Optimization of aicraft trajectories in North Atlantic oceanic airspace

Olga Rodionova, Mohammed Sbihi, Daniel Delahaye, Marcel Mongeau

Optimization of aicraft trajectories in North Atlantic oceanic airspace

Olga Rodionova, Mohamed Sbihi, Daniel Delahaye, Marcel Mongeau,
Laboratory of Applied Mathematics, Informatics and Automatics for Aviation (MAIAA)
Ecole Nationale de l'Aviation Civile (ENAC)
7 av. Edouard Belin, 31055 Toulouse cedex 4, France

***Abstract*—Air traffic control in non-radar oceanic airspace has always been relatively limited. Current American and European projects aim at transiting from ground-based systems to satellite-based systems. New technologies will allow to reduce significantly the present aircraft separation standards. As a consequence, the aircraft crossing the North Atlantic will be able to follow better routes, which will improve the air traffic situation by decreasing of flight durations and congestion in pre-oceanic continental airspace. In this paper an optimization model is introduced and a simple methodology to solve this problem is proposed that yields encouraging preliminary results.**

Keywords-Aircraft trajectory optimization; North Atlantic oceanic airspace (NAT); Automatic Dependent Surveillance-Broadcast (ADS-B); Simulated Annealing

I. INTRODUCTION

A. North Atlantic airspace

The North Atlantic (NAT) is the busiest oceanic airspace in the world. For the most part in the North Atlantic, Air Traffic Services (ATS) radar surveillance is unavailable, and Air Traffic Control (ATC) must rely significantly upon the aircraft HF voice position reports. Aircraft separation assurance and safety are nevertheless highly demanded.

Due to the constraints of large horizontal separation criteria and a limited economical flight level (FL) band (FL310-400), the airspace is congested at peak hours. In order to provide the best traffic control service, a system of tracks referred to as the *Organized Track System* (OTS) is constructed to accommodate as many flights as possible on, or close to, their minimum time tracks and altitude profiles [1].

As a result of passenger demand and time zone differences, much of the NAT air traffic contributes to two major alternating flows: westbound and eastbound, whose effect is to concentrate most of the traffic unidirectionally. Due to the meteorological conditions, including the presence of jet streams, consecutive eastbound and westbound minimum-time tracks are seldom identical. The creation of a different OTS is therefore necessary for each of the major flows. In general the east-west tracks are situated more northerly than the west-east tracks (Figure 1).

Currently, about half of NAT flights are planned on the OTS [1]. OTS flights operate on great circle tracks joining successive significant waypoints and are commonly planned so that the specified ten degrees of longitudes (10°W, 20°W, etc.) are crossed at integer degrees of latitude. The aircraft may be planned at any of the flight levels published for that track. Unless otherwise requested by ATC, the flights on tracks should report their position at the designated reporting points listed in the flight plan.

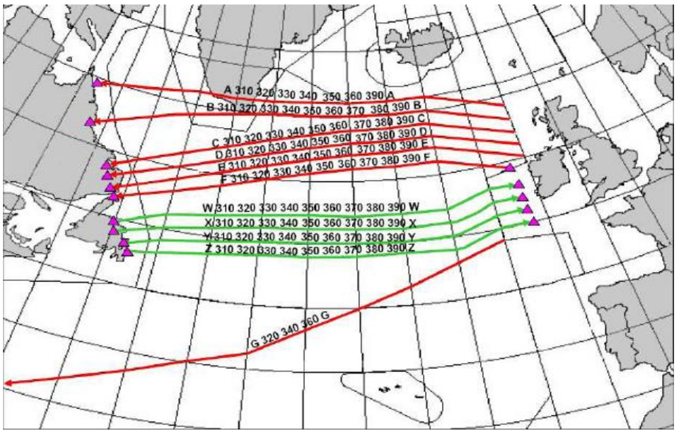

Figure 1. *North Atlantic oceanic airspace with Organized Track System*

To mitigate against communication equipment failures and poor propagation conditions, ATC often employs strategic traffic planning, and issues *Oceanic Clearances* (authorization to follow the specified track at a defined speed and flight level). Flights that continue to follow such pre-coordinated strategic oceanic clearances are thereby guaranteed conflict-free progress to oceanic exit. Every effort is therefore made to clear aircraft as per their flight plans. However, this is not always possible, especially during peak traffic flow periods.

B. Separation standards

The current situation shows that the traffic often follows a routing which is not optimal in terms of flow management from an overall ATC point of view [2], and therefore some re-routing is necessary when entering continental radar control areas. This leads to increasing length and duration of the flights as well as to increasing the congestion in the continental airspace. It seams reasonable to enable the re-routing of aircraft directly within the OTS-covered area from the initial predefined track to another oceanic exit point that is more suitable in view of its final destination. This would reduce ATC complexity in continental airspace near the exit of the track system, and improve the aircraft's remaining route towards destination.

A major factor in preventing traffic re-routing with current oceanic separation standards is the additional longitudinal separation required for flights which have not reported over a common point. Two consecutive aircraft following the same track should be separated 10 minutes apart (Figure 2), as their longitudinal relationship is established by their position reports and any errors in forward position estimates can be assumed to cancel out since they both experience the same weather. But this assumption can not be applied in the case where a particular aircraft desires to re-route onto the adjacent track (Figure 2), as it has experienced weather different from that of any aircraft following this track. As a consequence, current regulations impose an increased longitudinal separation of 15 minutes in this case. Due to the high density of traffic on tracks, this re-routing maneuver can therefore rarely be applied.

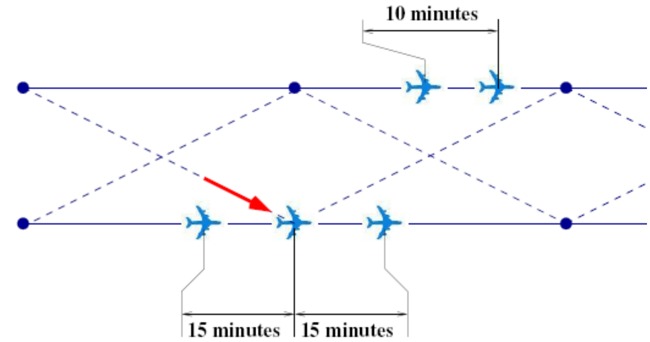

Figure 2. *Aircraft longitudinal separation criterion on the tracks*

The transition from present ATC tools to airborne-based systems and procedures proposed by the American and European projects NextGen (Next Generation Air Transport System) and SESAR (Single European Sky ATM Research) can help to overcome these drawbacks. The key component of these projects is the *Automatic Dependent Surveillance-Broadcast* (ADS-B) technology [3] that combines aircraft's positioning source, aircraft avionics, and a ground infrastructure to create an accurate surveillance interface between aircraft and ATC. Aircraft transponders receive GPS signals and use them to determine the aircraft's actual position. These and other data are then broadcast to other aircraft and to ATC (Figure 3).

The new approach will allow to decrease significantly the oceanic separation standards [4]. Indeed, with ADS-B systems the consecutive aircraft following the same track would be separated only 2 minutes apart, and the aircraft performing a re-routing to the adjacent track would be separated only 3 minutes from the aircraft on its new track. Obviously, the new separation standards will raise the limit on the total number of flights allowed for crossing the NAT airspace, and generally improve the aircraft routes by decreasing their length.

Several papers [5, 6, 7, 8] are devoted to treating the related problem in the Pacific oceanic airspace. However, to our knowledge, no research works have been published concerning the NAT airspace. This paper describes a preliminary study aiming at modeling the improvement of oceanic air traffic situation when using the ADS-B technology.

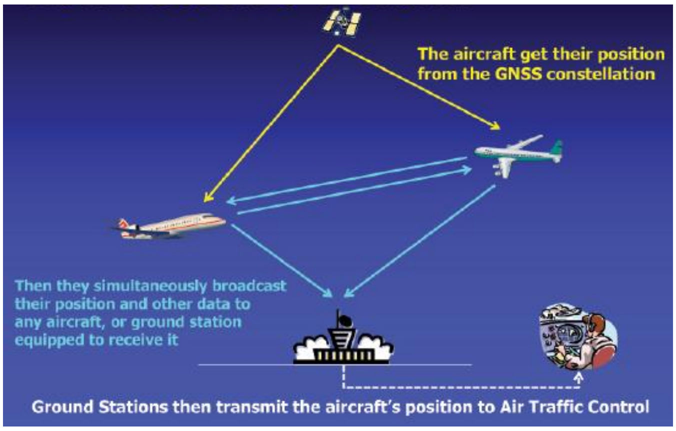

Figure 3. *Automatic Dependent Surveilllance-Broadcast operating principle*

II. PROBLEM FORMULATION

A. OTS modeling

Our aim is to provide a mathematical formulation of the problem consisting in searching for optimal flight routes for a set of flights within OTS. As the subject of the study is the NAT airspace, only the parts of the routes belonging to the OTS will be taken into account. Moreover, as the eastbound and westbound traffics are well separated in time and space, they can be considered independently. Thus, only the eastbound traffic (departing from North America and arriving in Europe) will be taken into consideration here. Obviously, the westbound traffic can be treated in the same way.

To define the flight route, first it is necessary to define the OTS structure. The OTS can be represented by a $Nx \times Ny \times Nz$ grid of waypoints, where Ny is the number of OTS tracks, Nx – the number of waypoints on each track, and Nz – the number of flight levels for each track. Thus, the 3D position of an eastbound aircraft located on track j at waypoint i at flight level k is completely specified by the vector (i, j, k).

A flight entering a predefined track at a predefined flight level is supposed to follow this track at the same level until a change of trajectory is made. In this preliminary study such changes are only allowed at the waypoints. Thus, from its current 3D position (i, j, k), the flight has several possibilities to pursue its route:

- to follow the same track at the same flight level,
- to re-route to an adjacent track at the same flight level,
- to change the flight level.

In the first two cases, the flight should follow the straight line connecting the current waypoint to the next one, where the next waypoint can be one of the following (see Figure 4):

- $(i+1, j, k)$ (same track);
- $(i+1, j+1, k)$ (northern reroute);

- (i+1, j–1, k) (southern reroute).

This grid of waypoints represents a network for which *nodes* are waypoints and *links* are route segments connecting such nodes (Figure 4). Some extra nodes have been added for modeling route segment crossings. These extra nodes are considered to be outside the network.

When an aircraft changes its flight level, it is only authorized to climb (not to descend), in order to satisfy as much as possible the optimal kerosene consumption flight profile provided by the company. Moreover, according to the current ATC rules, aircraft may change flight level only when crossing a waypoint (Figure 5). In our model the distance between the real horizontal position of the aircraft at the new flight level and the previous waypoint (see Figure 5) is neglected, as well as the time required to reach the new flight level. This instantaneous-climbing hypothesis is reasonable as the distance between the waypoints is much greater than the distance needed for flight level changing, so the aircraft is supposed to climb almost instantly. Thus, the next aircraft position in this case will be assumed to be (i, j, k+1).

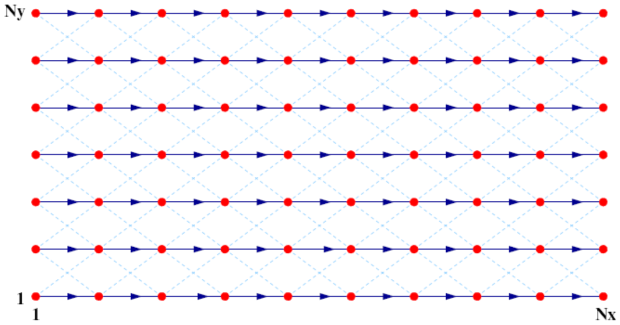

Figure 4. *OTS grid model with nodes and links*

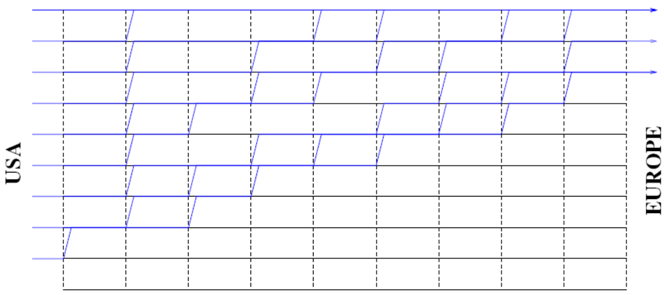

Figure 5. *Flight-level change model on a given track*

B. *Flight Modeling*

In the model we are introducing, each flight is represented by few parameter values. Some of them are fixed data of the problem while others may vary and will represent optimization variables of the problem. These flight parameters are:

- the entry and exit tracks,
- the entry time,
- the speed and flight levels at waypoints,
- the track change positions.

The number of options given to a flight depends on its entry and exit track numbers. For instance, if the entry and exit tracks are the same, the aircraft has no opportunity to change its trajectory. The flight-level changes are fixed according to the optimal performance of the aircraft and happen at some computed waypoint. They are not decision variables.

In this paper a simplified model has been considered. For each flight *f* the following **input data** are given:

- $Tr_{in}(f) \in \{1,2,..., Ny\}$ - the desired entry track;
- $Tr_{out}(f) \in \{1,2,..., Ny\}$ - the desired exit track;
- $t_{in}(f)$ - the desired entry time;
- $v(f) = const(f)$ - the aircraft speed;
- $FL_{in}(f) \in \{1,2,..., Nz\}$ - the entry flight level;
- $FL_{out}(f) \in \{1,2,..., Nz\}$ - the exit flight level $FL_{out}(f) \geq FL_{in}(f)$;
- $z_i(f),\quad i = 1,2,..., Nx-1$ - binary parameters defining the flight altitude profile, that depends on the aircraft weight, and satisfying the following conditions: $$z_i(f) = \begin{cases} 1, & \text{if flight } f \text{ changes its FL at waypoint } i, \\ 0, & \text{otherwise;} \end{cases}$$ $$\sum_{i=1}^{Nx-1} z_i(f) = FL_{out}(f) - FL_{in}(f).$$

For each flight *f*, we define **decision variables** as follows:

- $x_i(f), i = 1,2,..., Nx-1$ - binary variables defining the flight re-routing maneuvers: $$x_i(f) = \begin{cases} 1, & \text{if flight } f \text{ changes track at waypoint } i, \\ 0, & \text{otherwise.} \end{cases}$$ For each flight *f*, these variables form a binary vector of size $Nx-1$, where the number of ones is equal to the distance between actual entry and exit tracks. When $x_i(f)=1$, the aircraft leaves the current occupied track at waypoint i, and reroutes to the adjacent track (the next track towards the exit track).

Based on our first experiments, we have remarked that, for many situations, such a search space definition does not guarantee existence of conflict-free trajectories. In order to avoid this situation, entry and exit track numbers and entry time have been relaxed by allowing aircraft to enter (or to exit) an adjacent track with some delay. The delay (denoted as $d(f)$) can be chosen among a number N_d of discrete-valued multiples of a fixed slot duration (denoted as *slot*). The new associated **decision variables** are:

- $Tr_{in\_act}(f) \in \{1,2,\ldots,Ny\}$ - the actual entry track;
- $Tr_{out\_act}(f) \in \{1,2,\ldots,Ny\}$ - the actual exit track;
- $d(f)$ - the time delay at track entry.

Decision variables must satisfy the following **constraints**:

- $Tr_{in\_act}(f) \in \{Tr_{in}(f)-1, Tr_{in}(f), Tr_{in}(f)+1\}$, (tolerance with regards to required entry track);
- $Tr_{out\_act}(f) \in \{Tr_{out}(f)-1, Tr_{out}(f), Tr_{out}(f)+1\}$, (tolerance with regards to required entry track);
- $d(f) = \delta(f) \cdot slot$, where $\delta(f) \in \{0,1,..., N_d\}$, (acceptable delays);
- $\sum_{i=1}^{Nx-1} x_i(f) = |Tr_{out\_act}(f) - Tr_{in\_act}(f)|$, (total number of re-routing maneuvers).

C. *Objective function*

For a single flight, different route optimality criteria can be chosen, e.g. the total trajectory length, the total flight duration, the fuel consumption, etc. While constructing the optimal routes for a set of flights, we must ensure the additional collision avoidance requirement based on the separation standards.

In this preliminary study, only a collision-avoidance criterion is taken into account in order to simplify modeling, as the aim of this first step of the study is to verify whether a conflict-free solution exists when applying the new separation standards. In future work alternative measure characteristics (such as fuel consumption, flight duration, etc.) will be considered.

Based on the route network structure, conflicts may happen only at nodes and links. The following auxiliary variables are introduced:

- Cf_n - the number of conflicts on nodes;
- Cf_l - the number of conflicts on links.

Thus, in the current model, the objective function is simply given by $(Cf_n + Cf_l)$.

Given an instantiation of decision variables, the aircraft trajectories can be computed on the network based on a flight simulator. Each time an aircraft passes over a node, the passing time is recorded. The same recording process is applied for link entry and exit times.

A conflict is detected on a node if the longitudinal separation constraint is violated for two aircraft passing this node. To calculate the number of conflicts at a given node *n*, all the flights *f* passing this node are selected and sorted according to the time $t_n(f)$ at which they pass this node. To satisfy the longitudinal separation requirements, the condition $t_n(f+1) - t_n(f) > \Delta t_s(f, f+1)$ must be fulfilled for $f = 1,2,..., N_n - 1$, where N_n is the total number of flights passing the node, and $\Delta t_s(f, f+1)$ is the imposed longitudinal separation value depending on the flights' maneuvers. If for any pair of flights *f* and *f* +1 this condition is violated, then a conflict is detected and Cf_n is increased by one.

Based on the sequence order of aircraft at the link entry and exit, it is possible to detect conflict happening on this link. It happens if an aircraft is slower than the one following it on the same track. To compute the number of conflicts at a particular link *l*, all flights passing this link are selected and sorted into two lists according to the time at which they entry and exit this link. Then, the entry and exit lists are compared. If two aircraft are found swapped, then a conflict is detected and Cf_l is increased by the number of flights having different ranks in both lists.

III. OPTIMIZATION METHOD

The problem we have to solve is fully discrete with many decision variables. If we consider *N* flights with a number of discrete delays N_d and an average T_c of track changes among $Nx-1$ transition segments, the associated combinatorics is given by:

$$|S| = \left((N_d + 1) \times \binom{Nx-1}{T_c} \right)^N .$$

For instance, if $N_d = 5$, $T_c = 4$, $Nx = 10$ and $N = 500$, then the number of different potential solutions to be considered is:

$$|S| = \left(6 \times \binom{9}{4} \right)^{500} = 756^{500} .$$

Due to this high combinatorics, we have developed a methodology based on stochastic optimization and implemented the Simulated Annealing (SA) method. This optimization method is a metaheuristic inspired from the thermodynamics theory [9, 10]. It imitates the annealing of the metal, involving heating and controlled cooling. Each point *s* of the search space is analogous to a state of some physical system, and a function *E*(*s*) to be minimized is analogous to the internal energy of the system in that state. The goal is to bring the system, from an arbitrary initial state, to a state with minimal energy (the optimal solution of the problem).

At each step, the SA heuristic considers some neighboring state (solution) *s*' of the current state (solution) *s*, and probabilistically decides between moving the system to the new state *s*' or remaining in state *s*. This probability depends both on the difference between the corresponding function values *E*(*s*) and *E*(*s*') and also on a global parameter *T* (called the *temperature*) that is gradually decreased during the process. The dependency is such that the choice between the previous and current solution is almost random when *T* is large (during the first iterations), but increasingly selects the better solution as *T* goes to zero (as the number of iterations grows). Typically, this step is repeated until the system reaches a state that is considered good enough, or until a given computation budget has been exhausted.

Several notions and parameters are to be defined and adjusted for the SA algorithm:

- the representation of state of the system (solution) *s* from the search space;
- the energy function (the objective function of the minimization problem) $E(s)$;
- the method to generate the neighboring state (solution) from the current one ($s \to s'$);
- the probability of acceptance of the state transition as a function of the temperature $pr_{accept}(s \to s', T)$;
- the initial temperature T_0;
- the law of temperature decreasing;
- the number of transitions performed at a same temperature level;
- the stopping criterion for the algorithm.

In our context, the state of the system will represent the trajectories of the set of *N* flights using *N* vectors, each of which corresponds to a particular flight *f* and contains the decision variables defined above: $\delta(f)$, $Tr_{in\_act}(f)$, $Tr_{out\_act}(f)$, $x_i(f), i = 1,2,..., Nx - 1$ (see Figure 6). The flight trajectory is completely defined by these variables together with the flight predefined data such as the desired track entry time, the flight altitude profile and the aircraft velocity. The energy function is represented by our objective function: the total number of conflicts, $E(s) = Cf_n + Cf_l$.

To generate a neighboring state s' from the current state *s*, the *mutation operator* has to be defined. In the current model, a mutation consists in choosing randomly one flight *f*, and changing randomly some of its parameters. The mutation of re-routing variables involves choosing randomly two variables $x_i(f)$ and $x_j(f)$ having different values (0 and 1), and permuting their values (Figure 7). The goal is to explore widely the search space at high temperatures, and to reduce the search area as the temperature decreases in order to concentrate the search near the optimum. The probability of acceptance of the state transition is defined by the following formula:

$$pr_{accept}(s \to s', T) = \begin{cases} 1, & \text{if } E(s') < E(s); \\ e^{\frac{E(s)-E(s')}{T}}, & \text{if } E(s') \geq E(s) \end{cases}.$$

The initial temperature T_0 is to be adjusted from each particular problem in order to provide acceptance of most of the transitions. The process of the initial temperature setting is analogue to the process of the system heating. It starts with some small temperature value T_0^0 that can be obtained from examining a number *M* of system states. Then, at each step, the current temperature is increased by multiplying it with some value $\beta > 1$: $T_0^i = \beta \cdot T_0^{i-1}$. These steps are repeated until the acceptance probability at the current temperature becomes sufficient ($pr_{accept}(s \to s', T) \geq \mu$, for some user-defined value of μ). The last obtained temperature is then chosen as the initial temperature T_0.

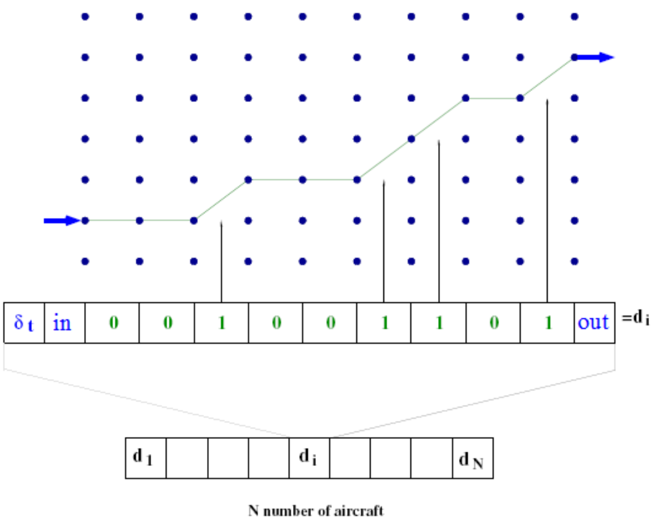

Figure 6. *System state representation*

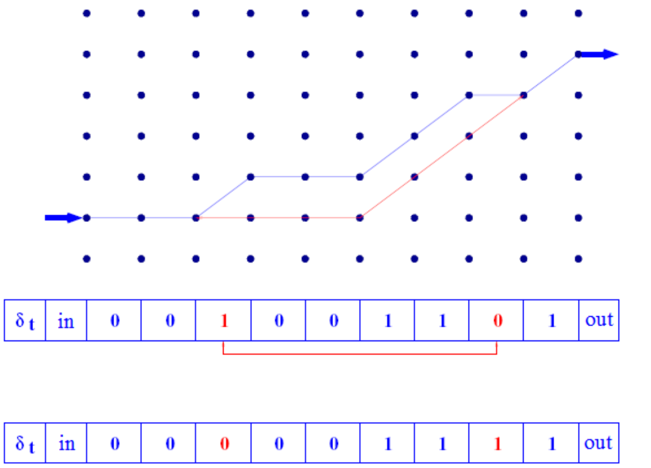

Figure 7. *System state mutation to perfrom transition to a neighbor state*

The process of temperature decreasing is analogue to the system cooling. The cooling should be made rather slow in order to allow the system to reach the thermodynamic equilibrium. The exponential cooling schema has been chosen: $T_i = \alpha \cdot T_{i-1}$, where α is close to but smaller than 1. The number of transitions N_t tested at each particular temperature level was supposed to be constant.

The stopping criterion is to stop either when an optimal (i.e., feasible) solution is found ($E(s) = 0$), or when the temperature goes below the predefined critical value T_f.

IV. Computational Results

This section represents the experimental settings and obtained results in order to validate the proposed optimization formulation and the methodology introduced.

A. *User-defined parameter values for SA*

After the number of tests having been held, the following parameter value were adjusted empirically for SA algorithm:

- $M = 1000$;
- $T_0^0 = \frac{0.01}{M} \cdot \sum_{i=1}^{M} E(s_i)$;
- $\beta = 0.2$;
- $\mu = 0.8$;
- $\alpha = 0.9, 0.95, 0.99$;
- $N_t = 1000$;
- $T_f = 0.0001 \cdot T_0$.

B. *Computational environment*

The SA algorithm was implemented in Java and was run under Windows-32 operational system, on Intel® Core™ 2 CPU with 1.73 GHz.

C. *Test problem description*

To test our algorithm five random aircraft flight sets were generated. An aircraft flight set consists of 500 flights defined by the values $Tr_{in}(f)$, $Tr_{out}(f)$, $t_{in}(f)$, $v(f)$, $FL_{in}(f)$, $FL_{out}(f)$, $z_i(f)$, for $f = 1,2,\ldots,500$, that were randomly generated on a 4-hour time period with realistic altitude profiles and velocities. The aircraft velocity $v(f)$ was chosen in the range from 450 knots to 500 knots.

Aircraft were allowed to select an entry time delay in the range {0, 5, 10, 15, 20, 25, 30} minutes.

The OTS was modeled with a regular grid consisting of $Ny = 7$ tracks each having $Nx = 10$ waypoints and $Nz = 10$ flight levels. The distance between the tracks is 1º, which corresponds to 60 NM (Nautical Miles). The distance between waypoints is 10º, i.e. 300 NM at the latitude of 60º (taken as an example). The distance between the flight levels is 1000 feet (304.8 m).

The tests were performed with two longitudinal separation values: 10 minutes (for the set of aircraft not equipped with ADS-B) and 2 minutes (for the set of aircraft equipped with ADS-B). In the case where an aircraft re-routes from another track, the value of longitudinal separation at the targeted waypoint $\Delta t_s(f, f+1)$ is multiplied by 1.5 as required by the standard (15 minutes and 3 minutes correspondingly). In the case where one of the consecutive aircraft changes its flight level, the value of longitudinal separation between these aircraft $\Delta t_s(f, f+1)$ is multiplied by 1.1, in order to compensate the error caused by neglecting the climbing time.

D. *Results*

Two types of tests were performed on the same generated flight sets. In the first case, it was supposed that the aircraft do not possess the ADS-B system. In the second case, all the aircraft were considered to be equipped with ADS-B.

For the flight sets not equipped with ADS-B it was shown that no collision-free solution exists. This confirms that aircraft have to fly the routes that are not optimal. Different values of the temperature decrease parameter α were tested in this case in order to obtain better solution. The algorithm is able to decrease the total number of conflicts by a factor 3 when compared with the initial number. The results of simulations for flight set 1 are represented in Table I and in Figure 8. The initial number of conflicts in this case was equal to 2775.

TABLE I. COMPARING THE RESULTS OF SA FOR A TEST FLIGHT SET WITHOUT ADS-B

α	**Number of iterations of cooling**	**Best solution, final number of conflicts**
0.9	88	1135
0.95	180	1068
0.99	917	817

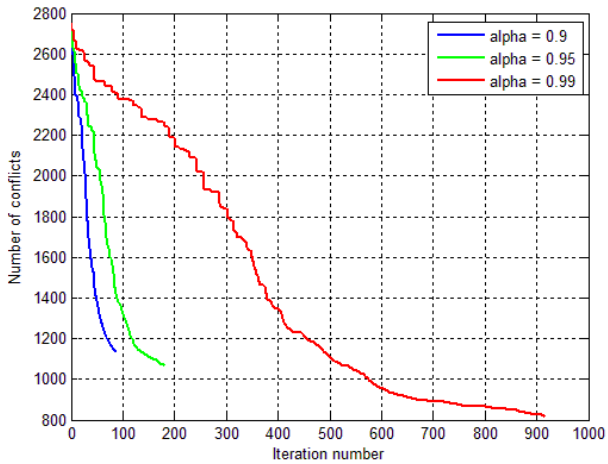

Figure 8. *Evolution of the number of conflicts with the steps of cooling process performed for temperature decreasing for flight set 1 without ADS-B*

For the flight sets where all aircraft are equipped with ADS-B, the SA algorithm finds an optimal (collision-free) solution for every test set under study. The average number of iterations in heating for the 5 tests is 11. The maximum number of decreasing steps during the cooling process that the algorithm can perform with the temperature decrease parameter $\alpha = 0.9$ is 88. For the tests with ADS-B, this number is sufficient, the value $\alpha = 0.9$ is therefore preferred to be used. In this case, the average number of iterations in cooling for the 5 tests is 37. The following graphs (Figures 9 and 10) represent the algorithm progress.

Table II represents the average computational time of algorithm execution for both types of performed test (with and without ADS-B).

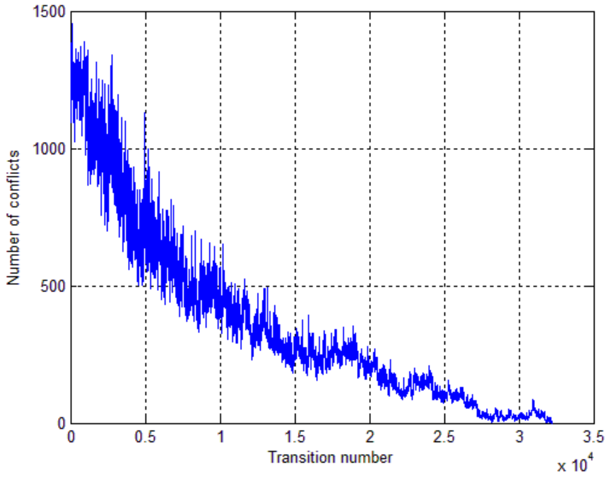

Figure 9. *Evolution of the number of conflicts with transitions performed during the process of cooling for flight set 1 with ADS-B*

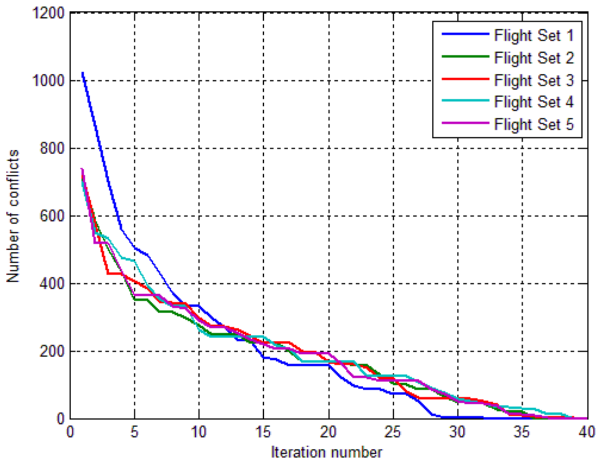

Figure 10. *Evolution of the number of conflicts with the steps of cooling process performed for temperature decreasing for flight sets with ADS-B*

TABLE II. SIMULATED ANNEALING ALGORITHN COMPUTATIONAL TIME

Test	**Computational parameters**	**Computational time, min**
Without ADS-B	Heating process	2
	Total, $\alpha = 0.9$	8
	Total, $\alpha = 0.95$	14.5
	Total, $\alpha = 0.99$	67
With ADS-B	Heating process	4
	Total, $\alpha = 0.9$	7.5

V. CONCLUSION

In this paper, a developed mathematical model of the oceanic air traffic is introduced. Based on the future performances of the ADS-B system, an optimization algorithm has been implemented in order to improve aircraft routes and to reduce the induced congestion at the exit continental airspace. Due to the high complexity of the combinatorial optimization problem, a Simulated Annealing algorithm has been applied.

The performed simulations prove that implementing the ADS-B technology has a strong positive effect on the current traffic situation in North Atlantic oceanic airspace. The reduction of the current separation standards makes it possible for the aircraft to perform re-routing within Organized Track System and therefore, to follow more optimal trajectories towards their destination. As a consequence, the total flight duration as well as the congestion level in the pre-oceanic airspace will significantly decrease.

In the future work it is planned to test the developed approach on real NAT traffic data and to compare the obtained optimized trajectories with the routes actually followed by aircraft, using appropriate aviation-focused metrics. Also it is planned to modify the objective function used in the model in order to take into account other performance measures (such as fuel consumption, flight duration, etc.). Future work could also consider other optimization algorithms to solve the described problem.

ACKNOWLEDGMENT

This work has been supported by French National Research Agency (ANR) through COSINUS program (project ID4CS n°ANR-09-COSI-005).

REFERENCES

[1] NAT Doc 007. Guidance concerning air navigation in and above the North Atlantic MNPS airspace. Edition 2010. World Air Ops online Library.

[2] ASEP-ITM simulations from traffic data. ASSTAR WP8 27/09/2007.

[3] http://www.faa.gov/nextgen/portfolio/trans_support_progs/adsb/

[4] P.L. Massoglia, M.T. Pozesky, G.T. Germana, The Use of Satellite Technology for Oceanic Air Traffic Control. Proceedings of the IEEE, Vol. 77, No. 11, November 1989.

[5] S.C. Mohleji, J. Hoffman, Performance Analysis of North Pacific Operations Using an Automated Air Traffic Control System Simulation. IEEE Transactions on Control Systems Technology, Vol. 1, No. 3, September 1993.

[6] S. Grabbe, B. Sridhar, N. Cheng, Central East Pacific Flight Routing. AIAA Guidance, Navigation, and Control Conference and Exhibition, 21-24 August 2006, Keystone, Colorado.

[7] S. Grabbe, B. Sridhar, A. Mukherjee, Central East Pacific Flight Scheduling. AIAA Guidance, Navigation, and Control Conference and Exhibition, 20-23 August 2007, Hilton Head, South Carolina.

[8] S. Grabbe, B. Sridhar, A. Mukherjee, Scheduling Wind-Optimal Central East Pacific Flights. Air Traffic Control Quarterly, Vol. 16(3) 187-210 (2008).

[9] S. Kirkpatrick, C.D. Gelatt, M.P.G. Vecchi, "Optimization by Simulated Annealing", Science, vol. 220, 671-680, 1983.

[10] E. Aarts, J. Korst, Simulated Annealing and Boltzmann Machine. John Wiley and Sons, 1989.