Optimal Design of SIDs/STARs in TMA Using Simulated Annealing

Jun Zhou, Sonia Cafieri, Daniel Delahaye, Mohammed Sbihi

Optimal Design of SIDs/STARs in TMA Using Simulated Annealing

Jun Zhou, Sonia Cafieri, Daniel Delahaye, Mohammed Sbihi

ENAC, MAIAA, F-31055
Univ. de Toulouse, IMT, F-31400
Toulouse, France
Email: junzhou@recherche.enac.fr
{sonia.cafieri, daniel.delahaye, mohammed.sbihi}@enac.fr

***Abstract*—In this study, we propose a Simulated Annealing (SA) method to solve the problem of designing multiple Standard Instrument Departure (SID) routes and Standard Terminal Arrival Routes (STAR) in Terminal Maneuvering Area (TMA) with optimal lengths. This work extends our previous work (J. Zhou et *al.*, 2015) for the design of one optimal route. The design of multiple routes addressed in this paper takes into account obstacle avoidance and routes separation as main constraints. Our preliminary numerical results show that the proposed SA method is effective in the generation of multiple routes in a TMA.**

I. Introduction

Terminal Maneuvering Area (TMA) is an area surrounding one or more neighboring airports, that is designed to handle aircraft arriving to and departing from the airports. Most of the airports have pre-designed procedures indicating how aircraft depart from or arrive to airports. These procedures are called Standard Instrument Departure (SID) routes and Standard Terminal Arrival Route (STAR). A SID is a flight route followed by aircraft after take-off from an airport until the start of the en-route phase. A STAR is a route which connects the last en-route way-point to the Initial Approach Fix. Currently, SIDs/STARs are designed manually according to operational requirements (ICAO Doc 8168), taking into account airport layout and nearby constraints. However, this kind of design is generally not very efficient and is not expected to optimize any specific criterion. Optimizing departure and arrival procedures in TMA is crucial to regulate air traffic flows whose continuous growth can affect the air traffic operations. The objective of this work is to automatically design SIDs/STARs in 3D, taking into account some constraints including obstacle avoidance and separation between routes, and optimizing the total length of the designed routes. The design is at a strategic level, as only static obstacles are taken into account. Moreover, rather than considering a route for one flight only, we aim at designing routes which are flyable for aircraft of different types and performances. Our method can be regarded as a decision support tool which provides guidelines for the routes design in real TMA, for example in the case of a new built airport.

The ICAO's Performance Based Navigation (PBN) concept is one of the key factors to enable future airspace and traffic flow design [1]. It offers operational benefits such as enhanced safety and increased efficiency. The Required Navigation Performance (RNP), a typical way of navigation within the PBN concept, is especially useful in complex airspaces such as TMAs since it provides higher design flexibility. In fact, unlike the conventional navigation routes which follow ground-based navigation aids instruments, the RNP routes follow a succession of waypoints which are defined by simple latitude and longitude coordinates. Pilot can choose the waypoints in the navigation database implemented in the Flight Management System (FMS). Currently the RNP-1, a type of RNP with specific performance level, enables the Radius-to-Fix (RF) functionality, which can be applied in SIDs/STARs design. An illustration of RF is shown in Fig. 1: it is defined as an arc with specified radius between two defined waypoints in a SID/STAR [1]. In order to ensure flyable routes, the radius of the arcs of circles is imposed to be at least equal to 3 Nm [2] [3]. The design in our study is based on the RNP concept, considering the RF functionality in SIDs/STARs.

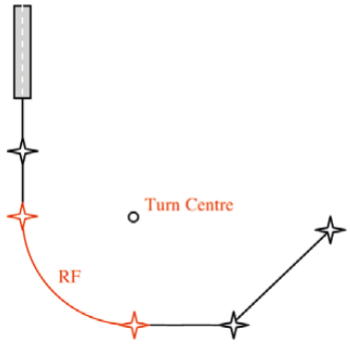

Fig. 1. Radius-to-Fix illustration

The considered problem is in the framework of path planning. Specifically, it is a route design problem: contrarily to trajectory design, we aim at designing routes that are not associated to any notion of time. The problem of path planning has been studied since 1980s especially in the robotic domain [4] [5]. Nowadays planning optimal aircraft paths becomes a rich and dynamic research area, some approaches have

been summarized in [6]. We refer the readers interested by a literature review on designing one optimal route to our previous work [7].

The optimal design of multiple 3D-routes satisfying the numerous constraints of TMAs (routes separation, obstacles avoidance, noise abatement, etc.) is a very complex problem. One of the main difficulties is to find suitable analytic mathematical expressions for some of these constraints, as well as to handle a large number of constraints. Exact methods are therefore difficult to apply in the considered context. This motivates the use of heuristic approaches. Two different strategies are proposed and applied in the literature. The first is a sequential 1-against-n strategy where the routes are generated one after the other according to their priority order, as in [8], [9]. The priority order is defined by the user, for example according to the decreasing traffic load on each route. The previously considered routes become obstacles for the route that will be considered later. The second one is a global strategy where all routes are generated simultaneously in order to minimize a global cost associated to the set of routes, as in [8]. More precisely, in [9] the author designs terminal routes getting around obstacles as well as satisfying the separation criterion using the sequential 1-against-n strategy combined with a modified A* algorithm. Even though the routes are designed in 3D, only horizontal deviation is applied in order to separate them from each other. In [8] static conflict-free routes connecting two airports are built using both a 1-against-n strategy combined with an A* algorithm and a global strategy combined with a Genetic Algorithm (GA). The horizontal deviation is applied only in the climb and descent process and vertical deviation is applied for the remaining part of the route.

In our study, we propose a Simulated Annealing (SA) method in order to identify feasible though not guaranteed optimal solutions, with respect to pairwise route separation and obstacle avoidance, minimizing route lengths. Each route is represented by a horizontal curve associated to a cone in the vertical plan. In order to avoid obstacles and to satisfy separation criterion, both horizontal and vertical deviations are allowed. A vertical deviation is realized by imposing a level flight. Indeed, imposing a level flight is an effective way as it enriches the space of possible maneuvers and corresponds to what is done in the reality in a TMA.

This paper is organized as follows. Section II introduces the routes and obstacles modeling. Section III presents the proposed SA-based approach to solve the multi-routes design problem. Section IV gives some preliminary simulation results. Finally, Section V draws conclusions and proposes future directions.

II. Problem statement and modeling

TMA is one of the most complex types of airspace. Many constraints have to be satisfied, falling into two categories: operational constraints related to air traffic operations (such as obstacle avoidance and separated routes), and environmental constraints (such as noise abatement). SIDs/STARs are designed to satisfy these constraints and to deal with the dense traffic converging to and diverging from airports. The constraints in TMA make the SIDs/STARs design a very complex problem. In a previous study [7], we considered the simpler problem of designing a single route avoiding obstacles and satisfying some other operational constraints. In this work, we use the same models for routes and obstacles as in [7], while considering the more complex problem of designing multi-routes.

Let $N \in \mathbb{N}$ be the total number of routes to be built. Following [7], a 3D-route $\gamma_i, i = 1, \cdots, N$ is defined by two elements: a curve γ_{i_H} in the horizontal plan associated with a cone γ_{i_V} in the vertical plan. The horizontal curve γ_{i_H} is composed by a succession of arcs of circles bypassing obstacles (corresponding to the RF legs in ATM) and segments connecting tangentially two arcs (corresponding to standard point-to-point legs in ATM). Moreover, The tangent points located at the extremities of arcs and segments correspond to the waypoints of this horizontal route under the concept of RNP. A starting point $A_i\ (x_{A_i}, y_{A_i})$ and an ending point $B_i\ (x_{B_i}, y_{B_i})$ are associated to γ_{i_H}. In a SID (respectively, STAR) case, the starting point is at the runway threshold (respectively, Final Approach Fix (FAF)) and the ending point is an exit (respectively, entry) point of a TMA. Note that, the starting and ending points are input data of our problem. The choice of these points are generally related to the runway configuration of the airport as well as the wind direction, but this is not included in the scope of the current work. The horizontal route γ_{i_H} is a smooth mapping defined as:

$$\gamma_{i_H} : [0,1] \to \mathbb{R}^2 \tag{1}$$

where $\gamma_{i_H}(0) = (x_{A_i}, y_{A_i})$ and $\gamma_{i_H}(1) = (x_{B_i}, y_{B_i})$. In a vertical plan, the starting point $A_i\ (x_{A_i}, y_{A_i})$ is associated with an altitude H_{A_i}. The vertical profile γ_{i_V} contains all ascent (or descent) profiles of the aircraft flying on this route, it is defined as:

$$\gamma_{i_V} : \begin{array}{ccc} [0,1] & \to & I^{\mathbb{R}} \\ t & \to & [h_{i_{inf}}(d_i(t)), h_{i_{sup}}(d_i(t))] \end{array} \tag{2}$$

where $I^{\mathbb{R}}$ defines the set of intervals of $\mathbb{R}$, and $d_i(t) = \int_0^t \|\gamma'_{i_H}(s)\|_2 \, \mathrm{d}s$ is the flown distance until t in the horizontal plan, $[h_{i_{inf}}(d_i), h_{i_{sup}}(d_i)]$ is the interval defined by the cross section of the cone at d. Figure 2 illustrates an example of how γ_{i_H} is associated with γ_{i_V} in the case of a SID, where d is the flown distance and $\alpha_{min,TO}$ (respectively, $\alpha_{max,TO}$) is the minimum (respectively, maximum) take-off rate of aircraft on this route.

The obstacles (together with their protection areas), in number of $m \in \mathbb{N}$, are modeled as cylinders in 3D as presented in Fig. 3. Each cylinder $\Omega_j, j = 1, \cdots, m$ is defined by $(C_j(x_j, y_j), r_j, z_{j_{inf}}, z_{j_{sup}})$, where $C_j(x_j, y_j)$ and r_j are the center and the radius of the two bases respectively; $z_{j_{inf}}$ and $z_{j_{sup}}$ are the altitude of the lower and upper bases.

It is known that a 2D shortest route avoiding circular obstacles is composed of segments (connecting tangentially two obstacles) and arcs of circles (lying on the border of

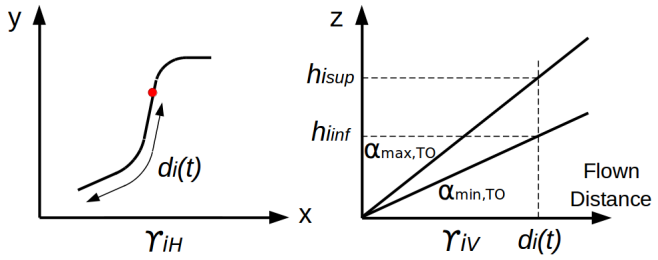

Fig. 2. An Example of γ_H and γ_V

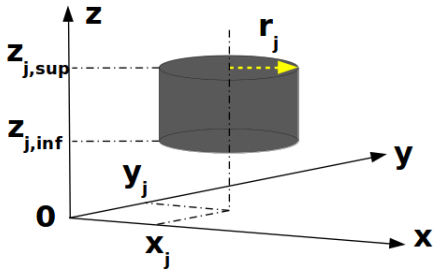

Fig. 3. Obstacle Modeling

obstacles) [10], [11]. This motivates us to search γ_{i_H} in form of a succession of arcs of circles and segments. Furthermore, a 3D route can maintain level flight under obstacles. Thus an obstacle is defined as *active* when it is touched by a route and it has to be avoided according to one of the following maneuvers: turn counter-clockwise, turn clockwise or impose a level flight. For a route γ_i, each cylinder Ω_j is associated with two decision variables s_{ij} and t_{ij}: s_{ij} defines whether Ω_j is active or not on the route γ_i:

$$s_{ij} = \begin{cases} 0, & \text{if } \Omega_j \text{ not active} \\ 1, & \text{if } \Omega_j \text{ active} \end{cases} \tag{3}$$

while t_j defines the ways an active obstacle Ω_j is avoided on the route γ_i:

$$t_{ij} = \begin{cases} 0, & \text{if turn counter-clockwise} \\ 1, & \text{if turn clockwise} \\ 2, & \text{if impose a level flight} \end{cases} \tag{4}$$

Besides obstacle avoidance, the specific separation criterion is also taken into account. The standard separation norm between aircraft in a TMA is 3Nm in the horizontal plan and 1000ft in the vertical plan as shown in Fig. 4. We apply this criterion for separation between routes, so that the separation between aircraft is automatically guaranteed.

Another constraint is the flyability of the designed route segments. As mentioned above, the horizontal curve of a designed route consists a succession of line segments which correspond to point-to-point legs in ATM and arcs of circles which correspond to RF legs of the RNP concept (introduced in Sect.I). Knowing that the radius of a flyable arc in TMA is at least equals 3Nm [2] [3]. A pre-processing is applied on the obstacles: if its radius is lower than 3 Nm, then the radius

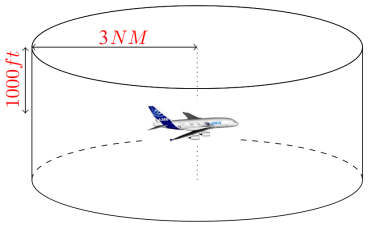

Fig. 4. Standard Separation Norm in TMA

is increased to 3 Nm. Moreover, the vertical profile of each route takes into account the aircraft maximum and minimum take-off (in a SID case) or landing (in a STAR) slopes.

Further constraints are related to level flights. First, the number of level flights on each route is bounded by a maximum number N_{max}, usually fixed to 2, for route γ_i:

$$\sum_{j=1}^{m} \max(t_{ij} - 1, 0) \leq N_{max} \tag{5}$$

Second, as the altitudes of imposed level flights have a direct impact on the noise pollution, a minimum altitude H_{min} for each level flight is defined. In practice, we impose the following constraints: for an obstacle Ω_j, if $z_{j_{inf}} < H_{min}$, then no level flight is imposed below it, therefore $\forall i, t_{ij} \in \{0,1\}$. Third, as to take into account the passengers comfort, the length of each level flight should not be too short, a minimum length L_{min} for each level flight is imposed.

For each route γ_i, we define a weighted sum L_{γ_i} of three terms: the length of γ_i in the horizontal plan, the length related to the level flights and the length of route section interfered in potential conflicts . More precisely:

$$L_{\gamma_i} = c_1 \int_0^1 \|\gamma'_{i_H}(t)\|_2 \, \mathrm{d}t + c_2 L_{min} \sum_{j=1}^{m} \max(t_{ij} - 1, 0) + c_3 \ell_i \tag{6}$$

where ℓ_i is the total length of route sections on route γ_i where the separation criterion is not satisfied. The coefficients c_1, c_2 and c_3 are three penalty parameters, their values are user-defined parameters depending on the importance of the corresponding term. We minimize the sum of $L_{\gamma_i}, i = 1, \cdots, N$:

$$L = \sum_{i=1}^{N} L_{\gamma_i} \tag{7}$$

By taking suitable values for c_1, c_2 and c_3, we aim at finding a solution satisfying the separation criterion between routes ($\sum_{i=1}^{N} \ell_i = 0$), and minimizing the routes lengths at the same time. The obtained problem is a combinatorial optimization problem. In the next section we explain the proposed solution approach for this problem.

III. A Simulated Annealing Method to Design Multiple SIDs/STARs

In order to perform an optimal design of N conflict-free routes avoiding obstacles, where a *conflict* between two routes

is defined as a violation of the minimum separation criterion, we propose the following approach:

- First, generate each route individually by applying the Branch and Bound (B&B) method presented in [7].
- Second, detect pairwise conflicts between the generated routes.
- If the routes are conflict-free, then the solution is already optimal. Else, apply the SA method in order to remove conflicts.

In the rest of this section, we describe the SA method that we propose for the considered problem.

Our problem is a highly combinatorial optimization problem. The SA, proposed by S. Kirkpatrick et al. in 1983 [12] and by V. Cerny in 1985 [13], appears to be particularly suitable for this kind of problems. Its principle is to emulate the physical process whereby a solid is slowly cooled so that when eventually its structure is frozen, a minimum energy configuration is obtained. A flow chart illustrating a SA algorithm is presented in Fig. 5, where the parameters and related notations are

- T_0: the initial temperature
- T: the current temperature
- T_f: the final temperature
- β: the temperature cool-down factor
- S_0: the initial solution
- S_C: the current solution
- S_N: the neighboring solution of the current solution
- S_B: the best solution
- E: the SA fitness function
- ΔE: the degradation of the SA fitness function value
- k: the counter of iterations
- N_I: the criterion for change of the temperature stage

The convergence speed of the SA method depends on the following user-defined control parameters: T_0, β, N_I and T_f, whose values are usually set by empirical adjustments. Readers can refer to [14] for practical suggestions for choosing these parameters. The fitness function to be minimized in the SA process is the objective function Eq.(7). According to the acceptation rule, at high temperature, the probability $\exp(\frac{-\Delta E}{T})$ is close to 1, thus the system probably accepts a worse solution; while at low temperature, $\exp(\frac{-\Delta E}{T})$ is close to 0, thus the system tends to accept only better solution. This mechanism helps to escape from a local minimum.

In the SA method that we propose, the initial solution S_0 is set as the routes generated individually by the Branch and Bound method. The way of generating a neighboring solution S_N is specifically tailored to our problem. The main steps of the generation of S_N are:

- *Step 1:* Detect pairwise conflicts between routes of current solution S_C.
- *Step 2:* Associate a *fictitious* obstacle in a cylinder form corresponding to each conflict zone.
- *Step 3:* For each route create a list of obstacles including all real obstacles and the fictitious obstacles where the route is involved.

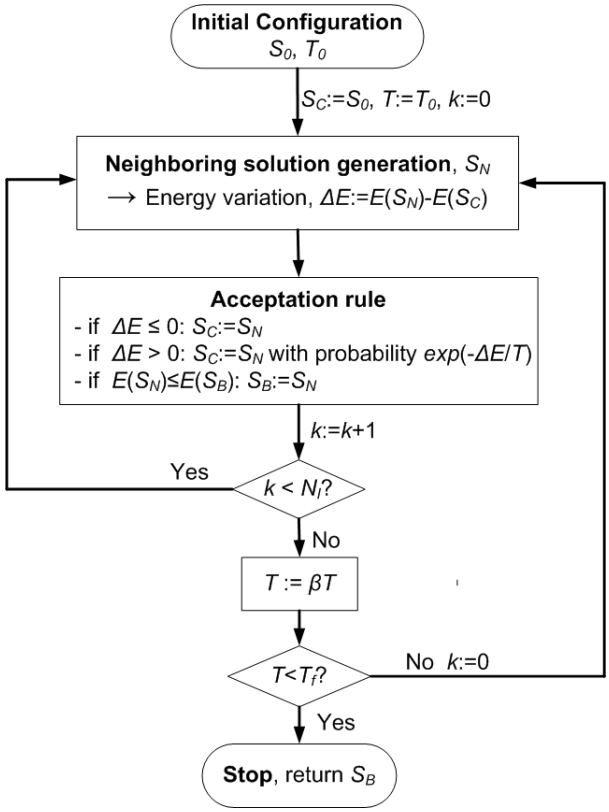

Fig. 5. Simulated Annealing Flow Chart

- *Step 4:* For each route apply a clustering technique to reduce the number of fictitious obstacles and add buffer obstacle near runways to meet operational requirements.
- *Step 5:* For the fictitious obstacles related to each route, choose randomly a strategy to avoid them among the three strategies (counter-clockwise bypassing, clockwise bypassing, imposing level flight).
- *Step 6:* Build new routes according to the chosen techniques and strategies.
- *Step 7:* Post-process the obtained routes and generate the neighboring solution S_N.

Note that, by choosing strategies to avoid the fictitious obstacles, the routes are modified locally around the conflicts zones. Moreover, by applying this process to separate routes, these are modified in such a way to stay close to their initial structure generated by the B&B method. In the following, we present more in detail the generation of a neighboring solution.

A. Step 1: Detecting conflict and computing the length of conflicting sections ℓ_i

Detecting conflicts between 3D-routes, especially when the routes vertical profiles are cones instead of curves, is not an easy problem. Thus we propose a two-steps scheme to deal with this problem. First we detect in the horizontal plan whether pairs of routes lose the 3Nm separation. Afterwards, for the route sections involved in a horizontal conflict, we evaluate whether they are separated in the vertical plan. If a

route section loses both horizontal and vertical separations, then it is in conflict.

1) Detecting in a horizontal plan: We propose a 2D-grid which covers all the horizontal curves $\gamma_{i_H}, i = 1, \cdots, N$ with 3Nm margin at the boundaries. The dimension of a cell in the 2D-grid is 3Nm$\times$3Nm, as defined by the horizontal separation norm. Afterwards we discretize each horizontal curve with a discretization step dt. A post-processing is applied to the discretization so that for each curve there is at least one discretization point in the occupied cells. In this way, each discretization point is associated to a specific cell in a 2D-grid. We define a *curve section* as the section of a curve between two successive discretization points. A *horizontal violation* is detected when the minimum distance between two curve sections (on two different curves) is less than 3Nm. For each occupied cell, we check only the neighboring no-empty cells instead of checking the whole grid, since the horizontal violation with other curves only occurs in the same or neighboring cells. By repeating this operation along $\gamma_{i_H}, i = 1, \cdots, N$, all the violated curve sections are found.

2) Detecting in a vertical plan: Once a horizontal violation is detected, a further check in the vertical plan is needed. Suppose that two curve sections are in a horizontal violation, they are in a *vertical violation* when the minimum distance between the cross section corresponding to their extremities in the vertical plan is less than 1000ft. In fact if two curve sections are separated vertically at their extremities then they are separated along the sections. The reason is that route vertical profiles are monotonously increasing (in a SID case) or decreasing (in a STAR case). Even though this detection method brings an additional margin in the separation, it has the advantage of being simple to implement. Finally, once two curve sections are violated in both horizontal and vertical plan, a conflict is identified.

3) Computing the length of conflicting sections ℓ_i: After that the curve sections which are violated in both horizontal and vertical plan are detected, the length $\ell_i, i = 1, \cdots, N$ is computed by summing up the lengths of these curve sections in potential conflicts on route γ_i.

B. Step 2: Creating Fictitious Obstacles

As mentioned above, each conflict zone is associated to a cylinder-shaped fictitious obstacle, modeled in the same way as presented in Sect. II. In order to generate fictitious obstacles, we define a *violated cell* as a cell containing at least one curve section in a potential conflict. We define two cells are *adjacent* to each other when they have one common edge in x-axis or y-axis. The adjacent violated cells are clustered into the same group. Afterwards, each group of violated cells is associated to a fictitious obstacle, modeled as presented in Fig. 3. The two bases of the fictitious obstacle is the smallest circle contouring all violated cells in the corresponding group in a horizontal plan and the altitude of the lower (respectively, upper) basis is the minimum (respectively, maximum) altitude of the violated curve sections in the corresponding group.

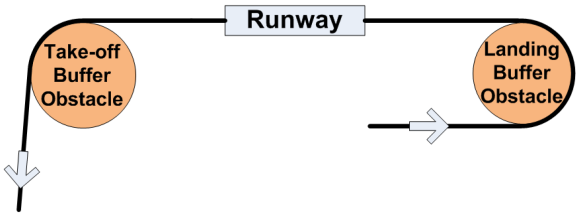

Fig. 6. Buffer Obstacle Illustration

C. Step 4: Clustering fictitious obstacles and shifting runway buffer obstacle

In order to reduce the number of fictitious obstacles and to meet the operational requirements, we also introduce some particular techniques in the SA process to handle the generation of neighboring solution. Two examples are presented in the following.

1) Clustering overlapped fictitious obstacles: The number of fictitious obstacles increases along the SA process. At each iteration, after creating a list of obstacles for each route, if there are overlapping fictitious obstacles in the same list, we cluster them into groups. Then, each group is replaced by a new fictitious obstacle contouring the overlapped ones.

2) Shifting runway buffer obstacle: In order to design routes joining smoothly the following route section from a take-off leg in a SID case, and heading straightly to the runway in a STAR case, we propose the notion of *buffer* obstacles as shown in Fig. 6. Each route γ_i is associated with one buffer obstacle modeled as a cylinder Ω_{bi}, defined by $(C_{bi}(x_{bi}, y_{bi}) + \delta_i \vec{T}_i, r_{bi}, z_{bi_{inf}}, z_{bi_{sup}})$, where $C_{bi}(x_{bi}, y_{bi})$ is the reference center of the buffer obstacle bases, $\delta_i \vec{T}_i$ corresponds to a shift from the referenced center, parallel to the corresponding runway, r_{bi} is the radius of the buffer obstacle bases and $z_{bi_{inf}}$ (respectively, $z_{bi_{sup}}$) is the altitude of the lower (respectively, upper) base. Moreover, since the aircraft have low speed close to runway, they are able to follow a turn with a radius smaller than 3 Nm. The values of those parameters, except δ_i, are given in such a way that the segment connecting tangentially the buffer obstacle and the corresponding runway threshold is parallel to the corresponding runway, so that the designed route joins straightly the runway. The value of the shift step δ_i is randomly chosen within $\{0, 1, 2, 3\}$ in the SA process. Besides, the routes with the same starting point may have the same runway buffer obstacle as well as the buffer obstacle shift step. A buffer obstacle is always active and bypassed by a turn, the turn orientation is also given as an input value with respect to the relative position between the buffer obstacle and the runway. Buffer obstacle shift gives more flexibility when designing routes and it is coherent with the real TMA procedures. Figure 7 illustrates buffer obstacle shift.

D. Step 6: Building a new route

For a route γ_i, once the obstacles avoidance strategies and buffer obstacle shift step are chosen, the active real and

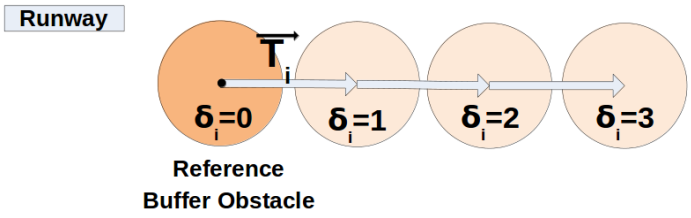

Fig. 7. Buffer Obstacle Shift

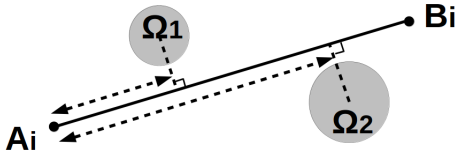

Fig. 8. Obstacles Numbering

fictitious obstacles are numbered in an increasing order of $length(A_i, Proj_{(A_iB_i)}C_j)$ where $Proj_{(A_iB_i)}C_j$ is the projection of C_j onto the line (A_iB_i). An illustration is presented in Fig. 8. The horizontal route is computed by connecting tangentially first the buffer obstacle then the successive active ($s_{ij} = 1$) real and fictitious obstacles which are associated by counter-clockwise ($t_{ij} = 0$) or clockwise ($t_{ij} = 1$) turns in the increasing order of their numbering. Routes are therefore constrained to lie on the border of obstacles. These arcs can be followed using RNP introduced previously. Then, a vertical profile is associated to the horizontal route, taking into account $\alpha_{min,TO}$, $\alpha_{max,TO}$ in a SID case (respectively, $\alpha_{min,LD}$, $\alpha_{max,LD}$ in a STAR case), and imposing a level flight below the active obstacle when $t_{ij} = 2$. If some active obstacle with $t_{ij} = 2$ is not intersected by the cone associated with the horizontal route, then the route is unfeasible regarding to our definition of "active obstacle".

E. Step 7: Post-processing a route

Once a route is built according to the corresponding obstacle avoidance strategies and buffer obstacle shift step, a post-processing is applied. The post-processing first checks whether the route intersects a real obstacle; if so, the real obstacle is set as active and a bypassing strategy is randomly chosen among the three strategies (counter-clockwise bypassing, clockwise bypassing, imposing level flight). Afterwards, the post-processing checks whether the corresponding central angle of an arc on its associated buffer obstacle is more than 180°; if so, the turn orientation is corrected to the opposite direction. By applying the post-processing on all routes from the, the neighboring solution is obtained.

IV. Simulation Results

We implemented the proposed methodology in Java on a Linux platform with a 2.4 GHz processor and 8 GB RAM. The radar data of real traffic arriving to and departing from Paris Charles de Gaulle (CDG) airport during one day is analyzed,

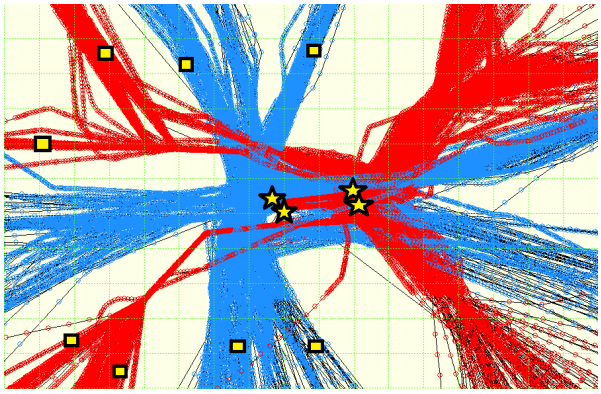

Fig. 9. Traffic of CDG Airport

as shown in Fig. 9, where the departure (respectively, arrival) traffic are in blue (respectively, red) color. The departure and arrival routes are generally located alternately in order to decrease the interaction between them. In the case of CDG TMA, we notice (Fig. 9) that the main conflicts occur between the arrival flows from the north-west and the departure flows to the north, and between the arrival flows from the south-west and the departure flows to the south. We also notice that the arrival flows from the north-east and from the south-east and the departure flows to the west and to the east are not in conflict. Therefore, for the sake of simplification, without loss of generality, in the following we consider only the generation of the 8 routes corresponding to conflicting flows.

The starting and ending points of the routes to be generated in our simulation are presented in Table I (unit in Nm). They are taken according to the real traffic presented in Fig. 9, the starting (respectively, ending) points are marked by stars (respectively, square). Moreover, the corresponding buffer obstacles are presented in Table II. Other input parameters are:

- real obstacle $(x_j, y_j, r_j, z_{j_{inf}}, z_{j_{sup}})$=(86.4Nm, 107.5Nm, 4Nm, 0, 3300ft)
- $H_{A_i} = 0, i = 1, \cdots, 12$
- $z_{bi_{inf}}, z_{bi_{sup}} = (0, 40000\text{ft}), i = 1, \cdots, 12$
- taking-off slope $\alpha_{min,TO} = 5\%$, $\alpha_{max,TO} = 10\%$
- landing slope $\alpha_{min,LD} = 1.6\%$, $\alpha_{max,LD} = 4.8\%$
- $N_{max} = 2$, $L_{min} = 5\text{Nm}$, $H_{min} = 3000\text{ft}$
- SA parameters $T_0 = 40$, $T_f = 5$, $\beta = 0.95$, $N_I = 30$
- objective function penalty parameters $c_1 = 1$, $c_2 = 0$, $c_3 = 100$

The coefficients c_1, c_2 and c_3 are taken in such a way that the length of level flight is not penalized, and since the length of route sections involved in a conflict is crucial to our problem, it is associated with a higher penalization.

The initial routes generated individually by the B&B method as well as the fictitious obstacles corresponding to initial conflicts are illustrated in Fig. 10. At the initial state, the total length in horizontal plan ($\sum_{i=1}^{8} \int_0^1 \|\gamma'_{i_H}(t)\|_2 \, dt$) is 745.2Nm and the length of route sections involved in initial conflicts ($\sum_{i=1}^{8} \ell_i$) is 99.2Nm.

TABLE I
INPUT ROUTES INFORMATION

i	SID/STAR	(x_{A_i}, y_{A_i})	(x_{B_i}, y_{B_i})
1	SID	(99.8, 123.1)	(67.4, 178.5)
2	SID	(99.8, 123.1)	(120, 190)
3	SID	(101.2, 121.5)	(121.4, 67.5)
4	SID	(101.2, 121.5)	(93.4, 67.5)
5	STAR	(112, 124.5)	(19.5, 139.1)
6	STAR	(112, 124.5)	(39, 183.9)
7	STAR	(112.5, 122.5)	(34.1, 59.8)
8	STAR	(112.5, 122.5)	(1.1, 79.8)

TABLE II
BUFFER OBSTACLES

i	(x_{bi}, y_{bi}, r_{bi}) (in Nm)	$\vec{T}_i$	Orientation
1,2	(96.3, 124.9, 2)	(-9.95, -1)	clockwise
3,4	(95.3, 118.7, 3)	(-9.95, -1)	counter-clockwise
5,6	(115.1, 129.6, 4.9)	(9.95, 1)	counter-clockwise
7,8	(116.8, 120.3, 2.5)	(9.95, 1)	clockwise

We run the proposed SA algorithm 50 times with the same input parameters. Table III shows the minimum/maximum/average value of $\sum_{i=1}^{8}\int_0^1 \|\gamma'_{i_H}(t)\|_2 \, dt$ and $\sum_{i=1}^{8}\ell_i$ within the 50 simulation results. The proposed algorithm solves all conflicts ($\sum_{i=1}^{8}\ell_i = 0$) for 44 out of 50 simulations and the average time for each simulation is 157s. Figures. 11-15 illustrate the results corresponding to the simulation associated to the best value of the objective function (total horizontal route length) and conflict-free. In this simulation, the total horizontal route length ($\sum_{i=1}^{8}\int_0^1 \|\gamma'_{i_H}(t)\|_2 \, dt$) is 780.2Nm. Figure. 11 illustrates the final conflict-free configuration in the horizontal plan. Figures. 12-15 illustrate the vertical profile of the 4 SIDs. Notice that, no level flight is imposed on the STARs. This preliminary result shows that the proposed SA method is effective on real TMA example.

V. CONCLUSION AND PERSPECTIVES

In this paper, we introduce a methodology for generating 3D SIDs/STARs in TMA at strategic level, performed by a

TABLE III
SA RESULTS

	$\sum_{i=1}^{8}\int_0^1 \|\gamma'_{i_H}(t)\|_2 \, dt$ (in Nm)	$\sum_{i=1}^{8}\ell_i$ (in Nm)
Initial value	745.2	99.2
Minimum value	780.2	0
Maximum value	1206.5	10.45
Average value	963.1	0.56
Standard deviation	115.3	1.9

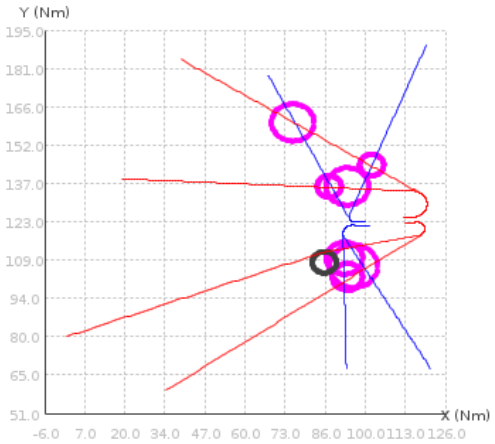

Fig. 10. Horizontal profiles of 8 routes in CDG generated individually by B&B (initial solution for the SA). The projection of a fictitious (respectively, real) obstacle in the horizontal plan is in pink (respectively, black) color.

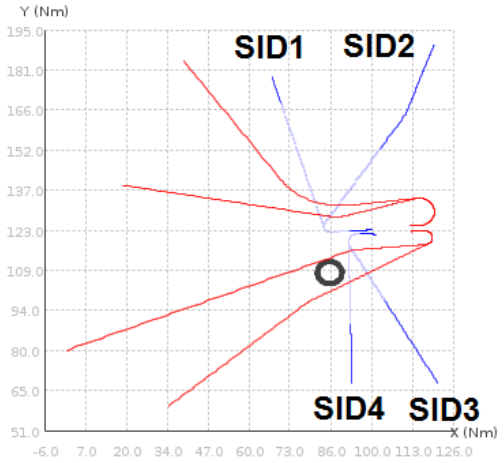

Fig. 11. Results obtained by the SA: horizontal profiles of 8 routes in CDG. The light blue color correspond to the sections where level flights are imposed

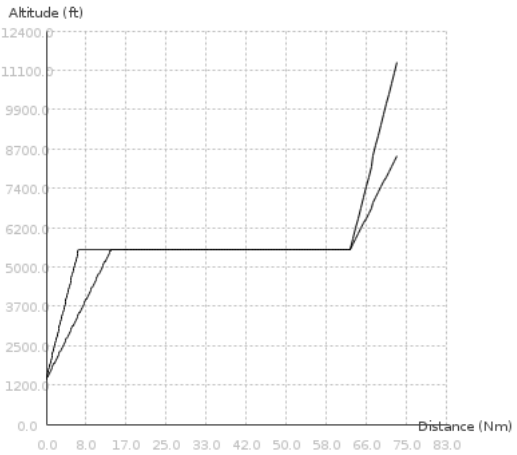

Fig. 12. SID1 in the Vertical Plan

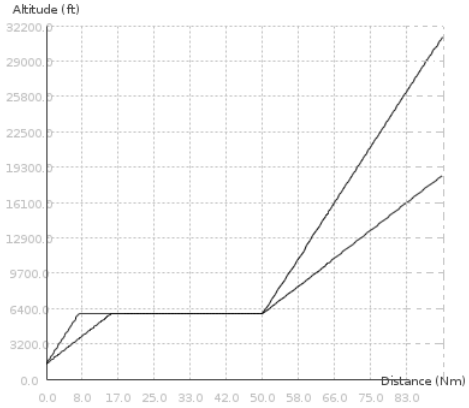

Fig. 13. SID2 in the Vertical Plan

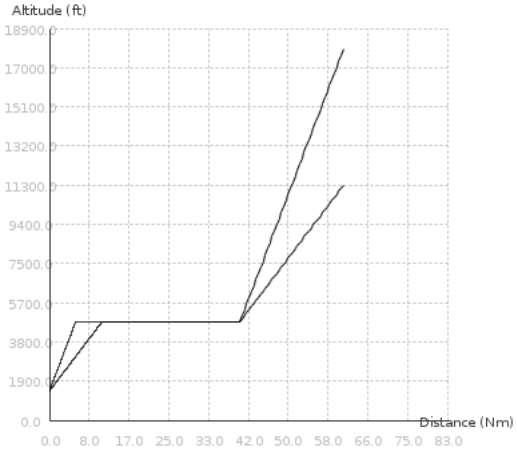

Fig. 14. SID3 in the Vertical Plan

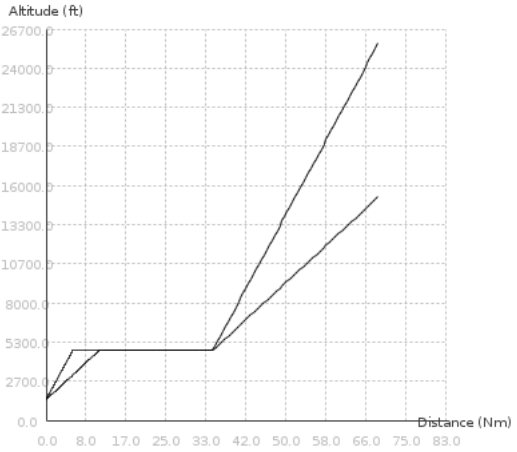

Fig. 15. SID4 in the Vertical Plan

stochastic global optimization approach, namely, a SA algorithm. The algorithm was tested on 8 routes of the CDG TMA. The result shows that the proposed SA method is effective to solve the conflict between routes. In reality the procedure design often considers a variety of constraints that are not completely included in this work. Thus the proposed approach can be regarded as an automatic decision support tool to provide preliminary route structures as guidelines for the manual design. In future work, we will consider to solve the same problem of designing several routes using a Branch and Bound based deterministic method, and compare the result with the one obtained in this work.

VI. ACKNOWLEDGEMENT

This work has been partially supported by Civil Aviation University of China, by National Natural Science Foundation of China (NNSFC) through grant NNSFC 61201085 and by French National Research Agency (ANR) through grant ANR 12-JS02-009-01 ATOMIC. The authors would like to thank Serge Roux from the French Civil Aviation University (ENAC), for the interesting discussions and useful comments.

REFERENCES

[1] Eurocontrol, "European Airspace Concept Handbook for PBN implementation."
[2] FAA, "FAA Order 8260.54A."
[3] ——, "FAA Order 8260.58."
[4] P. Gallina and A. Gasparetto, "A technique to analytically formulate and to solve the 2-dimensional constrained trajectory planning problem for a mobile robot," *Journal of Intelligent and Robotic Systems*, vol. 27, no. 3, pp. 237–262, 2000.
[5] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, and P. Feyzeau, "Path planning: A 2013 survey," in *Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference on*, Oct 2013, pp. 1–8.
[6] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Feron, "Mathematical models for aircraft trajectory design: A survey," in *Air Traffic Management and Systems: Selected Papers of the 3rd ENRI International Workshop on ATM/CNS (EIWAC2013)*. Tokyo: Springer Japan, 2014, pp. 205–247.
[7] J. Zhou, S. Cafieri, D. Delahaye, and M. Sbihi, "Optimizing the design of a route in terminal maneuvering area using branch and bound," in *2015 ENRI International Workshop on ATM/CNS (EIWAC2015)*, Tokyo, Japan, Nov. 2015.
[8] D. Gianazza, N. Durand, and N. Archambault, "Allocating 3D-trajectories to air traffic flows using A* and genetic algorithms," in *CIMCA 2004, international conference on Computational Intelligence for Modelling, Control and Automation*, Gold Coast, Australia, Jul. 2004.
[9] D. M. Pfeil, "Optimization of airport terminal-area air traffic operations under uncertain weather conditions," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.
[10] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
[11] V. Polishchuk, "Generating Arrival Routes with Radius-to-Fix Functionalities," 2016.
[12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *SCIENCE*, vol. 220, no. 4598, pp. 671–680, 1983.
[13] V. Cerny, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.
[14] J. Dreo, A. Petrowski, P. Siarry, and E. Taillard, *Metaheuristics for Hard Optimization*. Springer, 2006.