leftarrow T \times 1.1$ ▷ heat up
    for $i = 0$ to $nbTransitions$ **do**
      initState($\vec{x}_i$);
      CriterionCalculation $y_i = f(\vec{x}_i)$;
      $\vec{x}_j$ = generateNeighbour($\vec{x}_i$);
      CriterionCalculation $y_j = f(\vec{x}_j)$;
      if accept(y_i,y_j,T,minimisation) **then**
        acceptCount++;
      end if
    end for
    χ_0 = acceptCount/nbTransitions;
  end while
  T_{init}=T;
  return T_{init}
end procedure

procedure COOLINGLOOP(T_{init})
  $\alpha \leftarrow 0.95$; ▷ geometrical law
  initState($\vec{x}_i$);
  CriterionCalculation $y_i = f(\vec{x}_i)$;
  $T = T_{init}$;
  while $T > \varepsilon \times T_{init}$ **do** ▷ ε defines ending temp.
    for $i = 0$ to nbTransitions **do**
      $\vec{x}_j$ = generateNeighbour($\vec{x}_i$);
      CriterionCalculation $y_j = f(\vec{x}_j)$;
      if accept(y_i,y_j,T,minimisation) **then**
        $x_i = \vec{x}_j$;
        $y_i = y_j$;
      end if
    end for
    $T = T \times \alpha$;
  end while
end procedure

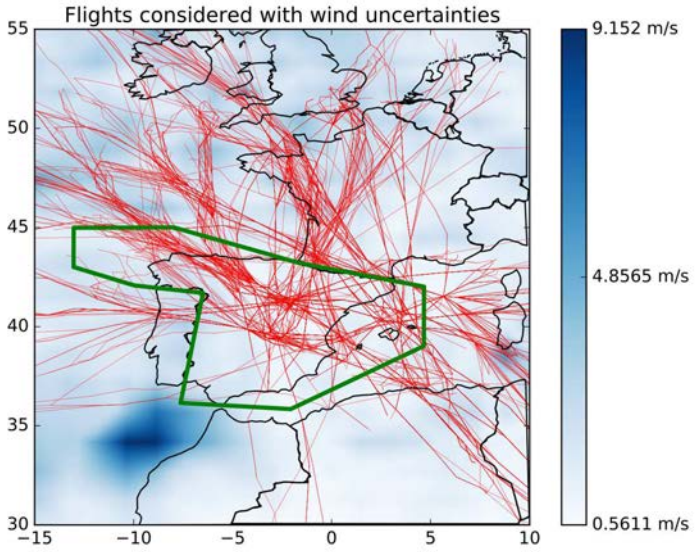


Figure 4: Visualisation of traffic (red). It corresponds to that over the Spanish airspace (green) on 26th July 2016 between 12. am and 4. pm. Contour plot denotes wind uncertainties -blues-.

4.4.1. Stopping criterion

The algorithm is terminated when the final temperature reaches the value $T_{\text{init}} \cdot \varepsilon$, where ε is a slack coefficient, and T_{init} is the initial temperature. We set ε so that it stops the cooling loop when the algorithm is not progressing. Indeed, when the temperature is low enough, the process accepts only better solutions and then it converges to the closest local optimum. This value can vary from one problem to another, and that is why it should be determined based on preliminary tests revealing the temperature level where optimality progress seems to stall.

5. Case Study

The data set (downloaded from the DDR2 Eurocontrol's database) corresponds to air traffic over Spanish airspace on 26th July 2016 between 12:00 AM and 4:00 PM (CEST - Central European Summer Time). Figure 4 shows the resulting 1060 flights together with the computed wind uncertainties according to the associated MétéoFrance PEARP EPS forecast.

The study is focused on a 3D case: We assume each aircraft flies its corresponding vertical profile,[5] reproducing the real layout of traffic. Note that the flight level structure already allocates Eastwards/Westwards flows in Odd/Even flight levels. In this case, conflicts are detected if there is a loss of horizontal separation (under uncertainty) or a loss of vertical separation.

The proposed SA algorithm is implemented in Python and simulated on an intel Core i5 2.4 GHz processor with 8 GB RAM. All CPU times refer to this infrastructure. We initially choose the following SA parameters:

- Number of transitions: $N_{trans} = 200$
- Geometric law coefficient $\alpha = 0.96$
- Stopping criterion coefficient $\varepsilon = 10^{-4}$

They result as a trade off between achieved objetive values and CPU times (around a half hour). Recall that the goal is to find a sufficiently good solution in a relatively short computing time. With the aim of studying the computational performance of the algorithm, the simulation experiments have been also solved considering SA settings $\alpha = 0.98$ and $N_{trans} = 400$. These results are presented, discussed, and compared in Section 6.0.1.

6. Results

Results are presented in Table 1. In this table, $\tilde{c}$ represents the virtual conflict number. This count is the value of function Φ in Equation (27), i.e.:

$$\Phi(\Delta V_a^f) = \underbrace{\sum_{v \in \mathcal{V}} \phi_{\mathcal{V}}(v)}_{\text{Vertex conflicts}} + \underbrace{\sum_{l \in \mathcal{L}} \phi_{\mathcal{L}}(l)}_{\text{Link conflicts}}$$

It is herein coined "virtual" because this function is evaluated by aircraft pairs $(a, b) \in \mathcal{F}$ at every link $l \in \mathcal{L}$ and vertex $v \in \mathcal{V}$ and at every time t. This count might (depending on its relative speeds) count twice the conflict (a with b and b with a), yet also count the same aircraft pair conflict in two or more sufficeintly close instants of time. It is thus overestimating the conflict count compared to conventional methods. Note, however, that $\tilde{c}$ is used in the algorithm to weight the importance of some particular conflicts, resulting on

[5] note, however, that a constant true airspeed profile is assumed in ascent/descent phases. This is for the sake of simplicity in the model, even though it is unrealistic.

	Without horizontal unc.	With horizontal unc.
ine $\tilde{c}$ (before resolution)	437	1181
ine $\tilde{c}^*$ (after resolution)	34	140
ine $\tilde{p}$	**92.2%**	**88.1%**
ine ine c (before resolution)	149	327
ine c^* (after resolution)	22	76
ine p	**85.2%**	**76.7%**
ine ine ine Computing time ine	1408 s	1498 s

Table 1: 3D results with SA settings $\alpha = 0.96$ and $N_{trans} = 200$.

some aircraft being more regulated than others. On the contrary, c represents the number of aircraft pairs (excluding the above explained *double* count and its multiple count over sufficiently close time instants) involved in a conflict, i.e., it can be named the "real" conflict count. Note the reader that the virtual conflict number provides an idea of algorithm performance, whereas the real number of conflicts provides a more operational and realistic perspective.

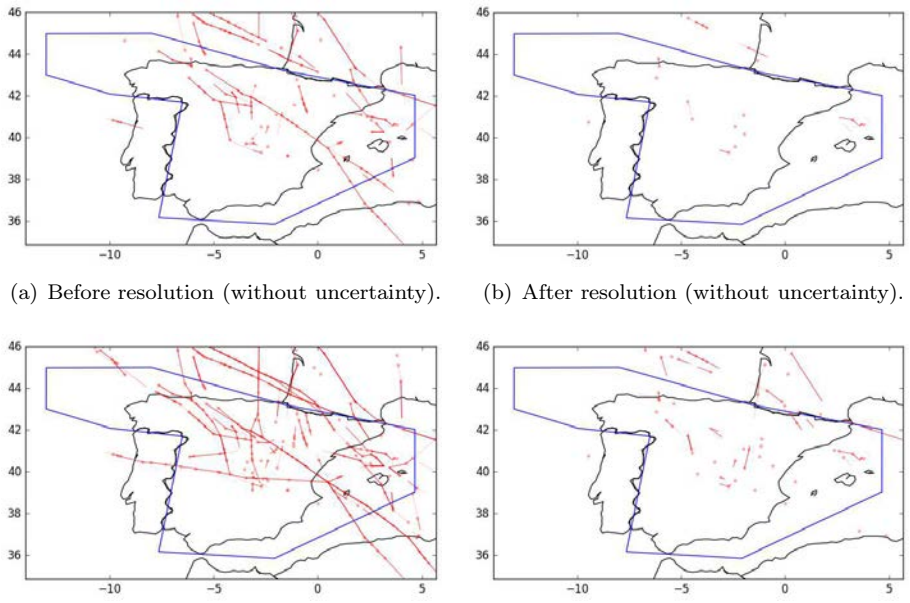
Both counts are presented before (attending to the flight schedule, i.e., $\Delta V_a^f = 0$) and after the resolution (represented by the symbol $(\cdot)^*$, i.e., the resulting count after the optimization in which $\Delta V_a^f \neq 0$ for some flights $f \in \mathcal{F}$). Finally, the variable p represents to the percentage of resolved conflicts in both counts.

Looking thus at Table 1 results, It can be readily noticed that both virtual and real conflict resolution rates are substantially improved both with and without uncertainty. The real number of conflicts considering uncertainty is reduced from 327 to 76 conflicts (76.7% reduction), at a comparatively lower computational time, i.e., less than 30 minutes. Looking at the virtual number of conflicts (the real indicator of algorithmic performance), conflict resolution rate grows to almost 88.1%. One can also compare the effects of both wind and temperature uncertainty in the simulation by comparing results with and without uncertainties, e.g., the real number of conflicts more than doubles (from 149 to 327) when incorporating uncertainty and the resolution rate degrades roughly 10 points. Figure 5 presents a qualitative map representing conflicts before and after resolution (with and without uncertainty).

The algorithm resolves virtual conflicts by 75-80%. When considering the real number of conflicts, resolution performance degrades to roughly 50% resolution success. In this case, note that the effects of uncertainty are impactful, as the resolution rate would be 15-20 points higher without the consideration of uncertainty.

6.0.1. Numerical performances

To illustrate the effects of the annealing parameters in numerical performances, we present simulation results with different SA parameter settings. The idea is to make the algorithm explore a larger set of the solution space at



(a) Before resolution (without uncertainty). (b) After resolution (without uncertainty).

(c) Before resolution (with uncertainty). (d) After resolution (with uncertainty).

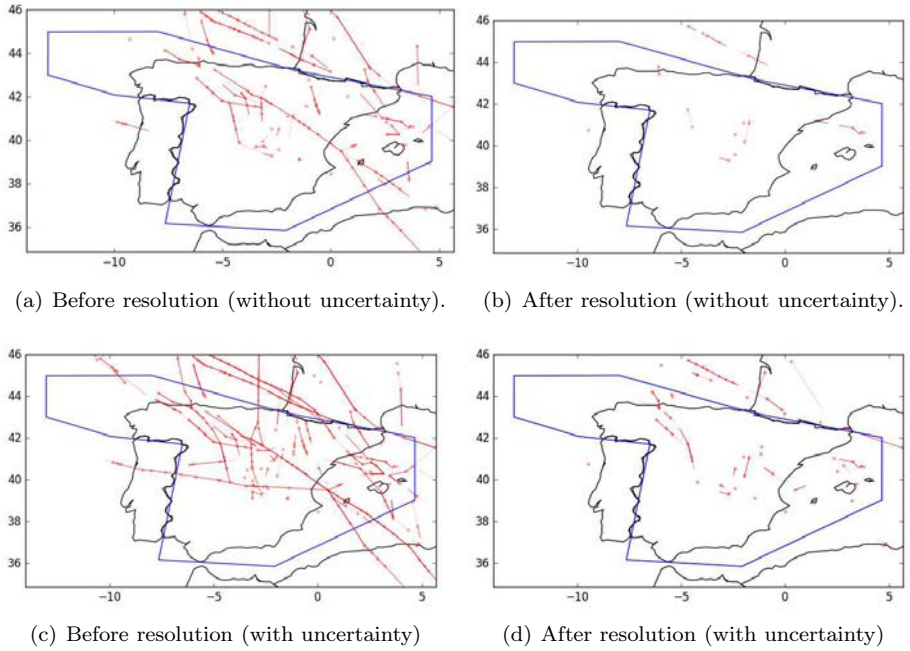
Figure 5: Visualisation of conflicts (solid-red lines corresponding to link conflicts; red dots corresponding to vertex conflicts) for 3D traffic before and after resolution and SA settings $\alpha = 0.96$ and $N_{trans} = 200$.

	Without horizontal unc.	With horizontal unc.
$\tilde{c}$ (before resolution)	437	1181
$\tilde{c}^*$ (after resolution)	32	121
$\tilde{p}$	**92.7%**	**90%**
c (before resolution)	149	327
c^* (after resolution)	19	76
p	**87.2%**	**76.7%**
CPU time	5682 s	5714 s

Table 2: 3D Results for SA settings $\alpha = 0.98$ and $N_{trans} = 400$.

the expense of increased computing times. We select a coefficient α equal to 0.98 (slower temperature decrease) and a number of transitions equal to 400 (the algorithm evaluates twice the number of states ΔV_a^f between two temperature changes). Table .6 shows the simulation results for the 2D case. Improvements in resolution rate are in the order of 2-5% at the cost of four times more CPU usage (roughly 2 hours against 30 min).

Table 2 shows the simulation results for the 3D case. Improvements range from 5% to 10% in the conflict resolution rates at the cost of four times more



(a) Before resolution (without uncertainty).

(b) After resolution (without uncertainty).

(c) Before resolution (with uncertainty)

(d) After resolution (with uncertainty)

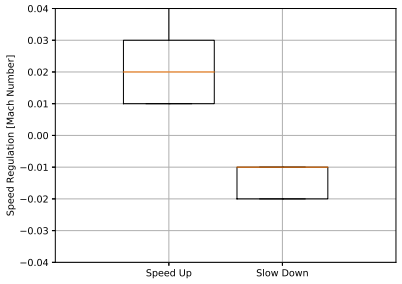
Figure 6: Conflict visualisation (solid-red lines corresponding to link conflicts; red dots corresponding to vertex conflicts) for 3D traffic before and after resolution with SA settings $\alpha = 0.98$ and $N_{trans} = 400$: with uncertainties (bottom) and without uncertainties (top).

CPU time (roughly 1.5 hours against 25 min.). Figure 6 presents a qualitative map representation of conflicts in the 3D case before and after resolution with SA settings $\alpha = 0.98$ and $N_{trans} = 400$. All in all, in the 3D case considering uncertainty, potential conflicts would be reduced from 289 to 50.

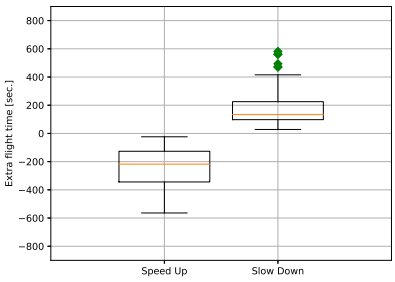
6.0.2. Impact on efficiency

The reduction in the number of conflicts via speed regulations comes at the cost of modifications of the scheduled flight times and nominal fuel consumption. These are parameters set according to airline's policies via Cost Index, which are assumed to be the ones preferred by the airline. Thus, any modification (in the sense of speeding up/slowing down the aircraft) would have a negative impact in airline's economics. With the aim at characterizing this impact, we group advisories into two groups, namely: speed-up and slow-down flights. Figure 7 presents disaggregated information on speed regulations for the 3D case with uncertainty and SA settings $\alpha = 0.98$ and $N_{trans} = 400$. Medians values of true airspeed are of 6.00 kt. and -7.00 kt. for the speed up and slow down groups, respectively. All in all, roughly 500 aircraft (out of an scenario of 1060 flights) have been regulated. Figure 7 also presents the speed regulation effects in flight

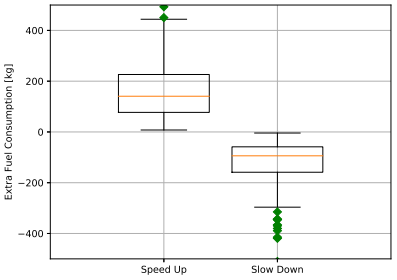
time and fuel consumptions[6]. Detailed information can be checked in Table (3).



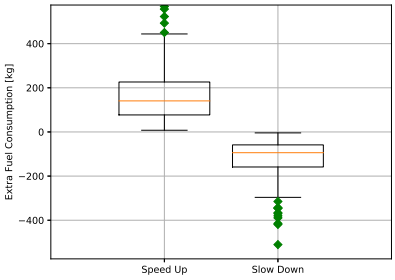
(a) Speed regulations (Δv_a^f)



(b) Extra flight time



(c) Extra Fuel Consumption (zoom)



(d) Extra Fuel Consumption (complete)

Figure 7: Impact on efficiency (box plots): 3D case with uncertainty; SA settings $\alpha = 0.98$ and $N_{trans} = 400$.

	Speed up	**Slow down**
number of aircraft regulated	262	156
Average speed variation (M)	0.02	-0.013
Min/Max speed variation (kt)	$[0.01, 0.04]$	$[-0.01, -0.02]$
Average time ahead/behind schedule (s)	-246	$+170.9$
Min/Max time ahead/behind schedule (s)	$[-564, -23, 8]$	$[28.1; 581.5]$
Average consumption variation (kg)	172	-144
Min/Max extra fuel consumption (kg)	$[0; 987]$	$[0; -1569]$

Table 3: Impact on flight time and fuel burnt: case with uncertainty; SA settings $\alpha = 0.98$ and $N_{trans} = 400$.

[6]Fuel consumption calculations have been computed following BADA 3 formulas

7. Conclusion

We proposed a formulation for strategic deconfliction based on speed regulations, where conflicts should be reduced or ideally avoided without any spatial change in aircraft trajectories. Wind uncertainties are included in the modeling. All in all, in the scenario used as application, we can we are able to solve around 80-90% of conflicts only by regulating speeds before departure. This would be at the toll of modifying flight times. The associated fuel cost can be consider neglectable (-1.47 kg per flight). Thus, it is shown that this strategic conflict resolution approach could a priori relax controllers' workload and indirectly increase capacity. One of the main drawbacks is a rather high CPU time of one/two hours for only four hours of traffic over Spain, which is not acceptable. This can be partially attributed to Python modelling language. CPU times improvements are demanded and a clear direction of research, e.g., to implement it in JAVA. Other natural extension could be: the implementation of other separation maneuver types such as Heading or FL changes; the resolution via ground delays; or the consideration of other sources of uncertainty such us aircraft dynamics or departure times.

	Without Unc.	With Unc.
ine ine $\tilde{c}$ (before resolution)	1407	2496
ine $\tilde{c}^*$ (after resolution)	300	604
ine $\tilde{p}$	**78.7%**	**75.8%**
ine ine c (before resolution)	312	427
ine c^* (after resolution)	116	224
ine p	**62.8%**	**47.5%**
ine ine Computing time ine	1458 s	1493 s

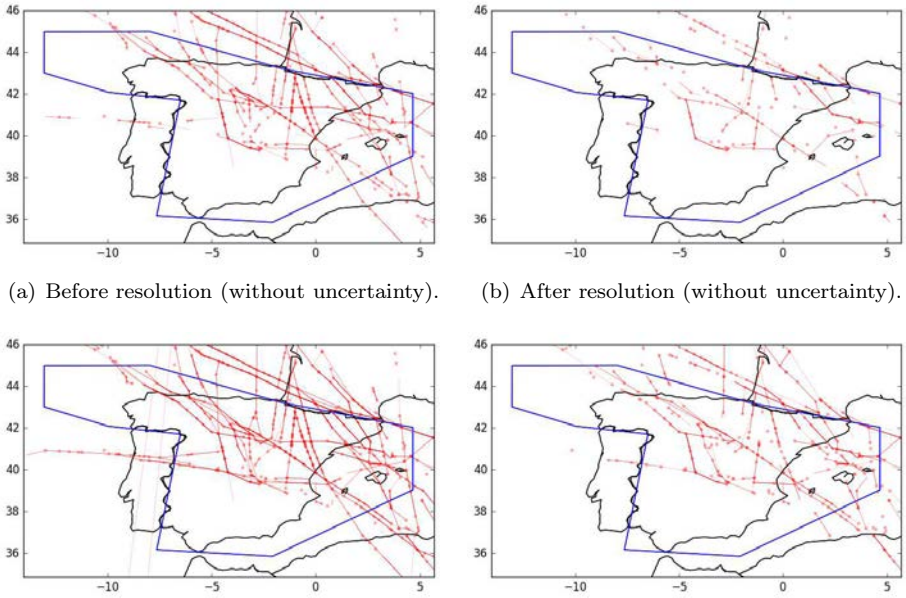
Table .4: Results for the flights flying West without/with uncertainties: 523 flights. SA settings $\alpha = 0.96$ and $N_{trans} = 200$.

	Without Unc.	With Unc.
ine $\tilde{c}$ (before resolution)	1239	2405
ine $\tilde{c}^*$ (after resolution)	211	469
ine $\tilde{p}$	**83.0%**	**80.5%**
ine ine c (before resolution)	289	457
ine c^* (ater resolution)	81	198
ine p	**72.0%**	**56.7%**
ine ine Computing time ine	1816 s	1960 s

Table .5: Results for flights flying East without/with uncertainties: 537 flights. SA settings $\alpha = 0.96$ and $N_{trans} = 200$.

Direction	East	West
ine $\tilde{c}$ (before resolution)	2405	2496
ine $\tilde{c}^*$ (after resolution)	432	507
ine $\tilde{p}$	**82.0%**	**79.7%**
ine ine c (before resolution)	457	427
ine c^* (after resolution)	182	199
ine p	**60.2%**	**53.4%**
ine ine CPU time ine	7233 s	6615 s

Table .6: 2D Results (with uncertainties) for SA settings $\alpha = 0.98$ and $N_{trans} = 400$.



(a) Before resolution (without uncertainty). (b) After resolution (without uncertainty).

(c) Before resolution (with uncertainty). (d) After resolution (with uncertainty).

Figure .8: Visualisation of the 2D case conflicts (solid-red lines corresponding to link conflicts; red dots corresponding to vertex conflicts) for East and West traffic altogether before and after resolution with SA settings $\alpha = 0.96$ and $N_{trans} = 200$.

References

Alonso-Ayuso, A., Escudero, L.F., Martín-Campo, F.J., 2011. Collision avoidance in air traffic management: A mixed-integer linear optimization approach. IEEE Transactions on Intelligent Transportation Systems 12, 47–57.

Bicchi, A., Pallottino, L., 2000. On optimal cooperative conflict resolution for air traffic management systems. IEEE Transactions on Intelligent Transportation Systems 1, 221–231.

Cafieri, S., Durand, N., 2014. Aircraft deconfliction with speed regulation : new models from mixed-integer optimization. Journal of Global Optimization 58, 613–629.

Chaimatanan, S., Delahaye, D., Mongeau, M., 2014. A hybrid metaheuristic optimization algorithm for strategic planning of 4d aircraft trajectories at the continental scale. IEEE Computational Intelligence Magazine 9, 46–61.

Courchelle, V., Delahaye, D., González-Arribas, D., Soler, M., 2017. Simulated annealing for strategic traffic deconfliction by subliminal speed control under wind uncertainties, in: Proceeding of the 7th SESAR Innovation Days.

Durand, N., 1996. Optimisation de trajectoires pour la résolution de conflits aériens en route. Ph.D. thesis. Institut National Polytechnique de Toulouse.

Durand, N., Alliot, J.M., 2009. Ant colony optimization for air traffic conflict resolution, in: in Proceedings of 8th ATM Seminar 2009.

Durand, N., Alliot, J.M., Noailles, J., 1996. Automatic aircraft conflict resolution using genetic algorithms. In Proceedings of the Symposium on Applied Computing, Philadelphia .

Durand, N., Gotteland, J., 2003. Algorithmes génétiques appliqués à la gestion du trafic aérien. Journal sur l'enseignement des sciences et technologies de l'information et des systèmes. 2, 1–6.

Hernández, E., Valenzuela, A., Rivas, D., 2017. Probabilistic aircraft conflict detection and resolution considering wind uncertainty, in: Proceeding of the 7th SESAR Innovation Days.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.

Lecchini, A., Glover, W., Lygeros, J., Maciejowski, J., 2006. Monte carlo optimization for conflict resolution in air traffic control. IEEE Transactions on Intelligent Transportation Systems 7, 470–482.

Liu, W., Hwang, I., 2011. Probabilistic trajectory prediction and conflict detection for air traffic control. Journal of Guidance, Control, and Dynamics 34, 1779–1789.

Liu, W., Hwang, I., 2014. Probabilistic aircraft midair conflict resolution using stochastic optimal control. IEEE Transactions on Intelligent Transportation Systems 15, 37–46.

Maria Prandini, Jianghai Hu, J.L., Sastry, S., 2000. A probabilistic approach to aircraft conflict detection. IEEE Transactions on Intelligent Transportation Systems 1, 199–220.

Martín-Campo, F.J., 2010. The collision avoidance problem: Methods and algorithms. Ph.D. thesis. Universidad Rey Juan Carlos.

Matsuno, Y., Tsuchiya, T., Wei, J., Hwang, I., Matayoshi, N., 2015. Stochastic optimal control for aircraft conflict resolution under wind uncertainty. Aerospace Science and Technology 43, 77–88.

Pallottino, L., Feron, E.M., Bicchi, A., 2002. Conflict resolution problems for air traffic management systems solved with mixed integer programming. IEEE Transactions on Intelligent Transportation Systems 3, 3–11.

Rey, D., Constans, S., Fondacci, R., Rapine, C., 2010. A mixed integer linear model for potential conflict minimization by speed modulations., in: Proceeding of the 4th International Conference on Research in Air Transportation, pp. 513–515.

Rodionova, O., Sridhar, B., Ng, H.K., 2016. Conflict resolution for wind-optimal aircraft trajectories in north atlantic oceanic airspace with wind uncertainties, in: Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th, IEEE. pp. 1–10.

Ruiz, S., Piera, M.A., Nosedal, J., Ranieri, A., 2014. Strategic de-confliction in the presence of a large number of 4d trajectories using a causal modeling approach. Transportation Research Part C: Emerging Technologies 39, 129–147.

Vela, A., Solak, S., Singhose, W., Clarke, J.P., 2009. A mixed integer program for flight-level assignment and speed control for conflict resolution, in: IEEE (Ed.), Proceedings of the 48th IEEE Conference on Decision and Control, pp. 5219–5226.