Simulated Annealing for Strategic Traffic Deconfliction by Subliminal Speed Control under Wind Uncertainties

Valentin Courchelle, Daniel Delahaye
ENAC -Ecole Nationale de l'Aviation Civile-
Toulouse, France
Email: valentin.courchelle@gmail.com, delahaye@recherche.enac.fr

Daniel González-Arribas, Manuel Soler
Universidad Carlos III de Madrid
Leganés, Spain
Email: dangonza, masolera@ing.uc3m.es

***Abstract*—This paper introduces an algorithm that minimises conflicts between aircraft at the strategic level taking into account uncertainties on aircraft position due to errors into wind forecast. The strategy relies on subliminal speed control. 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. Traffic has been retrieved from NEST Eurocontrol database with the corresponding wind ensemble probabilistic forecasts from the European Centre for Medium-Range Weather Forecasts. Due to uncertainties and to the little range of speed changing allowed by a subliminal control, it becomes not possible to resolve all conflicts. However, their number can be significantly reduced by slightly modifying flight plan speeds while not touching the selected route by the airspace user.**

***Keywords*—ATM, Conflicts, Speed control, Wind uncertainties, Simulated annealing**

I. Introduction

An increase in global air traffic is foreseen in the coming decades. According to the International Civil Aviation Organization (ICAO), it will double by 2030 to reach 6 billion passengers. In order to increase airspace capacity and therefore avoid their saturation, the air traffic management (ATM) system needs to be improved. The development of advanced algorithms and tools capable of anticipating conflict detection and resolution are necessary as this would lighten future air traffic controller (ATC) workload. To this end, coping with uncertainty is absolutely paramount. The development of computer aided conflict resolution tools of this type is aligned with the goals and technological solutions within the future ATM system in Europe, built under the umbrella of the Single European Sky ATM Research (SESAR). The strategic (in this context, before departure) conflict resolution strategy seeks to deviate as little as possible aircraft from the original aircraft flight plan, minimising the impact of the separation maneuvers on the flight efficiency.

A large number of strategies have been proposed for so-called conflict detection and resolution problems; refer for instance to the non exhaustive review is provided in [1] or more recently [2]. According to its time horizon, conflict detection & resolution algorithm can be classified into tactical (real time algorithms within a sector) and strategic (planning level algorithm within a network).

For the former, the typical approach is to consider different separation maneuver, e.g., velocity changes [3], [4], heading changes [3], [5], or even combined actions on velocity and Flight Level (FL) changes [6]. Each of these used mixed integer optimisation models. Metaheuristics can be also effective, e.g., using ant colony [7] or genetic algorithm [8], both including heading changes. Aiming to provide robustness against uncertainties, some previous work has also considered different probabilistic approaches to the conflict detection and resolution problem at the tactical level, e.g., [9] (using Monte Carlo), [10] (Markov Chain) or other tools [11]. More recently, a two aircraft encounter was solved using wind uncertainties extracted from ensemble probabilisitic forecasts [12].

Nevertheless, aircraft conflict resolution is a highly combinatorial problem that cannot be solved using classical optimisation techniques and realistic models when the number of aircraft becomes significant. This is the case when the problem is tackled at the strategic level, which imposes to consider a macro-scaled airspace and deal with thousands flights. In this so-called strategic deconfliction context, previous work includes for instance [13] (with FL assignment and speed control) or [14] (with real air traffic on a day in the European airspace and conflicts solved by heading changes). However, uncertainties are not taken into account, and they greatly affect traffic and thus potential conflicts. Other studies proposed models that consider uncertainties in the European airspace [15] or in the North Atlantic oceanic airspace [16]. Both resolved conflicts following a ground delay strategy and the modification of trajectory's geometrical shape, showing that it is possible to increase airspace capacity under uncertainties. Nevertheless, and to the best of author's understanding, the deconfliction using speed control on a macroscaled traffic under uncertainties is an unexplored field.

ERASMUS is a related Eurocontrol funded project to study methods and technologies to increase levels of automation in ATM, in particular air traffic control. An important finding of ERASMUS is the so-coined 'subliminal control'. In this

approach, with minor speed control, significant portions of traffic could de-conflicted while ATCOs workload is reduced (since those minor speed modifications are not perceived by ATCOs). Publications related to the ERASMUS project and subliminal control include for instance [17], [18], [19].

In this paper we propose a strategic de-confliction method through subliminal speed regulations. Wind uncertainties are considered to be the unique source of uncertainty. Thus uncertainties on aircraft positions are taken into account, deduced from a real wind ensemble probabilistic forecasts. An application to real traffic into an structured, continental airspace is shown as case study. The number of conflicts is minimized by small speed deviations from that in the flight plan, while leaving the flight plan's route untouched. Given the large number of planes that can transit into a given airspace, we resorted to a resolution by a metaheuristic approach using simulated annealing.

The next sections of this document are organised as follows: Section II elaborates on probabilistic wind forecast and associated uncertainties. Section III introduces the mathematical modelling. Section IV describes the simulated annealing algorithm to solve the problem. Section V presents the numerical results. Finally, some conclusions and future directions of research are drawn in Section VI.

mds
January 11, 2007

II. Wind Uncertainty

Uncertainty of wind fields and convective regions will be derived 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. 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 others) at the TIGGE dataset by the European Center for Medium-Range Weather Forecasts (ECMWF).

A. MétéoFrance PEARP EPS

The MétéoFrance PEARP (Prévision d'Ensemble ARPège) is the probabilistic form of the MétéoFrance global numerical weather prediction model ARPEGE. The EPS probabilistic forecast has been based on 35 integrations with approximately 10-km resolution in France (60-km at the antipodes) performing forecasts up to 4.5 days with 90 vertical levels.

The MétéoFrance PEARP EPS 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). Each forecast is based on a model which is close, but not identical, to our best estimate of the model equations, thus representing also the influence of model uncertainties on forecast error.

The divergence, or spread, of the control plus 34 forecasts -35 in total- gives an estimate of the uncertainty of the prediction on that particular day. On some days, the spread might be small implying that the atmosphere is very predictable and users can trust that the reality will fall somewhere in the narrow range of forecasts. On other days or in other areas, the 35 forecasts might diverge considerably after just a few forecast days, indicating that the atmosphere is especially unpredictable. The variable ensemble spread gives users potentially very useful information on the range of uncertainty. Having a quantitative flow-dependent estimate of uncertainty allows users to make better informed weather-related decisions.

III. Mathematical Modelling

We assume all aircraft to fly at the same flight level. This hypothesis simplifies the implementation of the modelling because the algorithm deals only with two spatial dimensions. Notice however that it overemphasises the number of conflicts. Also, aircraft flying Eastwards are separated from those flying Westwards. Otherwise, face to face conflicts would not be solvable only by speed regulations. This is at the cost of running twice the algorithm: one time on flights Eastwards and the other on flights Westwards.

In the modelling, aircraft are assumed to fly a constant True Air Speed (TAS) profile, to be assigned by the algorithm within the sublimal speed control bounds. However, notice that the motion of the aircraft with respect to Earth is governed by its ground speed, which depends on its TAS and the existing wind according to the following function:

$$v_g = v_a + v_w . \cos(\chi_w - \chi) \tag{1}$$

with v_g the ground speed, v_a the TAS, v_w the wind speed, χ_w and χ the wind and aircraft tracks, respectively. Thus, wind uncertainties heavily affect aircraft positions.

A. Uncertainties modelling

In order to insert uncertainties into the model, we use the MétéoFrance PEARP ensemble forecast E downloaded from TIGGE dataset. For each aircraft a and each members e, knowing the departure time and the flight plan, we compute the arrival time T_e^a. As Figure 1 illustrates, from this set of possible arrival times, different metrics can be obtained, e.g., the mean time and the range of times. Consequently, for each aircraft $a \in A$ we associate a maximum error on arrival time as follows:

$$\Delta T^a = \max(\delta T^a_{min}, \delta T^a_{max}) \tag{2}$$

where

$$\delta T^a_{max} = \max_{e \in E} T^a_e - \sum_{e \in E} \frac{T^a_e}{34}$$

$$\delta T^a_{min} = \sum_{e \in E} \frac{T^a_e}{34} - \min_{e \in E} T^a_e$$

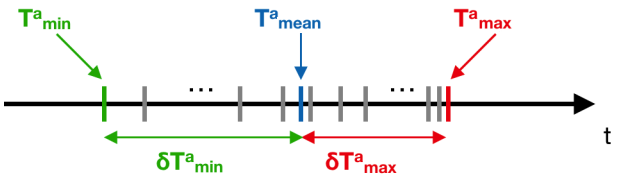

Figure 1. Illustration of aircraft's a time arrival error.

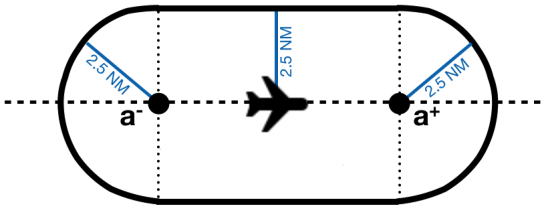

Figure 2. Illustration of the protected area around the aircraft a at time t. If two areas is overlapping, there is a potential conflict.

With the maximum error on arrival time, we can extend the protected area around an aircraft over time by considering an additional margin on the separation norm as illustrated in Figure 2. For each aircraft $a \in A$, two fictive positions a^+ (in front of a) and a^- (behind a) delimit the segment of possible positions of the aircraft a regarding uncertainties. If we call T_0^a the departure time and T^a the arrival time, for each time of the flight we can compute a protecting time gap as follows:

$$\forall a \in A, \forall t \in [T_0^a, T^a], \Delta T^a(t) = \Delta T^a \times \frac{t - T_0^a}{T^a - T_0^a} \quad (3)$$

Then, a^+ is the future position of a at time $t + \Delta T^a(t)$ and a^- is the previous position of a at time $t - \Delta T^a(t)$. Note that for each aircraft a the margin is zero at T_0^a and grows over time to reach its maximum at T^a.

B. Conflict evaluation

Two types of conflicts can be distinguished. The first typology occurs at the intersection (called node) of two different routes and it will be coined node conflict. The second typology occurs when two aircraft are in the same portion of route between two nodes (called link) and when one of the two aircraft is catching up the other. This type of conflict will be coined link conflict.

Link conflict:

A link conflict can occur at the entry of a link l (at the first node) and at its exit (at the second node). Let us consider two aircraft a and b flying on link l such that a is ahead of b. Let $v_g^a(l)$ and $v_g^b(l)$ be the ground speeds of aircraft a and b on link l, respectively. Two time intervals must be considered:

For the entry link, the first time interval is $[t_{in,l}^{a^-}, t_{in,l}^{a^-} + \frac{S_0}{v_g^a(l)}]$ between the time a^- is at the entry and the time it is at $S_0 = 5NM$ after the entry. The second time interval $[t_{in,l}^{b^+}, t_{in,l}^{b^+} + \frac{S_0}{v_g^b(l)}]$ is the equivalent for b^+. If these two intervals overlap, which means that $t_{in,l}^{b^+} - (t_{in,l}^{a^-} + \frac{S_0}{v_g^a(l)}) < 0$ there is a conflict. See Figure 3.

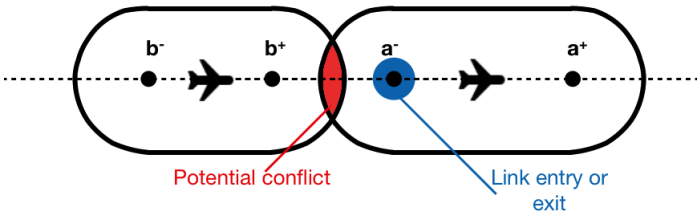

Figure 3. Illustration of a conflict at the entry/exit of a link.

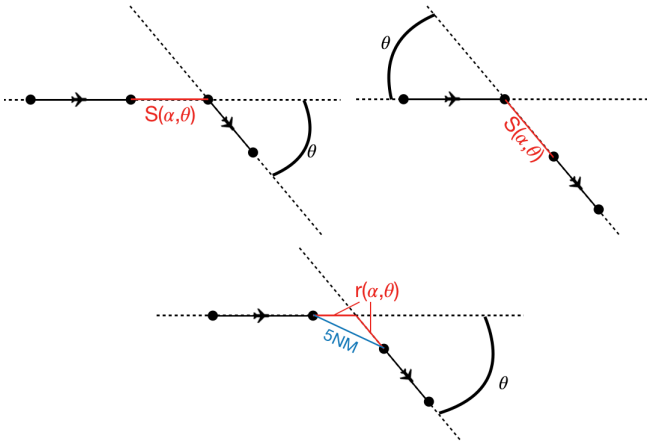

Figure 4. Three configurations to detect a conflict at a node.

For the exit link, the same reasoning holds but replacing in by out. To evaluate all conflicts which occur on link l let us define the following function:

$$\forall l \in \mathcal{L},\ \phi_{\mathcal{L}}(l) = -\sum_{(a,b)\in A_l^{in}} t_{in,l}^{b^+} - (t_{in,l}^{a^-} + \frac{S_0}{v_g^a(l)}) - \sum_{(a,b)\in A_l^{out}} t_{out,l}^{b^+} - (t_{out,l}^{a^-} + \frac{S_0}{v_g^a(l)}) \quad (4)$$

where $\mathcal{L}$ is the set of links and A_l^{in} is the set of aircraft pairs (a, b) involved into a conflict at the entry link l, giving that a flies ahead of b. A_l^{out} is the same for the link exit. By construction this function is positive.

Node conflict:

The detection of a node conflict relies on the same principle. Three different cases need to be modelled to cover all configurations. These are illustrated in Figure 4.

For the first two configurations -upper sketches in Figure 4- aircraft a would be slightly behind or aircraft b would be slightly ahead. Around node n, both a and b don't follow the same link so their tracks are different. This difference, denoted $\theta_{a,b}^n$, has an impact on the required distance between

a^- and b^+, which goes accordingly with their difference in velocities. The required distance is then not 5 NM anymore but $S(\alpha_{a,b}, \theta^n_{a,b})$, where $\alpha_{a,b}$ is the ratio between the ground speeds (a over b). It can be proofed that [20]:

$$S(\alpha_{a,b}, \theta^n_{a,b}) = S_0 \times \frac{\sqrt{\alpha^2_{a,b} - 2.\alpha_{a,b}.\cos(\theta^n_{a,b}) + 1}}{|\sin(\theta^n_{a,b})|}$$

As for the third configuration, the distance $r(\alpha_{a,b}, \theta^n_{a,b})$ is chosen in order to have 5NM between a^- and b^+ when they are both at $r(\alpha_{a,b}, \theta^n_{a,b})$ from the node. Then this distance has to be:

$$r(\alpha_{a,b}, \theta^n_{a,b}) = S_0 \times \frac{\sqrt{\alpha^2_{a,b} - 2.\alpha_{a,b}.\cos(\theta^n_{a,b}) + 1}}{2.\cos(\frac{\theta^n_{a,b}}{2})}$$

As it is done for a link conflict, we focus, for each way of detection, on the overlapping of the specific intervals. If one of these detection ways reveals an overlap, aircraft *a* and *b* are in conflict. To evaluate all conflicts which occur on a node n we define the following function:

$$\begin{aligned}
&\forall n \in \mathcal{N},\\
&\phi_{\mathcal{N}}(n) = -\sum_{(a,b)\in A^1_n} (t^{b^+}_n - \frac{S(\alpha_{a,b}, \theta^n_{a,b})}{v^b_g(n)}) - t^{a^-}_n\\
&\quad - \sum_{(a,b)\in A^2_n} t^{b^+}_n - (t^{a^-}_n + \frac{S(\alpha_{a,b}, \theta^n_{a,b})}{v^a_g(n)})\\
&- \sum_{(a,b)\in A^3_n} (t^{b^+}_n - \frac{r(\alpha_{a,b}, \theta^n_{a,b})}{v^b_g(n)}) - (t^{a^-}_n + \frac{r(\alpha_{a,b}, \theta^n_{a,b})}{v^a_g(n)}) \quad (5)
\end{aligned}$$

where $\mathcal{N}$ is the set of nodes and A^1_n is the set of pairs (a, b) of aircraft involved into conflicts in the first configuration at node n, where a reaches n before b. A^2_n reads the same but for conflicts in the second configuration. Finally A^3_n denotes the set conflicts detected in the third configuration. By construction this function is positive.

C. Mathematical modelling setting up

1) State space:

The state space is the set of vectors $X = (x_i)_{i=1..N} \in \mathbb{Z}^N$ with dimension *N* equal to the number of aircraft considered. Each component x_i of these vectors corresponds to a variation of TAS applied to the aircraft *i*. These velocity variations are integers, something operationally consistent -pilots might set autopilot speed with precision M0.01-. Moreover, they can be positive or negative because the pilot can be asked to accelerate or decelerate. Knowing TAS and wind, one can readily get the ground speed used into the conflict evaluation function.

2) Constraints:

Variable definition constraint:

$$\forall k \in [\![1, N]\!],\ x_k \in \mathbb{Z} \quad (6)$$

Subliminal control constraint:

As explained in Section I, subliminal control requires minor speed changes. A reasonable interval in which speed variations should be located could be -6% to +3% of the initial speed [4].

$$\forall i \in [\![1, N]\!],\ -0,06 \times v_i \leq x_i \leq 0,03 \times v_i \quad (7)$$

3) Objective:

The aim is to minimise conflicts with the least impact on aircraft performance. We define the function which evaluates conflicts corresponding to the current state X as follow:

$$\Phi(X) = \sum_{n\in\mathcal{N}} \phi_{\mathcal{N}}(n) + \sum_{l\in\mathcal{L}} \phi_{\mathcal{L}}(l) \quad (8)$$

So the objective function is:

$$min\ f = M \times N \times \Phi(X) + \sum_{i=1}^{N} |\ x_i\ | \quad (9)$$

where M is a coefficient used to weight the minimisation of conflicts w.r.t speed changes. The multiplication by the number of aircraft N plays an analogous role.

4) Problem Statement:

All in all, the problem is stated as follows:

Objective function:

$$min\ f = M \times N \times \Phi(X) + \sum_{i=1}^{N} |\ x_i\ |$$

Subject to:

$$\forall i \in [\![1, N]\!], \quad -0,06 \times v_i \leq x_i \leq 0,03 \times v_i$$

Where:

$$\begin{aligned}
\Phi(X) &= \sum_{n\in\mathcal{N}} \phi_{\mathcal{N}}(n) + \sum_{l\in\mathcal{L}} \phi_{\mathcal{L}}(l)\\
\forall l \in \mathcal{L},\ &\phi_{\mathcal{L}}(l) \leftarrow (4)\\
\forall n \in \mathcal{N},\ &\phi_{\mathcal{N}}(n) \leftarrow (5)\\
\forall k \in [\![1, N]\!],\ &x_k \in \mathbb{Z}
\end{aligned}$$

D. 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 [21] "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 [22] that the set of permissible 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 solution make the problem highly combinatorial. That is the reason behind conflict resolution problems being hard optimisation problems. Metaheuristic are possibly more suitable.

IV. Simulated annealing (SA)

A. General description of simulated annealing

SA is a metaheuristic inspired by the annealing process in metallurgy. It consists in bringing the system, from a disordered random state, to a global-minimum energy state, involving heating process and cooling process. A global parameter, temperature T is applied to control these two processes. The objective function is analogical to the internal energy of the physical problem. SA compares the neighbouring state to its current state and moves from one to another probabilistically. When T is high, deteriorated solutions (with high energy) are more likely to be accepted. When T decreases, better solutions are found. At last, a state considered to be good enough is reached. SA is well known for its ability to trap out of the local minimum by allowing random neighbourhood changes. Moreover, it can be easily adapted to various kinds of problems with continue or discrete space states.

B. Adaptation of SA for our problem

In order to adapt the SA algorithm to our problem, several parameters and functions need to be considered.

1) Neighbourhood function:

A *neighbourhood function* is used to generate a local change from the current solution. Two criteria should be considered: the computational time should be low and the change should remain local, so as to avoid this change to resemble to a pure random search. The neighbourhood generation function is described in seudo code 1.

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

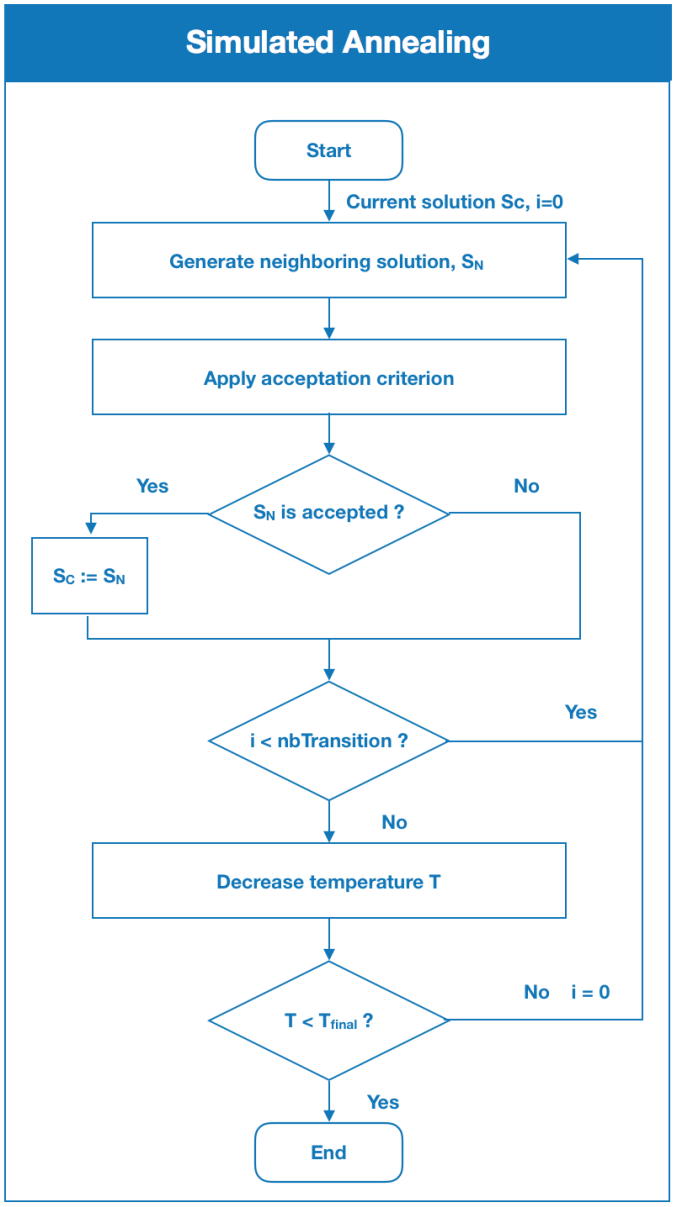

Figure 5. Simulated annealing algorithm.

2) Initial temperature and acceptance probabilities:

The temperature parameter, $T(k)$ -at iteration k of the SA-, is used to control the acceptance of a solution's degradation. If at step k, $T(k)$ is high, then all the neighbourhoods have almost the same probability to be accepted and large degradation are more likely to be produced. To the limit, when $T(k)$ approaches infinity, all neighbours are systematically accepted. On the contrary, if $T(k)$ is low, a movement that degrades the solution is unlikely to be kept. 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, we evaluate a temperature which can bring an acceptance rate of 80%. This evaluating method is described by the *HeatUpLoop* procedure of Algorithm 2 .

3) Cooling loop:

Among the different methods to decrease the temperature, we decide to use the geometric law which is a classical method for the SA.

$$T_{i+1} = T_i \times \alpha, \qquad 0 < \alpha < 1$$

At each iteration, we get the new temperature via multiplying by a predefined coefficient α. The choice of α is delicate because if α is too large, the temperature decreases very slowly and the convergence toward the optimum is likely to be too long. However, if α is chosen too small, the temperature

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

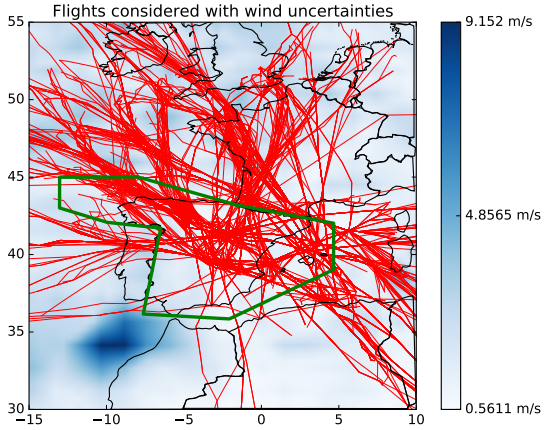

Figure 6. Visualisation of the traffic (red) considered in Spanish airspace (green) under wind uncertainties -blues-.

decreases fast and the algorithm risks to be quickly blocked at a local optimum. That's why this parameter has to be adapted to a problem. The precise cooling process is described by the CoolingLoop procedure of Algorithm 2.

4) Stopping criterion:

The termination criterion is set to be the final temperature reaching value $T_{\text{init}} \times \varepsilon$, where ε is a predefined coefficient, and T_{init} is the initial temperature for cooling process. We set ε based on tests.

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

V. RESULTS

The proposed algorithm is implemented in Python and simulated on an intel Core i5 2.4 GHz processor with 8 GB RAM. The data set (downloaded from the NEST Eurocontrol database) corresponds to air-traffic over Spanish airspace on 26th July 2016 between 12. am and 4. pm. Figure 6 shows the resulting 1060 flights together with the computed wind uncertainties according to the associated MétéoFrance PEARP EPS forecast.

The simulated annealing parameters used are:

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

These parameters result as a trade off between the objetive

vaue and computing time (around a half hour). Indeed the aim of a metaheuristic approach isn't to find the optimal solution but a satisfying one in a short computing time.

Results are presented in tables I and II.

TABLE I
RESULTS FOR THE FLIGHTS TO WEST WITH AND WITHOUT UNCERTAINTIES: 523 FLIGHTS

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

TABLE II
RESULTS FOR THE FLIGHTS TO EAST WITH AND WITHOUT UNCERTAINTIES: 537 FLIGHTS

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

In the tables, $\tilde{c}$ represents the virtual conflict number, which is used by the algorithm to put more weight to a certain aircraft regulation than another. It is a "virtual" count because a conflict can be counted several times if it occurs on several nodes or links. At the opposite, c represents the number of aircraft pairs involved into a conflict so the "real" conflict count. Both counts are computed for initial flight schedules and after the resolution (represented by the symbol*). Finally, the parameter p corresponds to the percentage of resolved conflicts.

The virtual conflict number gives a clear idea of the algorithm performances. In fact we can note that the algorithm reduces this number at least by 70% (perceptible in Figure 8), but it never solves all conflicts because of the short maneuver range in speed change that a subliminal control allows.

To illustrate the effects of the annealing parameters we did simulations with other settings making the algorithm exploring a larger part of the solution space but requiring a longer computing time (around two hours). Then with a coefficient α equal to 0.98, the temperature decrease is slower. Moreover with a number of transitions equal to 400, the algorithm evaluate twice more states X between two temperature changes

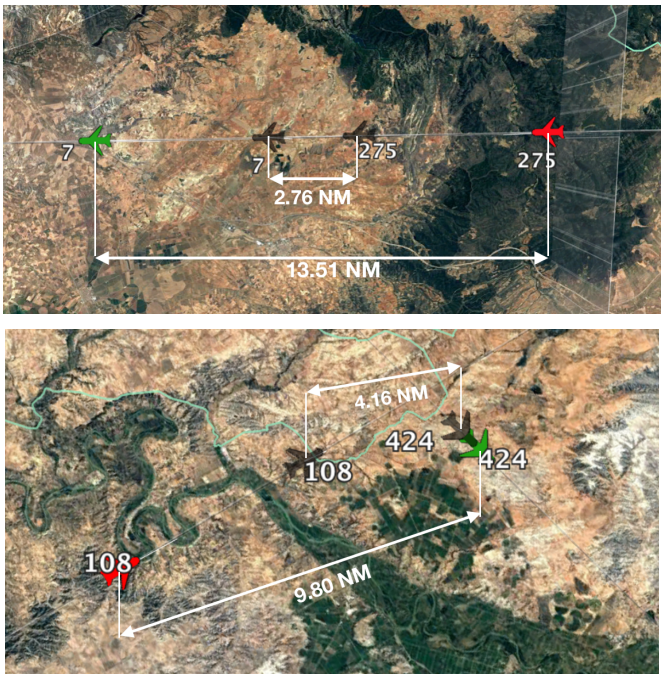

Figure 7. Visualisation of a link conflict resolution (top) and a node conflict one (bottom): an aircraft in green is accelerated and in red it is decelerated. Aircraft in black represent their positions without resolution. Capture from *KML* file readable by *Google Earth*

TABLE III
RESULTS FOR A LONGER SIMULATION AND FOR EACH DIRECTION UNDER UNCERTAINTIES

Direction	East	West
$\tilde{c}$	2405	2496
$\tilde{c}^*$	432	507
$\tilde{p}$	**82.0%**	**79.7%**
c	457	427
c^*	182	199
p	**60.2%**	**53.4%**
Computing time	7233 s	6615 s

than during the previous simulations. The table III shows the results for the simulation of each direction considering uncertainties. We can see that a only speed regulation is able to solve more the half of real conflicts considering uncertainties. Moreover, what we have to keep in mind is that if we take into account the initial vertical separation, the number of conflict, real and virtual, would be lower and maybe the algorithm would succeed to resolve them all. Of course a computing time higher than two hours for a resolution of only four hours of traffic is not acceptable but we can think that optimal parameters exist which could bring similar performances to the algorithm for an acceptable computing time. Quite evidently, all this work shows that conflict resolution through speed regulation could offer a significant help for controllers but it will still need their monitoring because the total deconfliction

is not guaranteed.

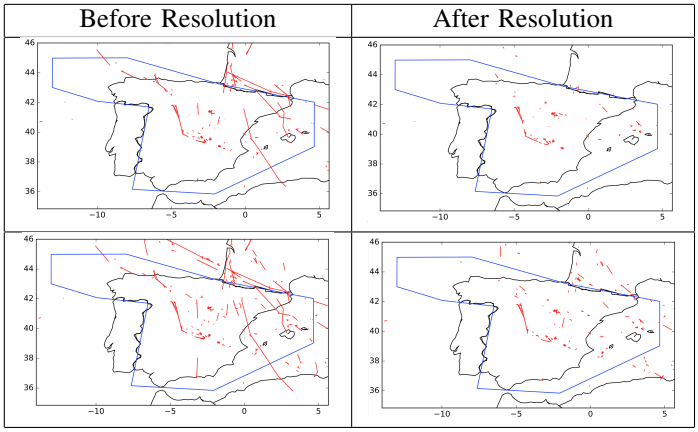

Figure 8. Visualisation of the conflicts (red) for the whole traffic before and after resolution: with uncertainties (bottom) and without uncertainties (top)

VI. Conclusion

We proposed a formulation for deconfliction based on speed regulation, where conflicts should be reduced or ideally avoided without any spatial change in aircraft trajectories. In this modelling, existing conflicts are evaluated and then aircraft True Air Speeds are changed in order to minimise them. Wind uncertainties are included in the modeling. By hypothesis that all aircraft fly at the same flight level, we simplified the modelling but we also increased interactions between aircraft. We solved the problem by simulated annealing with promising results: Around 55% -80% not considering uncertainty. of the total number of conflicts could be reduced by simply slightly modifying the flight plan speeds. This study can be carried on by the implementation of the third spatial dimension and an other separation maneuver types such as Heading or FL changes, or by delays on departure times.

References

[1] James K. Kuchar and Lee C. Yang. A review of conflict detection and resolution modeling methods. *IEEE*, 2000.

[2] Martín Campo and Francisco Javier. The collision avoidance problem: Methods and algorithms. 2010.

[3] Eric M. Feron Lucia Pallottino and Antonio Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 3(1):3, March 2002.

[4] N. Durand S. Cafieri. Aircraft deconfliction with speed regulation : new models from mixed-integer optimization. *Journal of Global Optimization*, 58(4):613–629, April 2014.

[5] Eric Feron Zhi-Hong Mao and Karl Bilimoria. Stability of intersecting aircraft flows under decentralized conflict avoidance rules. In *18th Applied Aerodynamics Conference, Fluid Dynamics and Co-located Conferences*, 2000.

[6] Laureano F. Escudero Antonio Alonso-Ayuso and F. Javier Martín-Campo. Collision avoidance in air traffic management: A mixed-integer linear optimization approach. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 12(1):47–57, March 2011.

[7] Jean-Marc Alliot Nicolas Durand. Ant colony optimization for air traffic conflict resolution. *ATM Seminar 2009, 8th USA/Europe Air Traffic Management Research and Developpment Seminar*, 2009.

[8] Joseph Noailles Nicolas Durand, Jean-Marc Alliot. Automatic aircraft conflict resolution using genetic algorithms. *In Proceedings of the Symposium on Applied Computing, Philadelphia*, February 1996.

[9] John Lygeros Andrea Lecchini Visintini, William Glover and Jan Maciejowski. Monte carlo optimization for conflict resolution in air traffic control. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 7(4):470–482, December 2006.

[10] Inseok Hwang Weiyi Liu. Probabilistic aircraft midair conflict resolution using stochastic optimal control. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 15(1):37–46, February 2014.

[11] John Lygeros Maria Prandini, Jianghai Hu and Shankar Sastry. A probabilistic approach to aircraft conflict detection. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 1(4):199–220, December 2000.

[12] Eulalia Hernández, Alfonso Valenzuela, and Damián Rivas. Probabilistic aircraft conflict detection considering ensemble weather forecast. 2016.

[13] William Singhose John-Paul Clarke Adan Vela, Senay Solak. A mixed integer program for flight-level assignment and speed control for conflict resolution. *48th IEEE Conference on Decision and Control*, 2009.

[14] Jenaro Nosedal-Andrea Ranieri Sergio Ruiz, Miquel A. Piera. 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, February 2014.

[15] D. Delahaye S. Chaimatanam and M. Mongeau. A hybrid metaheuristic optimization algorithm for strategic planning of 4d aircraft trajectories at the continental scale. *IEEE*, 2014.

[16] Banavar Sridhar Olga Rodionova and Hok K. Ng. Conflict resolution for wind-optimal aircraft trajectories in north atlantic oceanic airspace with wind uncertainties. In *35th Digital Avionics Systems Conference (DASC)*, 2016.

[17] R. Ehrmannstraut. The potential of speed control. In *Digital Avionics Systems Conference*, 2004.

[18] E. Cruck and J. Lygeros. Subliminal air traffic control: Human friendly control of a multi-agent system. In *American Control Conference*, 2007.

[19] R. Guerreau-R. Weber G. Gawinowski, J.-L. Garcia and M. Brochard. Erasmus: A new path for 4d trajecotry-based enablers to reduce the traffic complexity. In *26th Digital Avionics Systems Conference (DASC 2007)*, pages 1A31–1A311, 2007.

[20] R. Fondacci D. Rey, S. Constans and C. Rapine. A mixed integer linear model for potential conflict minimization by speed modulations. *Fourth International Con- ference on Research in Air Transportation*, pages 513–515, June 2010.

[21] J.B. Gotteland N. Durand. Algorithmes génétiques appliqués à la gestion du trafic aérien. *Journal sur l'enseignement des sciences et technologies de l'information et des systèmes*, 2003.

[22] N. Durand. *Optimisation de trajectoires pour la résolution de conflits aériens en route.* PhD thesis, Institut National Polytechnique de Toulouse, 1996.