

[EN-A-012] Optimizing the design of a route in Terminal Maneuvering Area using Branch and Bound

⁺ J. Zhou S. Cafieri D. Delahaye M. Sbihi

MAIAA Laboratory
Ecole Nationale de l'Aviation Civile (ENAC)
Toulouse, France

junzhou@recherche.enac.fr
[sonia.cafieri | daniel.delahaye | mohammed.sbihi]@enac.fr

Abstract: The sharp increase in air traffic flow causes traffic congestion in airspaces near airports, called Terminal Maneuvering Areas (TMA). The departure and arrival traffic of airports follow pre-designed routes named Standard Instrument Departure (SID) routes and Standard Terminal Arrival Routes (STAR). Optimizing these routes is crucial to regulate air traffic. Currently, SIDs and STARs are designed manually, based on the airport layout and nearby constraints. The objective of this research is to propose a methodology for designing an arrival/departure route in TMA, taking into account some constraints including obstacle avoidance. The shape of a route in horizontal plan is a succession of arcs of circles and segments. The originality of our study is, on the one hand, that the horizontal route is associated with a cone in vertical plan enveloping all ascent (or descent) aircraft profiles, and on the other hand, a branching strategy in a Branch and Bound (B&B) framework tailored on the problem is proposed.

Keywords: TMA, SID/STAR design, obstacle avoidance, modeling, Branch and Bound

1 INTRODUCTION

The continuously increase in air traffic flow density causes traffic congestion in the areas surrounding airports, thus affecting the normal operation of air traffic. An area surrounding one or more neighboring airports is called Terminal Maneuvering Area (TMA), and it is designed to handle aircraft arriving to and departing from the airports. Optimizing departure and arrival procedures in TMA is therefore crucial to regulate air traffic flows. 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) and Standard Terminal Arrival Route (STAR). Currently, SID/STAR 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 not expected to optimize any specific criterion. The objective of this work is to automatically design SID/STARs in 3D with respect to certain optimization criteria. Being this study at a strategic level, only static obstacles are taken into account. Rather than considering an individual flight, we deal with flows of flights, that is to say the flights following the same SID (or STAR) belong to the same flow.

The considered problem is in the framework of path planning. Specifically, it is a route design problem: as contrarily to trajectory design, we aim at designing routes that are not associated to the notion of time. The problem of path planning has been studied

since 1980s in the robotic domain [1] [2]. Nowadays planning optimal aircraft paths becomes a rich and dynamic research area, some approaches have been summarized in [3]. In particular, computing the shortest path between two points, given a number of obstacles, is one of the most extensively studied topic [4] [5] [6] [7] [8]. Most of these works search for the shortest path in a horizontal plan, while in our study we consider searching for the shortest path in 3D. Moreover, in several works the obstacle is modeled as polygon [4] [5] [6], and in a few works as circle [7] [8]. In the present study we model the obstacle as cylinder in 3D and the projection to the horizontal plan is in form of circle.

Specifically in aircraft paths designing domain, even though there is a large number of researches, routes design in TMA is a particular problem for which to our knowledge there is not a rich literature. In [9] the author designs terminal routes getting around obstacles with a modified A* algorithm. In our study, we design a route not only getting around obstacles, but also allowing level flights. Indeed, imposing a level flight in vertical plan is also an effective way to avoid obstacles, as it enriches the possible maneuvers and corresponds to what is done in the reality in a TMA. The different ways to avoid obstacles allow us to define specific branching strategies in a Branch and Bound tailored on the problem, that is another contribution of this work.

This paper is organized as follows. Section 2 describes the problem and introduces the route and obstacles modeling. Section 3 presents the proposed ap-

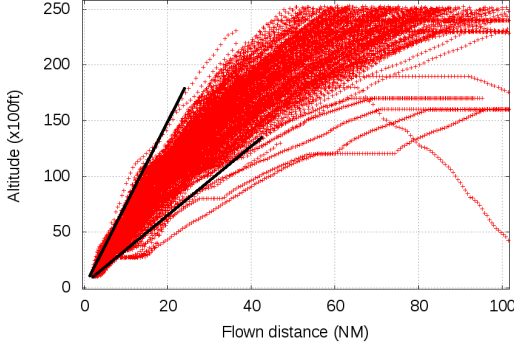
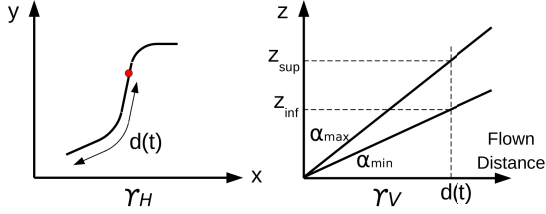


Figure 1 Take-off Profiles in CDG Airport

Figure 2 An Example of γ_H and γ_V

proach to solve the problem. Section 4 gives some preliminary simulation results. Finally, Section 5 draws conclusions and proposes future directions.

2 PROBLEM DESCRIPTION AND MODELING

2.1 Problem Description

TMA (also called Terminal Control Area in the U.S. and Canada) is a control area surrounding one or more airports [10]. It is designed to handle aircraft arriving to and departing from airports. TMA is one of the most complex types of airspace. Many constraints have to be satisfied, falling into two categories: operational constraints, to ensure safe air traffic operations (such as obstacle avoidance), and environmental constraints (such as noise abatement).

SID/STARs are designed to satisfy these constraints and to deal with the dense traffic converging to and diverging from airports. A SID is a flight route followed by aircraft after take-off from an airport until the start of en-route phase. A STAR is a route which connects the last en-route way-point to the Initial Approach Fix. Each aircraft flying under Instrument Flight Rules (IFR) through the TMA must follow a SID when departing from an airport, and a STAR when arriving to an airport.

The constraints in TMA make the SID/STAR design a very complex problem. Therefore in this study we consider the simpler subproblem of designing a single route avoiding obstacles and satisfying some

other operational constraints. The obstacles in TMA could be mountains, cities, military area, etc. In the following, we present the way we model routes and obstacles.

2.2 Problem Modeling

A 3D route γ is defined by two elements: a curve γ_H in a horizontal plan which is composed by a succession of arcs of circles (to bypass obstacles) and segments (to connect tangentially two arcs); a cone γ_V in a vertical plan that contains all ascent (or descent) profiles of the aircraft flying on this route. The cone is defined by two straight lines whose slopes are the minimum and maximum values of the take-off (or landing) rate of the aircraft on this route. The idea of taking a cone that contains all vertical profiles is inspired by the behavior illustrated in Fig. 1, which shows some real take-off data in runway 08L of Paris CDG airport. From the figure we can see clearly that the vertical profiles are contained in a cone defined by two straight lines. The vertical profiles for landing is similar. This behavior is mainly due to the different aircraft mass and performance and to the effect of the wind.

In a horizontal plan, we define a starting point $A(x_A, y_A)$ and an ending point $B(x_B, y_B)$. In a SID case, the starting point is at the runway threshold and the ending point is an exit point of a TMA. In a STAR case, the starting point is an entry point of a TMA and the ending point is the Final Approach Fix. The horizontal route γ_H is a smooth mapping defined as:

$$\gamma_H : [0, 1] \rightarrow \mathbb{R}^2$$

where $\gamma_H(0) = (x_A, y_A)$ and $\gamma_H(1) = (x_B, y_B)$. In a vertical plan, γ_V is defined as:

$$\gamma_V : \begin{array}{ll} [0, 1] & \rightarrow I^{\mathbb{R}} \\ t & \rightarrow [z_{inf}(d(t)), z_{sup}(d(t))] \end{array}$$

where $I^{\mathbb{R}}$ defines the set of intervals of $\mathbb{R}$, and $d(t) = \int_0^t \|\gamma'_H(s)\|_2 ds$ is the flown distance until t in horizontal plan, $[z_{inf}(d), z_{sup}(d)]$ is the interval defined by the cross section of the cone at d . Fig. 2 illustrates an example of how γ_H is associated with γ_V in the case of a SID, where α_{min} and α_{max} are the minimum and maximum values of take-off rate of aircraft on this route.

In the case of a SID, the starting point A is associated with an altitude z_A ; the ending point B is associated with an altitude interval $[z_B, \bar{z}_B]$ which indicates the altitude interval to exit TMA. Therefore the boundary conditions are $\gamma_V(0) = [z_A, z_A]$ and $\gamma_V(1) \subset [z_B, \bar{z}_B]$. Similarly, in the case of a STAR, the altitude interval of the starting point A is $[z_A, \bar{z}_A]$ which indicates the interval to enter TMA; the altitude of the ending point B is z_B . The boundaries conditions are: $\gamma_V(0) = [z_A, \bar{z}_A]$ and $\gamma_V(1) = [z_B, z_B]$.

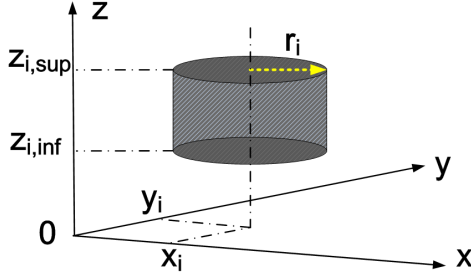


Figure 3 Obstacle Modelization

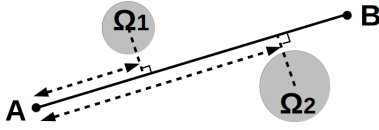


Figure 4 Numbering Obstacles

The obstacles, in number of $m \in N$, are modeled as cylinders in 3D, whose bases are parallel to the horizontal plan as presented in Fig. 3. Each cylinder Ω_i , $i = 1, \dots, m$ is defined by $(C_i(x_i, y_i), r_i, z_{i,inf}, z_{i,sup})$, where $C_i(x_i, y_i)$ and r_i are the center and the radius of the two bases respectively; $z_{i,inf}$ and $z_{i,sup}$ are the altitude of the lower and upper bases. These obstacles are numbered in an increasing order of $length(A, Proj_{(AB)}C_i)$ where $Proj_{(AB)}C_i$ is the projection of C_i onto the line (AB) . An illustration is presented in Fig 4.

Let us define an *active* obstacle as an obstacle that is touched by a route and has to be avoided according to one of the following maneuvers: turn counter-clockwise, turn clockwise and impose a level flight. Each cylinder Ω_i is associated with two decision variables s_i and t_i : s_i defines whether Ω_i is active or not:

$$s_i = \begin{cases} 0, & \text{if } \Omega_i \text{ not active} \\ 1, & \text{if } \Omega_i \text{ active} \end{cases} \quad (1)$$

while t_i defines the ways an active obstacle Ω_i is avoided:

$$t_i = \begin{cases} 0, & \text{if turn counter-clockwise} \\ 1, & \text{if turn clockwise} \\ 2, & \text{if impose level flight} \end{cases} \quad (2)$$

Once the values of decision variables are chosen, the horizontal route is computed by connecting tangentially the successive active ($s_i = 1$) obstacles with $t_i = 0$ or $t_i = 1$ in the increasing order of their numbering. Active obstacles with $t_i = 0$ are bypassed counter-clockwise and those with $t_i = 1$ are bypassed clockwise. This horizontal route is hence built piecewise: it is composed by $(\sum_{i=1}^m s_i - \sum_{i=1}^m \max(t_i - 1, 0) + 1)$ straight line segments and $(\sum_{i=1}^m s_i - \sum_{i=1}^m \max(t_i - 1, 0))$ arcs of circles. Then, a vertical profile is associated to the horizontal route, taking into account α_{min} ,

α_{max} , and imposing a level flight below (respectively above) the active obstacle Ω_i in a SID (respectively STAR) case when $t_i = 2$. If some active obstacles with $t_i = 2$ is not intersected by the cone associated with the horizontal route, then the route is unfeasible regarding to our definition of "active obstacle". Please note that the way of building a horizontal route simplifies the computation but it does not necessarily lead to the shortest horizontal route between A and B.

An illustration of different values of the decision variables for an example of a SID in a TMA with one obstacle is presented in Fig 5. In (Fig 5(a)), the obstacle is not active, so $s_i = 0$. The horizontal route is a straight line segment connecting A and B. It is associated with a cone in the vertical plan. This route in the considered example is not a feasible one, because it intersects the obstacle. Then when the obstacle is active ($s_i = 1$), 3 possibilities are considered to avoid it (Fig 5(b),(c),(d)): turn counter-clockwise, turn clockwise and impose a level flight under the obstacle at altitude $z_{i,inf}$, corresponding to $t_i = 0, 1, 2$ respectively.

Two examples of SIDs are presented in Fig. 6 showing how a route is computed, given the values of the decision variables, in the case with more than one obstacle. In example 1 (Fig. 6(a),(b)), given $(s_1, t_1) = (1, 0)$, $(s_1, t_1) = (1, 1)$, the horizontal route is composed by five parts: three segments and two arcs of circles. The three segments are used to connect tangentially the starting point to Ω_1 , Ω_1 to Ω_2 and Ω_2 to the ending point respectively. The two arcs are used to bypass Ω_1 counter-clockwise and Ω_2 clockwise respectively. In example 2 (Fig. 6(c),(d)), given

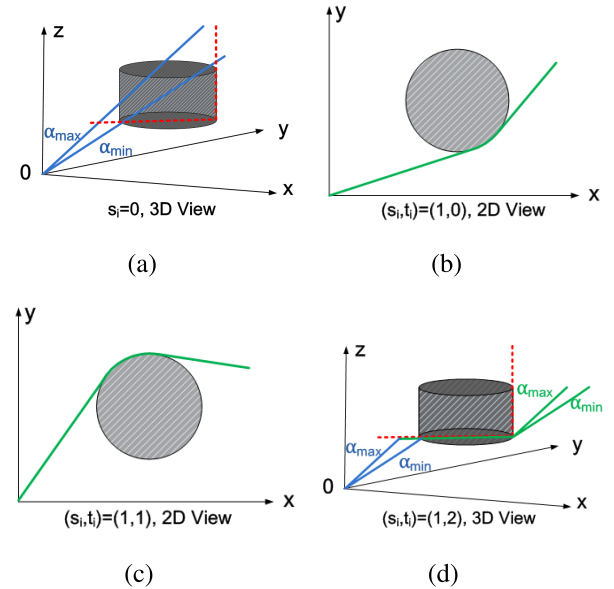


Figure 5 The Routes Associated to Different Values of the Decision Variables

$(s_1, t_1) = (1, 2), (s_1, t_1) = (1, 1)$, the horizontal route is constructed by only bypassing Ω_2 , thus it is composed by two segments and one arc of circle. In vertical plan, when the route reaches the altitude of the lower basis of Ω_1 , a level flight is imposed. The level flight ends at the flown distance where the horizontal route passes the boundary of Ω_1 and is no more intersected by Ω_1 .

Besides obstacle avoidance, further constraints are related to level flights. First, the number of level flights is bounded by a maximum number N_{max} for each route:

$$\sum_{i=1}^m \max(t_i - 1, 0) \leq N_{max} \quad (3)$$

N_{max} is a known value, usually fixed to 2.

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: in a SID case (respectively a STAR case), for an obstacle Ω_i , if $z_{i_{inf}} < H_{min}$ (respectively $z_{i_{sup}} < H_{min}$), then no level flight is imposed below (respectively above) it, therefore $t_i \in \{0, 1\}$.

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.

We minimize a weighted sum L_γ of the length of the route γ in the horizontal plan and the length related to the level flights. More precisely:

$$L_\gamma = c_1 \cdot \left(\int_0^1 \|\gamma'_H(t)\|_2 dt \right) + c_2 \cdot L_{min} \cdot \sum_{i=1}^m \max(t_i - 1, 0) \quad (4)$$

The coefficients c_1 and c_2 are two penalty parameters, their values depend on the importance of the corresponding term. The obtained problem (denoted by $(\mathcal{P})$) is a combinatorial optimization problem. In the next section we explain the proposed solution approach for this problem.

3 SOLUTION APPROACH: BRANCH AND BOUND

To solve the problem, we apply a Branch and Bound method. In [11], a path planning problem avoiding circular obstacles is studied in 2D. A branching strategy is proposed, where, for each obstacle, two branches are created depending on the clockwise or counter-clockwise obstacle bypassing. We extend this branching strategy to take into account the specificity of our problem, where obstacles can be avoided also by imposing a level flight above (SID case) or below (STAR case) the obstacle. Our branching strategy is illustrated in Fig. 7. We start by setting Ω_i as active ($s_i = 1$) or not ($s_i = 0$); when it is active, we develop three branches in order to account for the 3 possibilities of avoiding it: counter-clockwise ($t_i = 0$), clockwise ($t_i = 1$) or imposing a level flight ($t_i = 2$).

The lower bound for each subproblem is computed by generating the route which bypasses only the active obstacles and by calculating its length according to the objective function (4). The obtained lower bound is then used to identify whether a branch requires further subdivisions.

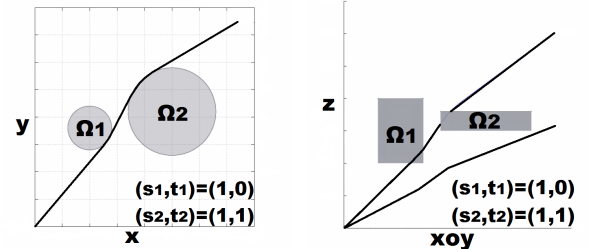
We present a step-by-step illustration (Fig. 8) to show how the Branch and Bound method works. The starting and ending points as well as two obstacles Ω_1, Ω_2 are presented in Fig. 8(a). We take $c_1 = 1, c_2 = 1$, that is to say we penalize the length of level flights in the objective function.

Step 1: We develop 4 branches on Ω_1 . We start by deviating the route counter-clockwise, and we obtain a route that does not intersect Ω_2 . The lower bound in this case is 100320m. Besides, the value of the objective function associated with solution $(s_1, t_1) = (1, 0)$,

$s_2 = 0$ is equal to the lower bound. Therefore, no further exploration is needed.

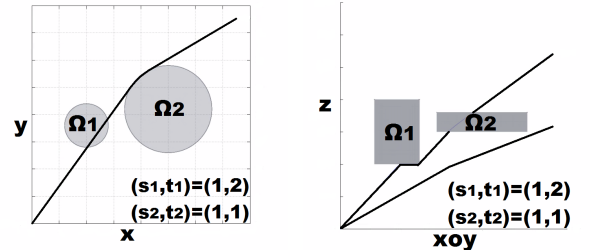
Step 2: Another branch on Ω_1 is developed by deviating the direct route clockwise around Ω_1 . The length of this horizontal route is the lower bound of this subproblem, the value is greater than the current best value. There is no possibility to get a better solution by further branching on Ω_2 , therefore we cut this branch.

Step 3: The third branch on Ω_1 is obtained by imposing a level flight, as shown in Fig. 8(d)(e). The



(a) Example 1: Horizontal Plan

(b) Example 1: Vertical Plan



(c) Example 2: Horizontal Plan

(d) Example 2: Vertical Plan

Figure 6 Routes Construction

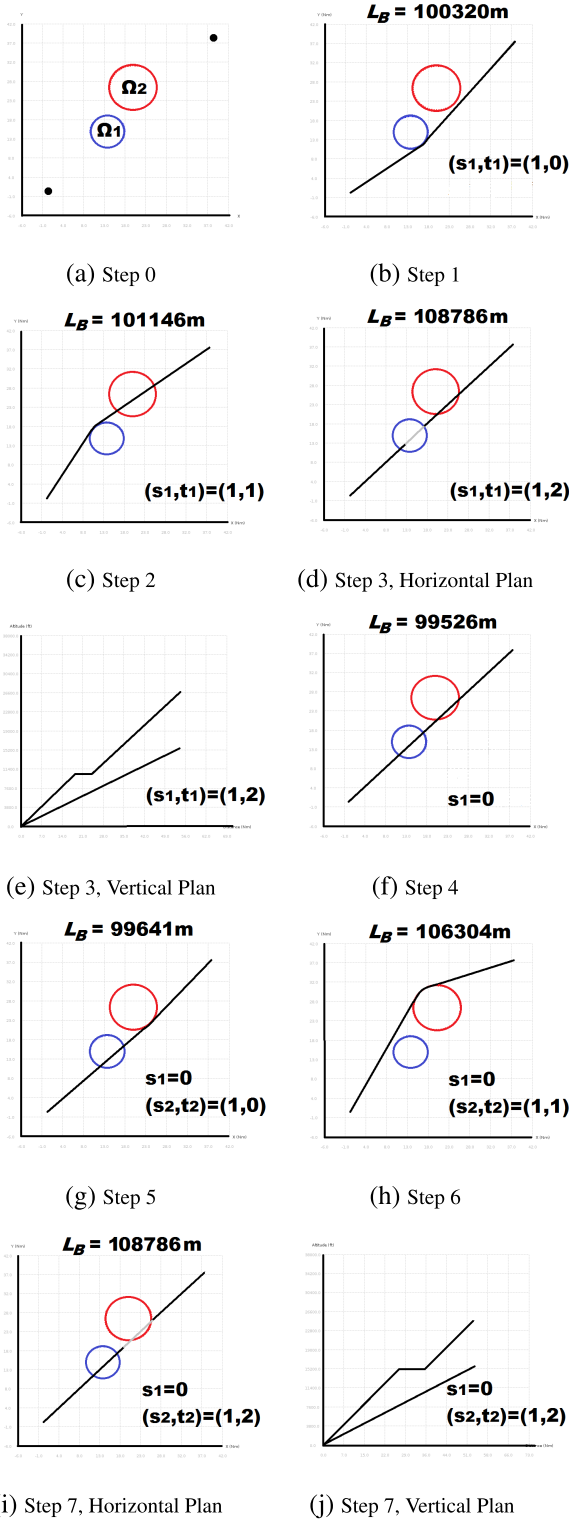


Figure 8 B&B Illustration

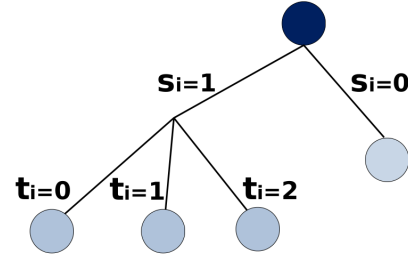


Figure 7 B&B Branching Strategy

lower bound is greater than the current best value, so the branch is cut.

Step 4: The last branch on Ω_1 is with $s_1 = 0$. The route intersects Ω_2 and its lower bound, which corresponds to the length of the direct route, is less than the current best length, thus 4 branches on Ω_2 are developed. In the case $s_2 = 0$, the route is not feasible thus not accepted.

Step 5: By branching counter-clockwise around Ω_2 , the obtained route is still intersecting Ω_1 , thus it is not accepted.

Step 6: By branching counter-clockwise around Ω_2 , we obtain a feasible route with length greater than the current best value, so it is not accepted.

Step 7: The last branch is obtained by imposing a level flight under Ω_2 as shown in Fig. 8(i)(j). This route is still encountered by Ω_1 , so it is not accepted.

All the possible branches are considered. The best distance is 100320m by taking $(s_1, t_1) = (1, 0)$, $s_2 = 0$.

The maximum size of the search space of our problem is 4^m . Indeed we have 4 possibilities to deal with each obstacle. But the size of the search space can be reduced significantly by applying the pre-processing techniques developed in [8]. In fact, the authors prove firstly the shortest path connecting two points that avoids circular obstacles must lie in an ellipse containing these two points. Moreover, they prove another more precise result, the shortest path exists in a convex hull of a few circular obstacles around the line segment connecting the starting and ending points. We apply these pre-processing techniques to reduce the search space before applying our approach. Some simulation results with and without the pre-processing techniques are presented in Section 4.

4 SIMULATION RESULTS

In this section we present two simulation examples. The first one is the case of a SID, the second one is the case of a STAR. Tests were run on a Linux platform with a 2.4 GHz processor and 8 GB RAM.

Different strategies are applied to choose the next sub-problem to solve and the next obstacle to branch on. The simulation results presented in this section use the Best Lower Bound (BLB) for selecting the

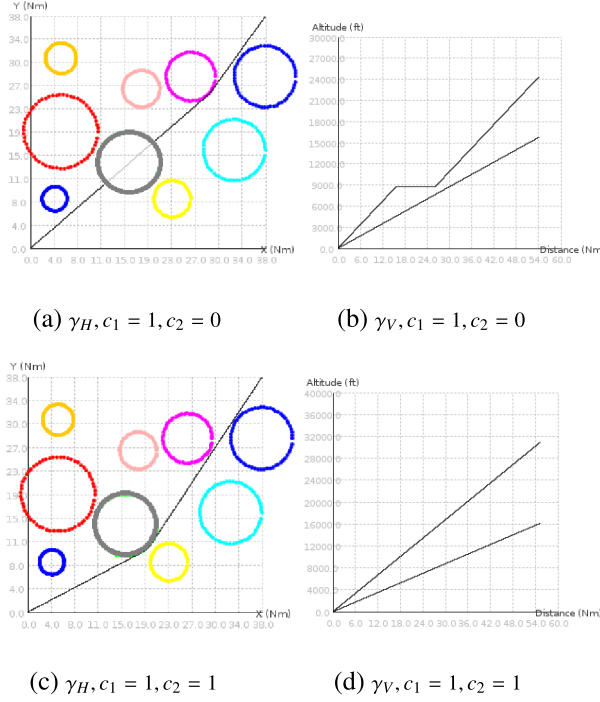


Figure 9 Example 1: Simulation Results

Table 1 Example 1: Characteristics of Nine Obstacles

(x_i, y_i)	r_i	$(z_{i_{inf}}, z_{i_{sup}})$
(7408, 14816)	3704	(701, 2377)
(9260, 35100)	11112	(884, 1707)
(9260, 57412)	4630	(2286, 4389)
(29632, 25928)	9260	(2682, 3109)
(33336, 48152)	5556	(2835, 5425)
(42596, 14816)	5556	(2286, 4389)
(48152, 51856)	7408	(3536, 6828)
(61116, 29632)	9260	(3048, 6706)
(70376, 51826)	9260	(4154, 8022)

next sub-problem, and use the First Intersected Obstacle (FIO) for selecting the next obstacle to branch on. In fact, the combination of strategies "BLB+FIO" gives the minimum computing time together with the minimum number of iterations in most of the tests.

In the first example, the input data are (unit in meter):

- Starting point A, $(x_A, y_A, z_A) = (0, 0, 0)$
- Ending point B, $(x_B, y_B, [z_B, \overline{z}_B]) = (70376, 70376, [4572, 9754])$
- $\alpha_{min} = 4.8\%$, $\alpha_{max} = 9.2\%$
- $N_{max} = 2, L_{min} = 9260, H_{min} = 914$

There are nine obstacles. Table 1 gives the center (x_i, y_i) , radius r_i and altitudes of the two bases $(z_{i_{inf}}, z_{i_{sup}})$ of obstacle $\Omega_i, i = 1, \dots, 9$. The unit is in meter. The simulation results are presented in Table 2 and Fig. 9. We carried out two tests with different values of c_1, c_2 : the first one with $c_1 = 1, c_2 = 0$; the second one with $c_1 = 1, c_2 = 1$. When $c_1 = 1, c_2 = 0$, the length of level flights is not penalized in the objective function, thus Fig. 9(a) and Fig. 9(b) show that the optimal route is obtained with a level flight to avoid the obstacle with the center coordinates (29632, 25928). However, when $c_1 = 1, c_2 = 1$ the length of level flight is penalized in the objective function, and as a result a counter-clockwise turn is made to avoid the mentioned obstacle instead of using a level flight, as shown in Fig. 9(c) and Fig. 9(d). Moreover, thanks to the ellipse and convex hull filters, which reduces the number of the potential obstacles from 9 to 5, the computation time and the number of iterations are reduced effectively.

In the second example, the input data are (unit in meter):

- Starting point A, $(x_A, y_A, [z_A, \overline{z}_A]) = (0, 0, [4572, 7620])$
- Ending point B, $(x_B, y_B, z_B) = (96304, 96304, 914)$
- $\alpha_{min} = 2.6\%$, $\alpha_{max} = 5\%$
- $N_{max} = 2, L_{min} = 9260, H_{min} = 914$

Table 2 Example 1: Simulation Results

		$c_1 = 1$ $c_2 = 0$	$c_1 = 1$ $c_2 = 1$
L_γ (in m)		100082	102478
Time (in s)	no pre-processing	2.6	8.5
	with pre-processing	0.62	1.1
Iterations	no pre-processing	254	1107
	with pre-processing	46	115

Table 3 Example 2: Characteristics of Seven Obstacles

(x_i, y_i)	r_i	$(z_{i_{inf}}, z_{i_{sup}})$
(22224,22224)	9260	(0,5486)
(27780,74080)	14816	(0,3048)
(48152,96304)	7408	(0,4572)
(59264,51856)	12964	(0,4572)
(72228,24076)	9260	(0,4572)
(77784,81488)	14816	(0,3048)
(96304,48152)	12964	(0,4572)

Table 4 Example 2: Simulation Results

		$c_1 = 1$	$c_1 = 1$
		$c_2 = 0$	$c_2 = 1$
L_γ (in m)		142230	142230
Time (in s)	no pre-processing	3.37	1.82
	with pre-processing	0.082	0.084
Iterations	no pre-processing	349	188
	with pre-processing	45	45

There are seven obstacles defined by the parameters presented in Table 3. The simulation results are presented in Table 4 and Fig 10. Note that the optimal route in the case with $c_1 = 1, c_2 = 0$ is the same as the one in the case with $c_1 = 1, c_2 = 1$. Even though by taking $c_2 = 0$, level flights are not penalized in the objective function, no level flight is imposed in the optimal solution. The reason is that the routes with level flights do not offer a better value of the objective function or do not satisfy the constraints that mentioned before. By applying the pre-processing techniques, the number of the potential obstacles is reduced from seven to three, therefore the computation time and the number of iterations are reduced significantly.

5 CONCLUSION AND PERSPECTIVES

In this paper, we introduce a methodology for generating a 3D route in TMA at strategic level, performed by a deterministic global optimization approach. The route is represented by a horizontal curve associated to a cone in the vertical plan. We develop three ways to avoid an obstacle: turn clockwise or counter-clockwise (2D) and level flight (3D) which correspond to branching strategies in a Branch and Bound tailored on the problem. By setting appropriately the penalization coefficients, it is possible to obtain continuous and smooth routes which are available for Continuous Climb Operations (CCO) and Continuous Descent Operations (CDO). To summarize, this approach can be regarded as a decision support tool for the designing of SID/STARS.

In future work, we will consider the problem of designing several routes taking into account the avoidance of obstacles and the separation between them. One possible approach is to decompose the problem in three parts: (a) generating each route separately by applying the Branch and Bound method presented in this paper; (b) detecting the conflicts between the generated routes; (c) eliminating the conflicts by adding fictitious cylinder obstacles at the position of the conflicts.

ACKNOWLEDGMENTS

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.

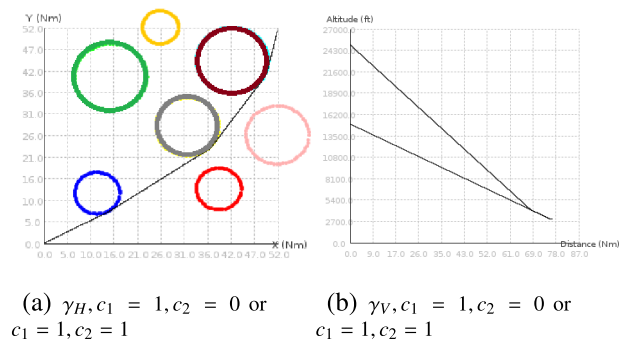


Figure 10 Example 2: Simulation Results

References

- [1] P. Gallina and A. Gasparetto, “A technique to analytically formulate and to solve the 2-dimensional constrained trajectory planning problem for a mobile robot”. *J. Intell. Robot. Syst. (Netherlands)*, vol.27, (no.3), 2000. p.237262.
- [2] O. Souissi et al., “Path planning: A 2013 survey”. *International Conference on Industrial Engineering and Systems Management (IEEE IESM 2013)*, I4e2, Rabat, Morocco, octobre.
- [3] D. Delahaye et al., “Mathematical models for aircraft trajectory design: A survey”. 2013 *ENRI International Workshop on ATM/CNS (EI-WAC2013)*, Tokyo, Japan, February 2013.
- [4] T. Lozano-Perez and M.A. Wesley, “An algorithm for planning collision-free paths among polyhedral obstacles”. *Commun. ACM*, vol22, (no.10), 1979. p.560-570.
- [5] H. Rohnert, “Shortest paths in the plane with convex polygonal obstacles”. *Inform. Process. Lett.*, vol.23, 1986. p.71-76.
- [6] J.A. Storer and J.H. Reif, “Shortest paths in the plane with polygonal obstacles”. *J. Assoc. Comput. Mach. (USA)*, vol.41, (no.5), 1994. p.982-1012.
- [7] M. Pocchiola and G. Vegter, “Minimal tangent visibility graphs”. *Comp. Geom. Theor. Appl. (Netherlands)*, vol.6, 1996. p.303-314.
- [8] D. S. Kim et al., “Shortest paths for disc obstacles”. *Computational Science and Its Applications-ICCSA*, vol.3045, 2004. p.6270.
- [9] D. M. Pfeil, “Optimization of Airport Terminal-Area Air Traffic Operations under Uncertain Weather Conditions”. PhD thesis, Massachusetts Institute of Technology, 2011.
- [10] “ICAO Doc 4444 Air Traffic Management”. Fifteenth Edition 2007
- [11] A. Eele and A. Richards, “Path planning with avoidance using nonlinear branch and bound optimization”. *J. Guid. Control. Dynam. (USA)*, vol.32, 2009. p.384394.