



HAL
open science

Strategic deconfliction of aircraft trajectories

Supatcha Chaimatanan, Daniel Delahaye, Marcel Mongeau

► **To cite this version:**

Supatcha Chaimatanan, Daniel Delahaye, Marcel Mongeau. Strategic deconfliction of aircraft trajectories. ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management, Jul 2013, Toulouse, France. p xxxx. hal-00868450

HAL Id: hal-00868450

<https://enac.hal.science/hal-00868450>

Submitted on 1 Oct 2013

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

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

Strategic deconfliction of aircraft trajectories

Supatcha CHAIMATANAN, Daniel DELAHAYE and Marcel MONGEAU
École National de l'Aviation Civile (ENAC), Toulouse, France
*Laboratoire de Mathématiques Appliquée, Informatique et Automatique
pour l'Aérien (MAIAA),*

Abstract. In this work, we present a methodology to minimize the number of potential conflicts between aircraft trajectories based on route-slot allocation techniques. The traffic assignment problem is modeled as a combinatorial optimization problem for which two metaheuristic optimization algorithms are developed and implemented. The first algorithm relies on a standard simulated annealing, while the second algorithm uses a hybrid-metaheuristic method. The proposed algorithms were implemented and tested on real air-traffic data for which an optimal solution for every trajectory is obtained within affordable computation time.

Keywords. 4D trajectory planning, Strategic deconfliction, Hybrid-metaheuristic optimization

Introduction

Air Traffic Management (ATM) paradigm in Europe is evolving around the Trajectory Based Operations (TBOs) concept, where air traffic is no longer constrained by artificial boundaries such as airspace sector boundaries, national borders, locations of beacons, etc. Instead, ATM will focus on trajectories together with an adapted airspace design. This new ATM concept of operations implies a possibility to resolve potential conflicts between aircraft trajectories in the strategic trajectory planning phase. This strategic deconflicting will alleviate the air traffic controller's tactical conflict resolution workload. More efficient trajectories with minimal number of potential conflicts can be strategically designed from a more global point of view, anticipating downstream effects. Once the aircraft is cleared to fly its *reference business trajectory* (optimal conflict-free trajectory), the controller's workload will thereby involve more monitoring and less conflict prediction and resolution. As a consequence, the needs of tactical intervention being reduced, more flights will be accommodated by the controller in a given airspace at a given time.

In this work, we propose a methodology to minimize the number of potential conflicts between aircraft trajectories for a full traffic day at the strategic level, using bi-allocation (route-departure slot allocation) techniques. This air traffic assignment problem is modeled as a combinatorial optimization problem for which two metaheuristic optimization approaches have been implemented and compared. The first proposed optimization approach relies on a standard Simulated Annealing (SA) algorithm, and the second one uses a hybrid optimization method. The proposed methodology was implemented and tested on a full day air traffic over the French airspace.

The remaining of this introduction surveys previous research work. Over past decades, several methods have been proposed to address the air traffic management problem aiming at balancing the air traffic demand and the airspace capacity, and to ease airspace congestion. There are two frequently-used strategies, the first one is to adapt the airspace capacity to the increasing demand. This strategy was, for example, considered in [6,20,16].

Another strategy is to regulate the demand to the current capacity which can be accomplished by optimizing the allocation of flight plans. This strategy is usually referred as air traffic flow management (ATFM). It aims at controlling and organizing the air traffic flow so as to minimize the airspace complexity, to ease the controller workload as well as to maximize the use of airspace. The best known approach based on this strategy is the ground delay allocation which presented, for instance, in [17], [22], and [2]. This approach attributes ground delay to aircraft before take-off taking into account the airspace sector and airport capacity. Delaying aircraft on ground reduces fuel consumption due to extra distance aircraft has to fly in order to avoid congested areas. However, with the increasing demand, significant delays have to be imposed in order to address all the congestion.

In recent years, many ATM work based on deterministic and stochastic optimization approaches were proposed. In a large-scale problem, the air traffic is usually modeled as a network flow model. Integer Linear Programming (ILP) is often used to address such problems, such as in [3] where a route is represented by a sequence of sectors flown over by an aircraft, or in [4] where ILP adapted for large-scale ATFM problems considering all flight phases is presented. An approach to find sequentially a conflict-free wind-optimal route in real time is presented in [11]. In [21] and [18], a methodology to reduce air traffic congestion without using the flow network was presented. It uses bi-allocation (route-slot allocation) techniques; an optimal route and departure time for each flight were computed using genetic algorithms. For more details, a comparison of different optimization methods used for traffic flow management is provided in [14].

This paper is organized as follows: The problem statement, optimization formulation and the size of the optimization formulation are presented in Section 1. A potential conflict detection approach and two optimization approaches used to address the potential conflict reduction problem are presented in Section 2. Finally, numerical results, and conclusions are presented in Section 3 and 4 respectively.

1. Problem Statement

In this section, we first present a potential conflict minimization method based on route-slot allocation techniques. Then, a mathematical formulation of the potential conflict minimization under the form of combinatorial optimization is introduced. Finally, complexity and size of the formulation of this particular discrete optimization problem is discussed.

In the framework of the above-described new ATM concept, a 4D trajectory is a sequence of 4D coordinates (x, y, z, t) that aircraft has to follow through the airspace. Aircraft are considered to be in *potential conflicts* when the minimum required *separation* distance between them (5 Nautical miles (Nm) horizontally and 1,000 feet (ft) vertically) is not ensured. This does not necessary leads to a collision, however, it is a situation where a risk of collision is elevated.

In this work, we propose a methodology to minimize the number of potential conflicts by strategically organizing aircraft trajectories. The proposed method separates aircraft trajectories in Cartesian space by modifying the shape of the nominal (initially planned) trajectories (re-routing), and separates the trajectories in the temporal space by shifting departure times.

1.1. Alternative Trajectory and Departure Time

The airspace is considered here as a Euclidean space. Latitudes and longitudes on the ellipsoid earth surface are transformed into (x,y) coordinates by a Lambert azimuthal equal-area projection with the center of projection located at the center of the airspace. The altitude in feet will be our z coordinate. Consider 4D trajectory given as a time sequence of discretized 4D coordinates (x,y,z,t) . A horizontal and vertical flight profile are illustrated in **Figure 1** and **2** respectively.

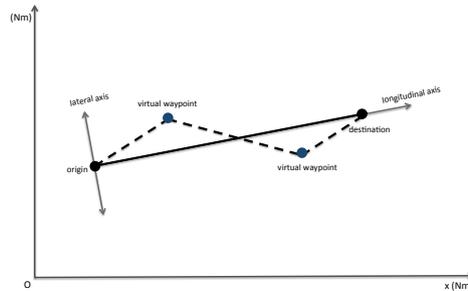


Figure 1. Horizontal Flight Profile.

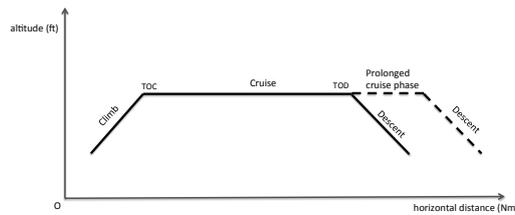


Figure 2. Vertical Flight Profile. (TOC= top of climb, TOD = top of descent)

Alternative Trajectory Design An alternative trajectory is constructed by placing M virtual waypoints along the nominal trajectory and then reconnect each waypoint with a straight line segment. The optimum cruise level, and both climb and descend altitude profiles are computed by the Flight Management System (FMS) in order to minimize fuel consumption or traveling time (according to cost index) based on the given aircraft performance and on the predicted wind conditions. Therefore, in this study, the virtual waypoints modifying the shape of flight path will be placed in the horizontal plane (while respecting its initial optimum altitude profile).

Consider a trajectory profile in the horizontal plane. We call *longitudinal axis* the axis that is tangent to the nominal trajectory, and the *lateral axis* is the axis that is perpen-

dicular to the longitudinal axis. For each flight, the position of each waypoint will be defined using these relative reference axes. Let $w_i = \{w_i^j \mid w_i^j = (w_{i,x}^j, w_{i,y}^j)\}$ for $j = 1, \dots, M$ be a set of virtual waypoints used to modify the trajectory shape of flight i , where M is the number of virtual waypoints on wishes to place, and $w_{i,x}^j$ and $w_{i,y}^j$ are the longitudinal and lateral component of w_i^j respectively. In **Figure 1**, a dashed line illustrates a horizontal flight profile of an alternative trajectory constructed with two virtual waypoints ($M=2$). The altitude profile is then prolonged at the top of descent while respecting the optimal climb and descent profile, as illustrated in **Figure 2**.

Alternative Departure Time In the strategic planning phase, the departure time of each flight can be shifted by a positive (delay) or a negative (advance) time shift. Let $t_{i,0}$ be the *nominal* (initially planned) departure time of aircraft i , and let $\delta_i \in \Delta_i$ be a departure time shift attributed to flight i , where Δ_i is an interval of the form $[\delta_{i,min}, \delta_{i,max}]$ of feasible time shift of flight i , $\delta_{i,min}$ and $\delta_{i,max}$ are user-defined parameters. The alternative departure time of aircraft i is therefore $t_i = t_{i,0} + \delta_i$.

1.2. Optimization Formulation

Consider a given set of N discretized 4D trajectories linking origin-destination (OD) pairs in a given airspace.

Decision Variables: Let $w = \{w_i^j\}$ for $i = 1, \dots, N$; $j = 1, \dots, M$ be a set virtual waypoint associated to trajectory i , where M is the number of waypoints (a parameter to be set by the user). Let $\delta = \{\delta_i\}$ or $i = 1, \dots, N$ be a departure time shift associated to trajectory i . Thus, the decision variable can be represented by (w, δ) .

Objective: The objective of the strategic potential conflict reduction problem is to minimize overall potential conflicts between trajectories by allocating optimal route and departure slots to each aircraft. Let Φ_i be the number of potential conflict encountered by aircraft i . The cost function to be minimized is $\sum_{i=1}^N \Phi_i(w, \delta)$.

Constraints:

To avoid sharp turns, the longitudinal position of the virtual waypoints should not be too close to each other. In this work, for each flight i , the longitudinal location is set to $w_{i,x}^j = \frac{j}{M+1}L_{i,0}$, where $L_{i,0}$ is the nominal route length of flight i (uniformly distributed longitudinal locations along the nominal route). Increased route length from the i^{th} aircraft nominal trajectory can be controlled by optimization through the lateral locations ($w_{i,y}^j$). This lateral deviation is limited so that the maximum increased-route-length remains within a given fraction, l , of the nominal distance: $L_i \leq (1+l)L_{i,0}$, where $0 \leq l \leq 1$ is a parameter set by the user. This constraint limits the maximum allowable lateral deviation of aircraft i , denoted by a_i , which sets upper and lower bounds for $w_{i,y}^j$: $w_{i,y}^j \in [-a_i, a_i]$.

In this study, we chose to discretize uniformly the range $[-a_i, a_i]$ of possible values that can take the decision variable $w_{i,y}^j$ into K possible values (where K is another parameter set by the user). Empirical tests showed that $M = 2$ and $K = 7$ leads to a sufficiently rich search space. This discretization creates $7^2 = 49$ possible route choices for each flight as illustrated in **Figure 3**.

Common practice in airports conducted us to rely also on a discretization of the interval of possible values for each of the departure time shifts $\delta_i \in \Delta_i$. Let $\delta_{i,min}$ and $\delta_{i,max}$

be the minimum and maximum departure time shift of flight i respectively. The departure time shift can be chosen from a uniformly discretized continuous interval $[\delta_{i,min}, \delta_{i,max}]$. Empirical tests showed that setting $-\delta_{i,min}, \delta_{i,max} = 60$ minutes, and discretizing the time intervals to every time step, d , of one minute yields a sufficiently rich search space. In other words, we consider $\Delta_i = \{-60, -59, -58, \dots, 0, \dots, 58, 59, 60\}$.

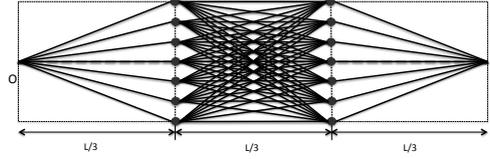


Figure 3. All K^M possible route choices for an alternative trajectory (here $K=7$, $M=2$ and the nominal trajectory is the segment OD)

With the specific parameter values we chose above, namely: $K=7$, $M=2$, $w_{i,x}^j = \frac{j}{M+1}L_{i,0}$, the potential conflict reduction problem for this particular discrete version of the route-departure slot allocation problem can be represented in a form of an optimization problem as follows;

$$\min_{(w_y, \delta)} \sum_{i=1}^N \Phi_i(w_i, \delta_i)$$

subject to

$$w_{i,y}^j \in \{-a_i, -\frac{2a_i}{3}, -\frac{a_i}{3}, 0, \frac{a_i}{3}, \frac{2a_i}{3}, a_i\} \quad i = 1, \dots, N; \quad j = 1, \dots, M;$$

$$\delta_i \in \{-60, -59, -58, \dots, 0, \dots, 58, 59, 60\} \quad i = 1, \dots, N;$$

$$L_i(w_{i,y}) \leq (1+d)L_{i,0}$$

where

Φ_i is the number of potential conflicts in which aircraft i is involved;

δ_i is the departure time shift of a i ;

$w_{i,y}^j$ is the y coordinate of the j^{th} virtual waypoint of trajectory i ;

$w_y = \{w_{i,y}^j\}$ is a vector of lateral component of w ;

L_i is the length of trajectory i ;

$L_{i,0}$ is the initial length of trajectory i ;

N is the total number of flights;

and M is the number of virtual waypoints used for each trajectory.

1.3. Complexity of the Problem

The mathematical model presented in the previous subsection involves manipulating discrete variables which introduce high combinatorics to the state space. The number of possible solution, (S_i) , for a flight i is:

$$S_i = |S_{w_i}| \cdot |S_{\delta_i}|$$

where $|S|$ represent the cardinality of a set S .

Therefore, the total number of possible solution, denoted $|State|$, is:

$$|State| = \prod_{i=1}^{i=N} S_i$$

where N is the total number of flights. In this paper, each flight involves the same number of possible trajectories. Therefore the cardinal of the feasible domain is:

$$|State| = (S_{w_i} \cdot S_{\delta_i})^N$$

In our case, $K = 7$, $M = 2$, $-\delta_{i,min} = \delta_{i,max} = 60$ minute, $d = 1$ minute, we have $S_{w_i} = K^M = 49$ and $S_{\delta_i} = (60 + 60 + 1)/d = 121$. If, for instance, $N = 10,000$, the cardinal of the feasible space becomes $|State| = (49 \cdot 121)^{10,000}$.

Noticed that the solution space is not continuous and the dimension of this discrete optimization problem grows exponentially with the size, N , of the problem. Moreover, the decision variables are not independent due to the influences of interaction between flights (the problem is not separable). The mono-objective function may have several equivalent optima (multimodal). The combinatorial optimization problem is NP-hard, which can be addressed by stochastic optimization approaches such as genetic algorithm, simulated annealing, ant colony algorithm, etc. Due to the computational cost of evaluating the objective function, $\sum_{i=1}^N \Phi_i(w, \delta)$, the implementation of the potential conflict reduction method in this work relies on non-population-based algorithms which will be discussed in detail in the following section.

2. The Strategic Potential-Conflict Reduction Method

To implement the potential conflict reduction methodology on real air traffic data, a method to evaluate the objective function and optimization algorithms adapted to the air traffic assignment problem is presented and discussed in this section.

2.1. Potential Conflict Detection Scheme

In order to evaluate the optimization problem cost function, a methodology to detect a potential conflict is developed and discussed in this section. As mention earlier, aircraft are considered to be in potential conflict when their horizontal separation is less than 5 Nm and their vertical separation is less than 1,000 ft. One can imagine that an aircraft has a protection zone defined by a three-dimensional cylinder, as illustrated in **Figure4**, in which no other aircrafts are allowed to enter.

To detect the potential conflict in large-scale problem, a grid-based conflict detection scheme is introduced in this study. First, the airspace is discretized using a 4D space-time grid as illustrated in **Figure 5**. The size of each cell in the grid is defined by the separation norm (5 Nm horizontally and 1,000 ft vertically). Then, the aircraft position along its discretized trajectory is associated to a corresponding cell in the 4D grid. Finally, po-

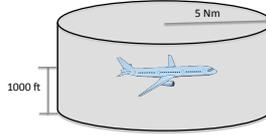


Figure 4. The separation-norm cylinder

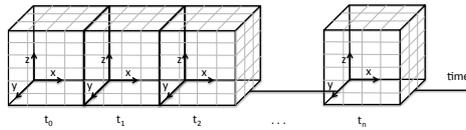


Figure 5. Four dimension space - time grid

tential conflicts can be detected by checking the $3^3 = 27$ neighboring cells of each non-empty cell in the grid. A potential conflict is identified if either a cell is co-occupied by different aircrafts or one of its neighbors is occupied by another aircraft.

The potential conflict detection scheme is implemented using *hash table* data structure. For a given discretized 4D trajectory, each sampled point is mapped to a cell in the 4D grid, where a list of flight identifications occupying the corresponding grid are stored. This data structure does not required to store the 4D coordinate, reducing thereby the memory space required in the computation. Moreover, it allows to update easily the total number of potential conflicts when route or departure slot choices of some flights are modified.

2.2. Simulated Annealing Algorithm

Simulated Annealing (SA) is a so-called metaheuristic optimization method that was introduced independently by S. Kirkpatrick et al. in 1983 and V. Cerny in 1985 [10]. Its popularity comes from its ability to avoid being trapped in a local minima, to find near-global optimal solutions for NP-hard combinatorial optimization problems, and the fact that it is easily implemented for costly black-box optimization problem.

The simulated annealing method is inspired by the annealing process in metallurgy where the state of a material can be modified by controlling the cooling temperature. The annealing process consists in heating up a material to bring it to a high energy state. Then, it is slowly cooled down by decreasing the temperature according to a properly pre-defined *cooling schedule*. By keeping at each temperature step a sufficient duration, the material reaches its thermal equilibrium before further reduction of temperature. As the temperature tends towards zero, the material reaches a crystallized solid state where the energy is at the absolute minimum. In contrast, if the temperature is decreased too rapidly, it yields states at locally-optimal energy levels.

When solving an optimization problem using the simulated annealing method, the cost function to be minimized is analogical to the energy of the physical problem, and the control parameters controlling the exploration of the solution space plays the role of the temperature. At each temperature step, the algorithm iterates until a fixed or dynamic number of transitions are reached, until some stopping criteria is satisfied, for instance

```

Simulated Annealing Algorithm
1. At  $T = T_0$ , start from a random solution  $S_{init}$ 
2. Evaluate  $Y_{init} := E(S_{init})$ 
3. Set the current best-known solution  $S_{best} := S_{init}$  and  $Y_{best} := Y_{init}$ 
4. Set the current solution  $S_i := S_{init}$  and  $Y_i := Y_{init}$ 
5. for ( $i = 0$  to  $k$ )
    Generate randomly a perturbed candidate solution  $S'$ 
    If  $E(S') < E(S_i)$ 
        accept  $S'$  as current solution
        set  $S_i := S'$  and  $Y_i := E(S')$ 
    else
        if  $r < \exp( (E(S_i) - E(S')) / T )$ , where  $r =$  random number
            accept  $S'$  as a current solution
            set  $S_i := S'$  and  $Y_i := E(S')$ 
        If  $Y_i < Y_{best}$ 
            set  $S_{best} := S_i$  and  $Y_{best} := E(S_i)$ 
6. Reduce the temperature  $T_{k+1} = \alpha T_k$ 
7. Repeat step 5 and 6 until termination criterion is reached

```

Figure 6. Standard simulated annealing algorithm

when a pre-defined (user-defined) maximum number of rejected movements is attained. A conventional simulated annealing method can be summarized as shown in **Figure 6**, where E denotes the energy function to be minimized, k is a pre-defined maximal number of steps to be performed, and α is a pre-defined reduction factor ($0 < \alpha < 1$).

2.3. A Hybrid Metaheuristics Method

Although the simulated annealing algorithm is generally able to find a good solution, in this case, it requires to perform a large number of objective function evaluations which leads to prohibitive computation time. In fact, at high temperatures, most of the computation time is spent on exploring the state space (accepting degraded solution), and on evaluating solutions which will not be accepted at lower temperatures. In order to improve the efficiency of this optimization method, a method to balance between exploration (diversification) and exploitation (intensification) of the solution space is now introduced. As the problem may have several equivalent optima (multimodal objective function), the idea is to switch to a local search method around the current solution, while allowing to accept degraded solutions when necessary in order to escape from a local trap.

We propose to integrate a local heuristic search method within the simulated annealing to exploit the solution space around a current solution in order to accelerate convergence. Obviously, a balance must be found, as relying too often to local search yield premature termination of the simulated annealing at a local optimum. Therefore, an adaptive probability to resort to local heuristic search is introduced. This probability should be low at high temperature in order to preserve diversity of the search. At low temperature, when simulated annealing becomes less effective, this probability becomes higher in order to emphasis on intensification of the search around the best solutions found so far. In this work, the probability of performing the local search is controlled as follows;

$$P_l(T) = P_{l,init} + (P_{l,max} - P_{l,init}) \cdot \frac{T_0 - T}{T_0}$$

where P_l is a parameter that controls the probability to have recourse to the local search; $P_{l,init}$ is the initial value of this probability;

$P_{l,max}$ is the maximum probability value;
 T_0 and T is the initial and current temperature respectively.

The hybrid metaheuristics method used in the paper is summarized in **Figure 7**.

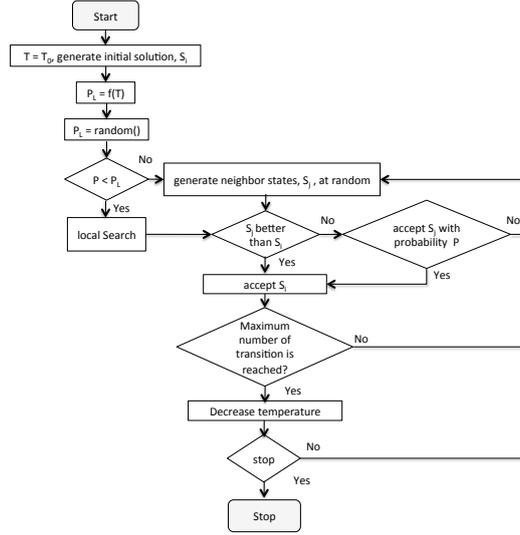


Figure 7. Hybrid metaheuristics algorithm

Local search methods. There exists several local search strategies whose efficiency depends largely on various features of the problem such as the landscape of the objective function, which is difficult to be determined in such a high-dimension problem. However, empirical experience (after performing the simulated annealing for several times), shows that the objective function is multi-modal with several optimal solution yielding the same value. Therefore, firstly in this study, it is proposed to use a hill-climbing-type (or, rather descending here as one minimizes) search strategy which is summarized in **Figure 8**. From a randomly chosen trajectory i with a current state (w_y, δ) and current objective function value $\sum_{i=0}^N \Phi_i(w_y, \delta)$, this local search algorithm generates a neighbor solution, denoted (w_{y_n}, δ_n) , according to a pre-defined neighborhood relation. If the neighbor solution value, $\sum_{i=0}^N \Phi_i(w_{y_n}, \delta_n)$, is better than the current solution $(\sum_{i=0}^N \Phi_i(w_{y_n}, \delta_n) \leq \sum_{i=0}^N \Phi_i(w_y, \delta))$, the local search algorithm accepts the neighbor solution as a current solution. Otherwise, the algorithm rejects the neighbor solution. Then, a new neighbor solution is generated, and the local search process repeats until a (user-defined) termination criterion are satisfied.

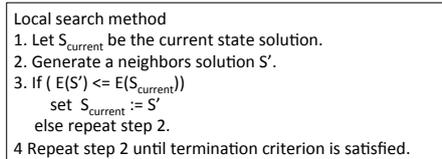


Figure 8. Greedy-type search method

Neighborhood relation. For a given state of trajectory, $(w_{i,y}, \delta_i)$, first the local search algorithm exploits the solution space by focusing on searching a better route and departure time options for a given aircraft i . Then, the algorithm tries to improve the solution by modifying routes and departure slot choices of every other aircraft which can be potentially interacting with aircraft i . The list of aircrafts which can be interacting with aircraft i is computed in a pre-processing phase and stored in a database.

Data processing. A list of aircraft potentially interacting with any given aircraft is computed off line in a pre-processing phase. Let us defined the *4D envelope* of an aircraft as the subset of $[R]^4$ containing all possible 4D trajectories for that aircraft when considering every feasible departure time and every possible position of the M waypoints. Two flights are considered to be potentially interacting with each other if their 4D envelopes intersect.

In practice, this is implemented as follows. For each aircraft, we first construct an envelope of all possible trajectories in horizontal plane as illustrated by the 2D convex hull in **Figure 9**. Next, these convex hulls are probed pairwise. If an intersection between any pair of 2D convex hull is detected, this intersection area is simplified as a (smaller) rectangular zone in each convex hull illustrated as a shaded area in **Figure 9**.

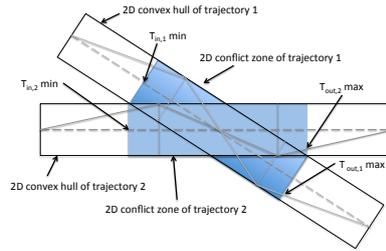


Figure 9. Intersection of 2D convex hulls of two trajectories

After that, the earliest possible time $(t_{in,i,min})$ each aircraft enters the simplified intersection zone and the latest possible time $(t_{out,i,max})$ each aircraft leaves the zone are computed. If the time interval $[t_{in,i,min}, t_{out,i,max}] \cap [t_{in,j,min}, t_{out,j,max}] \neq \emptyset$, the altitude ranges $[z_{min}, z_{max}]$ of both aircraft in the simplified intersection zone are computed. Finally, if the altitude ranges of both aircrafts overlap, both aircrafts are considered to be potentially interacting with each other.

3. Numerical Results

In this section, the proposed methodology is tested on a full day of air traffic over the French airspace obtained from CATS air traffic simulator. The reference en-route air traffic on 12 August 2008, consists of 8,266 trajectories, are discretized with sampling time of 15 seconds. The initial number of potential conflicts between aircrafts is 174,722 potential conflicts. The initial air traffic situation is illustrated in **Figure 10**. The maximum departure time shift is set to 60 minutes, and the maximum route length extension is set to 20% ($l = 0.2$) of the nominal route length. The maximum allowed lateral deviation a_i

is deduced from:
$$\sqrt{2 \left(\frac{L_i}{3} + a_i^2 \right)^2 + \left(\frac{L_i}{3} + (2a_i)^2 \right)^2} = (1+l)L_{i,0}.$$

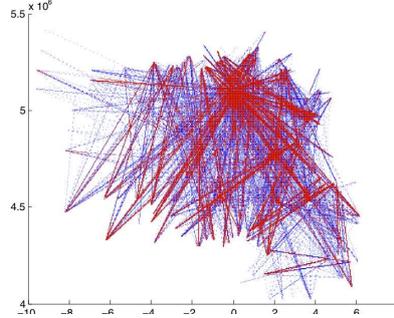


Figure 10. Initial air traffic situation over the French airspace

The first part of the implementation of the route-departure slot allocation algorithms introduced in the previous section consists in determining the (user-defined) control parameters of the optimization algorithms. Empirical tests conduct us to set these parameters as follows:

- The initial temperature, (T_0), is calculated as a preliminary steps using an algorithm proposed in [10]. First, 100 disturbances are generated randomly. Second, the average change of energy (ΔE) is evaluated. Then the initial temperature (T_0) is deduced from the relation: $e^{-\frac{\Delta E}{T_0}} = \tau_0$, where τ_0 is the initial rate of acceptance of degrading solutions. Empirical tests leads us to set $\tau_0 = 40\%$.
- In the cooling process, the temperature should be reduced slowly and smoothly. Lowering abruptly the temperature yields premature convergence towards a local minima that is not interesting. A widely-used cooling schedule is the geometrical law: $T_{i+1} = \alpha \cdot T_i$ evoked in **Figure 6**. Empirical tests conducts us to decrease the temperature after 4,000 iterations is performed at each temperature step with $\alpha = 0.99$.
- The algorithm is terminated when an optimal solution (routes and departure slots choice with zero potential conflict) is reached or when the final temperature T_f satisfied $T_f = \frac{T_{ini}}{1000}$.
- For the hybrid-metaheuristic method, empirical tests lead us to set $P_{l,init} = 0.001$, $P_{l,max} = 0.1$ which yields the probability of performing the local search: $P_l(T) = 0.001 + 0.1 \cdot \frac{T_0 - T}{T_0}$. In other words, the local search is performed with a probability of 0.1% up to about 10%.

The proposed potential conflict reduction method is implemented in Java on a Core2Duo 2.4 GHz computer with 8 GB DDR3 RAM on a Unix platform. We were able to obtain optimal solutions (conflict-free) from both optimization methods within reasonable computations time for a strategic planning. Numerical results obtained are presented in **Table 1**.

The method based on a standard simulated annealing yields an optimal solution for every trajectory with computation time ≈ 561 minutes after performing 13,760,000 transitions. **Figure 11** shows the value of the best solution found at each temperature step (yBest) and the value of the solution at the end of each temperature step (yCurrent) using standard simulated annealing. Since simulated annealing accepts more degraded solution at high temperatures, it can be noticed that the difference between the best solution

Optimization method	avg. computation time (minutes)	number of objective-function evaluations	avg. extra distance (km)	avg. departure time shift (minutes)
standard SA	561.37	13,760,000	22.83	21.6
hybrid-metaheuristic method	360.48	6,797,549	20.46	23.2

Table 1. Numerical results

and current solution is greater at high temperatures. These differences decrease as the algorithm slowly and smoothly converge to the optimal solution.

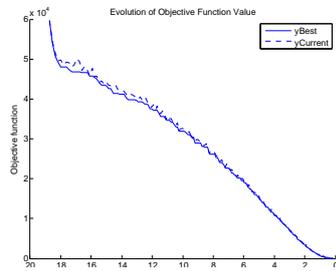


Figure 11. Evolution of objective function value of algorithm based on simulated annealing

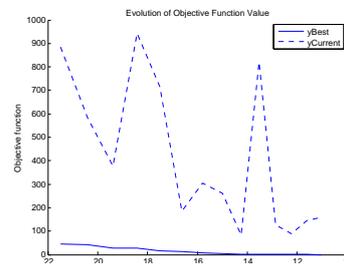


Figure 12. Evolution of objective function value of algorithm based on hybrid metaheuristics

The integration of the greedy-like local search method into simulated annealing significantly reduces the computation time and accelerates convergence. This hybrid method is able to reach an optimal solution within ≈ 360 minutes after performing 6,797,549 objective function evaluations, $\approx 50\%$ less than using standard simulated annealing. As shown in **Figure 12** the hybrid algorithms converge much faster.

An example of potential conflict free flight plan is illustrated in **Figure 13**. It can be noticed that the trajectories are less concentrated than the initial traffic scenario.

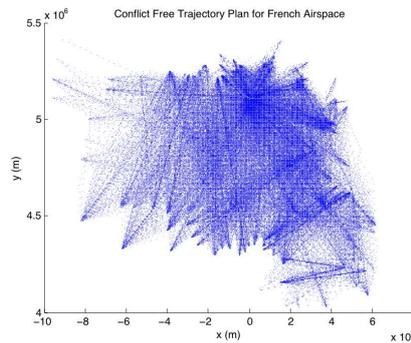


Figure 13. An example of conflict-free trajectory plan obtained by solving the potential conflict reduction problem

4. Conclusions

In this paper, we introduced a methodology to address strategic 4D trajectory planning. The proposed methodology minimizes the number of potential conflicts by modifying the shape of aircraft trajectories and by shifting their departure times. First, the potential conflict minimization problem is formulated under the form of a combinatorial optimization problem. Complexity and size of the formulation was discussed. This NP-hard problem is then addressed with two stochastic optimization algorithms that we have developed. Numerical results are presented on a full day of air traffic over the French airspace involving 174,722 potential conflicts. The first optimization method based on a standard simulated annealing yields encouraging results a conflict-free trajectory is found for every aircraft in a computational time which is viable for an operational context. A local search method is integrated into simulated annealing to accelerate convergence to optimal solution by a factor of 2.

In order to address larger (continent scale) instances, we plan to improve further the proposed air-traffic assignment (route-slot allocation) algorithm, by relaxing the discrete virtual-waypoint location constraint and by assessing more efficient design of the hybrid-metaheuristic algorithm. Furthermore, uncertainties of aircraft positions will be taken into account in the trajectory planning process.

References

- [1] Worldwide market forecast for commercial air transport 2010-2029. Technical Report YGR-5067, Marketing, Japan Aircraft Development Corporation, 2010.
- [2] N. Barnier and C. Allignol. 4d-trajectory deconfliction through departure time adjustment. In *8th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2009)*, 2009.
- [3] D. Bertsimas, G. Lulli, and A. Odoni. The air traffic flow management problem: An integer optimization approach. In *Integer Programming and Combinatorial Optimization*, volume 5035 of *Lecture Notes in Computer Science*, pages 34–46. Springer Berlin Heidelberg, 2008.
- [4] D. Bertsimas, G. Lulli, and A. Odoni. An integer optimization approach to large-scale air traffic flow management. *Operations Research*, 59(2):211–227, 2011.
- [5] S. Chaimatanan, D. Delahaye, and M. Mongeau. A methodology for strategic planning of aircraft trajectories using simulated annealing. In *International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, Florida, 2012.
- [6] D. Delahaye and S. Puechmorel. 3D airspace sectoring by evolutionary computation: Real-world applications. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 1637–1644, New York, 2006. ACM.
- [7] D. Delahaye and S. Puechmorel. Air traffic complexity based on dynamical systems. In *49th IEEE Conference on Decision and Control (CDC)*, pages 2069–2074, 2010.
- [8] N. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau. A light-propagation model for aircraft trajectory planning. *Journal of Global Optimization*, 2013. DOI: 10.1007/s10898-012-9896-1. To appear.
- [9] G. Dowek and C. Munoz. Conflict detection and resolution for 1,2,...,n aircraft. In *7th AIAA Technology, Integration and Operations Conference*, Belfast, 2007.
- [10] J. Dreot, A. Petrowski, P. Siarry, and E. Taillard. *Metaheuristiques pour l'optimisation difficile*. EYROLLES, 2 edition, 2005.
- [11] M. R. Jardin. Real-time conflict-free trajectory optimization. In *5th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2003)*, 2003.
- [12] M.R. Jardin. *Towards Real Time En Route Air Traffic Control Optimization*. PhD thesis, Stanford University, 2003.
- [13] J. Knopman and J.S. Aude. Parallel simulated annealing: An adaptive approach. In *Parallel and Distributed Processing Symposium*, page 522. IEEE International, 1997.

- [14] J. Lohn and J. Rios. A comparison of optimization approaches for nationwide traffic flow management. In *AIAA Guidance, Navigation and Control Conference*, Chicago, Illinois, 2009. IEEE International.
- [15] P. M., J. Hu, J. Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation System*, 1(4):199–220, 2007.
- [16] X. Min. Airspace Sector Redesign based on Voronoi Diagrams. *Journal of Aerospace Computing Information and Communication*, 6:624–634, 2009.
- [17] A.R. Odoni. The Flow Management Problem in Air Traffic Control. In *Odoni, A.R., Bianco, L., Szego, G. (eds.) Flow Control of Congested Networks*, pages 269–288. Springer, 1987.
- [18] S. Oussedik and D. Delahaye. Reduction of air traffic congestion by genetic algorithms. In *Parallel Problem Solving from Nature — PPSN V*, volume 1498 of *Lecture Notes in Computer Science*, pages 855–864. Springer Berlin Heidelberg, 1998.
- [19] T. Prevot, S. Shelden, and J. Mercer. ATM concept integrating trajectory-orientation and airborne separation assistance in the presence of time-based traffic flow management. In *In Proceedings of the 22nd Digital Avionics Systems Conference*, 2003.
- [20] H.D. Sherali. and J.M. Hill. Configuration of airspace sectors for balancing air traffic controller workload. *Annals of Operations Research*, pages 1–29, 2011.
- [21] S.Oussedik. *Application de l'Evolution artificielle aux problèmes de gestion du trafic aérien*. PhD thesis, École Polytechnique, Palaseau, France, 2000.
- [22] M. Terrab and A.R.Odoni. Strategic flow management for air traffic control. *Operations Research*, 41(1):pp. 138–152, 1993.