Optimization-Based Design of Departure and Arrival Routes in Terminal Maneuvering Area

Jun Zhou, Sonia Cafieri, Daniel Delahaye, Mohammed Sbihi

Optimization-Based Design of Departure and Arrival Routes in Terminal Maneuvering Area

J. Zhou[a] and S. Cafieri[b] and D. Delahaye[c] and M. Sbihi[d]
ENAC, Université de Toulouse, F-31055, France

The efficient design of departure and arrival routes in the airspace surrounding airports, called the Terminal Maneuvering Area (TMA), is crucial for increasing the capacity of such areas, and thus alleviating congestion around airports caused by world-wide air traffic growth. In this paper, an efficient method of designing departure and arrival routes in TMA is proposed, taking into account the configuration of the airport and nearby environment, as well as related operational constraints, such as obstacle avoidance and route separation. Each route is modeled in three dimensions (3D), and consists of two components: a curve in the horizontal plane and a cone in the vertical plane. A Branch and Bound (B&B)-based approach is developed, whose branching strategies are tailored to how the obstacles are avoided. Routes are generated sequentially, and each route is initially built in isolation. If the route is found to be in conflict with previously generated routes, it is perturbed locally around the conflict zones. Numerical tests, performed on artificially generated problems and the TMA of Paris Charles-de-Gaulle (CDG) airport, demonstrate that the proposed method is efficient and could be embedded in a decision-aid tool for procedure design.

I. Introduction

A Terminal Maneuvering Area (TMA) is an area surrounding one or more neighboring airports, that is designed to handle aircraft arriving at and departing from such airports. Most airports

[a] PhD candidate, ENAC, junzhou@recherche.enac.fr.
[b] Professor, ENAC, sonia.cafieri@enac.fr.
[c] Professor, ENAC, daniel.delahaye@enac.fr.
[d] Associate professor, ENAC, mohammed.sbihi@enac.fr.

have pre-designed routes to guide departures and arrivals, called Standard Instrument Departure (SID) routes and Standard Terminal Arrival Routes (STAR). A SID is a flight route followed by an aircraft from its take-off phase to the en-route phase. A STAR is a route which connects the last en-route waypoint to the Initial Approach Fix (IAF). The route segment connecting the IAF to the Final Approach Fix (FAF) is called the Initial Approach. After the FAF, the flight path is usually specified by ground-based instruments. In the present work, a STAR refers to the section of the route connecting a TMA entry point to a FAF. Currently, SIDs and STARs are designed manually according to operational requirements [1] and airport layout. This kind of design is generally not very efficient and is not designed to optimize any specific objective. Optimizing departure and arrival procedures in TMA is crucial for regulating air traffic flows, whose continual growth [2] can affect air traffic operations. The aim of this work is to automatically design SIDs and STARs in three dimensions (3D), with the objective of minimizing the total route length while satisfying various constraints, including obstacle avoidance and route separation. The routes are designed to be suitable for a range of aircraft, since the specifications and capabilities of aircraft using the routes can vary significantly. The proposed design is at a strategic level, thus only static obstacles are taken into account.

Performance Based Navigation (PBN) is one of the key navigation concepts to enable future airspace and traffic flow design [3]. It offers operational benefits such as enhanced safety and increased efficiency. Required Navigation Performance (RNP), a typical way of navigating within the PBN concept, is especially useful in complex airspaces such as TMAs, since it provides higher design flexibility. Currently the RNP-1, a type of RNP with a specified performance level, enables the Radius-to-Fix (RF) functionality, which can be applied in SID/STAR design. A RF leg is defined as an arc with specified radius between two defined waypoints [4]. Route construction in our proposed method is based on the RNP concept, in particular the RF functionality in SIDs/STARs.

The problem considered fits into the framework of path planning. More specifically, it is a route design problem. In contrast to trajectory design, the routes to be designed are not associated with any notion of time. The problem of path planning has been studied since the 1980s, especially in the domain of robotics, *e.g.*, [5, 6]. Recently, optimal flight path design has received increased

attention, and a range of solution approaches have been proposed (nicely summarized in [7]). In the following, several methods for optimal route design proposed in the literature are reviewed, according to different points of view: designing in 2D *vs.* in 3D, one route design *vs.* multiple routes design, exact design approach *vs.* heuristic design approach.

For 2D route design, a typical approach for computing the shortest path between two points that avoids a fixed set of polygonal obstacles, is to use a Visibility Graph, introduced by Lozano-Pérez and Wesley in 1979 [8]. It has been proven that the 2D shortest path avoiding polygonal obstacles is the shortest path on the corresponding Visibility Graph [8]. In the case when both polygonal and curved obstacles are present, the notion of a Tangent Graph is introduced by Liu and Arimoto in 1992 [9]. A Tangent Graph allows one to take into account obstacles in the form of circular discs, such as in [10, 11]. In [11], the authors prove that the shortest path lies in a convex hull of a few circular obstacles around the line segment connecting the start and end points. This significantly reduces the size of the search space. In addition to graph-based methods, several other approaches have been proposed for solving 2D optimal route design problems. For example, a Branch and Bound (B&B) method that builds an optimal route avoiding circular obstacles is proposed in [12]. The branching strategy is based on the decision to avoid an obstacle by turning clockwise or counter-clockwise. In [13], the path planning problem in a TMA is solved using convex hull generation combined with a Genetic Algorithm (GA). Again, the obstacles are avoided by turning counter-clockwise or clockwise around their boundaries. In [14], a dynamic programming approach, based on a variant of the Bellman-Ford method, is applied to compute an optimal turn-constrained route in the presence of weather hazards. The generated route is in a polygonal form in 2D.

Optimal route design in 3D is more difficult than in 2D, since the vertical profile has to be taken into account. One approach proposed in the literature is to extend the 2D visibility graph to 3D with polyhedral obstacles [15, 16]. However, the problem becomes NP-hard in 3D [17], primarily because the shortest path does not necessarily pass through the vertices of polyhedrons. Other works have contributed to the 3D-route design in the aeronautical domain. In [18], two different approaches are considered for designing routes connecting a pair of airports: an A* algorithm and a GA. The horizontal route is only allowed to vary at the boundary of a TMA, thus it is polygonal

in shape. The associated vertical profile is specified by connecting fixed level flights using linear climb or descent paths. In [19], the author designs airport terminal routes that avoid obstacles using a modified A* algorithm. Other operational constraints are also considered (*e.g.*, the need for an aircraft to perform smooth turns). The vertical profile is in the form of a cone, bounded by the minimum and maximum aircraft slopes. In [20], the departure and arrival routes are designed using the Fast Marching Method with an obstacle avoidance constraint. However, the proposed algorithm does not specifically deal with the take-off or descent slopes of aircraft. In [21], a B&B method is used to design a 3D-route avoiding obstacles. The branching strategy is tailored to the way the obstacles are avoided, namely turning clockwise or counter-clockwise, or imposing a level flight below an obstacle.

The majority of the works cited above concern the design of a single route ([8–17, 21]). The main difficulty when designing multiple routes is ensuring that distinct routes meet separation constraints, which is particularly important from the Air Traffic Management (ATM) point of view. In order to design multiple routes satisfying the separation criterion, some authors apply a sequential *1-against-n strategy* where the routes are generated one after another according to a user-defined priority order (for example according to the traffic load on each route). The previously constructed routes become obstacles for the route under construction. For the sequential strategy, the quality of a solution depends on the order in which routes are generated. In [18] and following works [22, 23], routes are generated sequentially according to traffic load, and the route separation is realized through an A* algorithm. The modified A* method proposed in [19] deals with route separation in a similar way. In [20], a Simulated Annealing (SA) method is applied to optimize the order in which the routes are generated. In [24], the design of STARs based on the RF functionality is considered, where routes are generated sequentially and merged gradually to the runway. Some other works apply a *global strategy*, where all routes are generated simultaneously in order to minimize a global cost associated with the set of routes. For example in [18], a GA is applied to generate routes simultaneously and the route sections involved in conflicts are penalized in the global cost. Similarly, in [25], a SA algorithm is used to simultaneously generate multiple 3D-routes in TMA.

As the design of routes is generally achieved by minimizing or maximizing a given objective,

the problem is naturally modeled as an optimization problem. Exact or heuristic methods can be applied to find a solution. An exact approach guarantees an optimal solution, if such a solution exists, but may be time-consuming for difficult problems. Two typical exact approaches based on graph models, Dijkstra's algorithm [26] and Bellman-Ford algorithm [27], are applied in the previously cited works [11, 14]. Another widely used exact approach is the B&B method [28], which is used to solve discrete and combinatorial optimization problems. Two applications of the B&B method in the considered context are presented in [12, 21]. These exact methods are efficient for the design of an individual route. However, the optimal design of multiple 3D-routes satisfying the numerous constraints of TMAs (route separation, obstacle avoidance, noise abatement, etc.) is a much more complex problem. One of the main difficulties is finding suitable analytic mathematical expressions for some of the problem constraints, making exact methods difficult to apply. In contrast to exact approaches, a heuristic approach seeks to produce good-quality but not necessarily optimal solutions in reasonable computation time. Thus, heuristics are more commonly used for solving 3D or multiple route design problems. The A* algorithm [29] is a widely used heuristic approach in graph-based shortest path design problems. Several previously mentioned works apply the A* (or modified A*) method, such as [9, 18, 19]. Meta-heuristics [30], including GA and SA, are also often applied in the ATM domain, such as in [13, 18, 20, 25].

In the present paper, a method based on an exact approach is proposed to solve the problem of designing multiple routes in a TMA. In contrast to previous work, where routes are usually considered in 2D, our methodology deals with routes in 3D by taking into account the specific aircraft take-off and landing slopes. Moreover, the most commonly used method to separate two routes in the literature is to modify routes in the horizontal plane. The method proposed in this paper extends previous approaches by also allowing modifications in the vertical plane. The obtained routes are in compliance with the RNP concept, and can be regarded as preliminary guidelines for manual design. In the proposed methodology, obstacle avoidance and route separation are considered as the primary constraints. An obstacle is modeled as a cylinder in 3D. Each 3D-route consists of a horizontal curve associated with a vertical cone. The horizontal curve is composed of a series of arcs, used to avoid obstacles, and straight line segments that tangentially connect two arcs. In

the vertical plane, for each route a cone is designed and the possibility of imposing a level flight below an obstacle is allowed. This is a potentially effective way of avoiding other flight paths, as it enriches the space of possible maneuvers and corresponds to what is done in reality (for instance, it is used on one STAR in Paris Roissy airport, see Sec .V). The problem of designing multiple SIDs/STARs is modeled as an integer optimization problem. A simpler subproblem of designing a single route avoiding obstacles is solved by B&B [21], where the branching strategies correspond to the different ways of avoiding an obstacle. To address the design of multiple routes, a separation constraint (3NM in the horizontal plane, or 1000ft in the vertical plane) between pairwise routes must also be satisfied. Two routes are defined as in *conflict* when a loss of separation occurs. A conflict detection technique is proposed to check pairwise conflicting routes. Moreover, a B&B-based method is developed to sequentially build minimal-length routes that satisfy obstacle avoidance and route separation constraints.

This paper is organized as follows. Section II introduces the routes and obstacles modeling. Section III explains the approach used for generating a single route. Section IV presents the method for solving the multiple route design problem. Section V presents some simulation results and, finally, Sec. VI draws conclusions.

II. Problem statement and mathematical model

In this study, the SIDs/STARs design in the TMA surrounding a single airport is considered. The input data are presented in Tables 1 and 2. In the following, the route and obstacle modeling as well as the constraints taken into account are presented.

Table 1 Input data related to airport configuration

the total number of obstacles to be avoided, $M \in \mathbb{N}$
the total number of the available runways
the usage of each runway (take-off or landing)
the direction of take-off or landing on each runway
the coordinates and altitudes of thresholds and center for each runway
the coordinate and altitude of the corresponding FAF for a runway used for landing

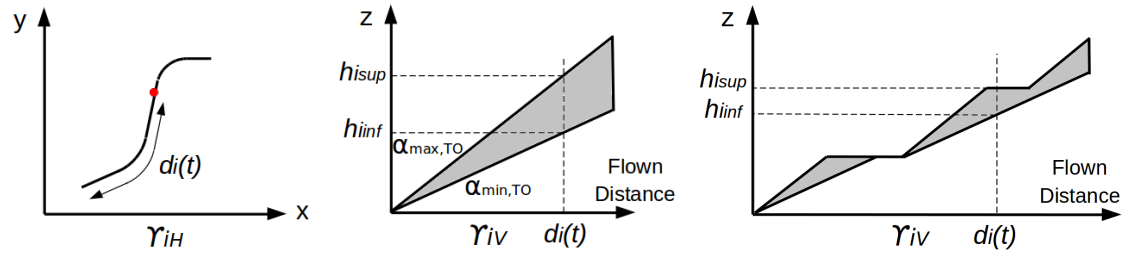
Table 2 Input data related to route design

the total number of routes to be built, $N \in \mathbb{N}$
the start point A_i and starting altitude H_{A_i} for route i, $i = 1, \cdots, N$
the end point B_i for route i, $i = 1, \cdots, N$
traffic load on each route
minimum (maximum) take-off slope $\alpha_{min,TO}$ ($\alpha_{max,TO}$)
minimum (maximum) landing slope $\alpha_{min,LD}$ ($\alpha_{max,LD}$)
minimum (maximum) radius on a RF leg R_{min} (R_{max})
maximum number of level flights on each route N_{max}
minimum altitude of each level flight H_{min}
minimum length of each level flight L_{min}

A. Obstacle and route modeling

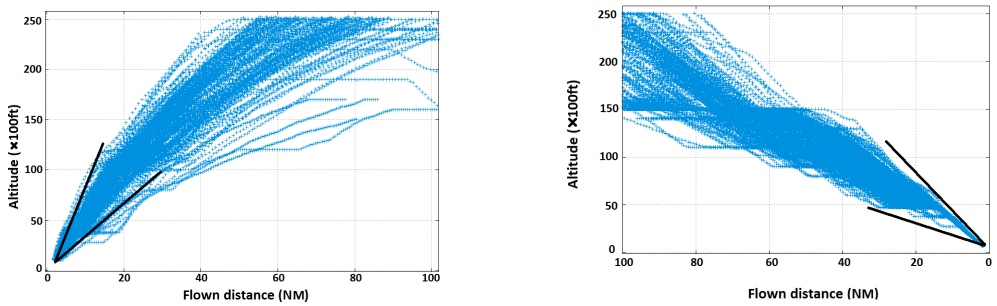
The obstacles (together with their protection areas) are modeled as cylinders in 3D, whose projections onto the horizontal plane are disks. Let $M \in \mathbb{N}$ be the number of cylinders. Each cylinder $\Omega_j, j = 1, \cdots, M$ is defined by $(C_j(x_j, y_j), r_j, z_{j_{inf}}, z_{j_{sup}})$, where $C_j(x_j, y_j)$ is the center and r_j is the radius of the top and the bottom of the cylinder; $z_{j_{inf}}$ and $z_{j_{sup}}$ are the altitude of the lower and upper bases respectively.

Let $N \in \mathbb{N}$ be the total number of routes to be built. A 3D-route $\gamma_i, i = 1, \cdots, N$ is defined by two elements: a curve γ_{i_H} in the horizontal plane and a cone γ_{i_V} in the vertical plane. The horizontal curve γ_{i_H} is designed in the form of a series of segments and arcs, motivated by the fact that the shortest path among circular obstacles in the horizontal plane consists of segments (tangentially connecting two obstacles) and arcs (lying on the border of obstacles) [9]. An example is shown in Fig. 1a). The form of the horizontal curve both simplifies the route modeling, and corresponds to the shape of SIDs/STARs under RNP. In particular, a segment defined by two tangent points corresponds to a standard point-to-point leg in ATM and an arc defined by start and end tangent points, turn center and radius, corresponds to a RF leg in ATM. The vertical cone γ_{i_V} is designed to contain all ascent (or descent) profiles of the aircraft flying on this route. The vertical



a) **b)** **c)**

Fig. 1 Examples of γ_{i_H} and γ_{i_V}.



a) Take-off profiles **b) Landing profiles**

Fig. 2 Take-off and landing profiles in Paris CDG airport.

profile is a band enclosed by two piecewise linear continuous functions. Two examples are shown in Figs. 1b), 1c), where the bands are represented by the shaded areas. More precisely, the slope of each segment on the function corresponding to the lower bound is either α_{min} (on a climbing or descending route section) or 0 (on a route section where the flight level is maintained). Similarly, the slope of each segment on the function corresponding to the upper bound is either α_{max} or 0. By abuse of language, this band is called a *cone*. The idea of taking a cone that contains all vertical profiles is inspired by the behavior illustrated in Fig. 2, which shows some real take-off and landing data at Paris Charles-de-Gaulle (CDG) airport. From the figure it can be seen clearly that in both cases the vertical profiles are contained in a region defined by two straight lines. The variation in vertical profile is mainly due to different aircraft mass and performance, and to the effect of the wind.

In the horizontal plane, γ_{i_H} connects a start point A_i (x_{A_i}, y_{A_i}) to an end point B_i (x_{B_i}, y_{B_i}). In a SID case, the start point is at the midpoint of a runway threshold and the end point is an exit point

of a TMA. In a STAR case, for the simplicity of implementation, the route is built by considering a FAF as start point and an entry point of TMA as end point. The FAF is usually aligned to the central line of the corresponding runway, and aircraft follow specific guidance (Instrument Landing System (ILS) as example) from the FAF to the runway threshold, thus the route section between FAF and the runway threshold is not considered in this study. The horizontal route γ_{i_H} is a smooth map 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 the vertical plane, the start point A_i (x_{A_i}, y_{A_i}) is associated with an altitude H_{A_i}, corresponding to the airport elevation in a SID case, and to the altitude of the FAF in a STAR case. The vertical profile γ_{i_V} 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 $\mathbb{R}$, $d_i(t) = \int_0^t \|\gamma'_{i_H}(s)\|_2 \, \mathrm{d}s$ is the flown distance until t in the horizontal plane, $[h_{i_{inf}}(d_i), h_{i_{sup}}(d_i)]$ is the interval defined by the cross section of the cone at d_i, and $\gamma_{i_V}(0) = [H_{A_i}, H_{A_i}]$. Figure 1 illustrates how γ_{i_H} is associated with γ_{i_V} in the case of a SID, where $H_{A_i} = 0$ and $\alpha_{min,TO}$ (respectively, $\alpha_{max,TO}$) is the minimum (respectively, maximum) take-off slope of aircraft on this route.

B. Decision variables

An obstacle is defined as *active* when it is touched by a route and must be avoided according to one of the following maneuvers: turn counter-clockwise, turn clockwise or impose a level flight. For a given route γ_i, each cylinder Ω_j is associated with two decision variables s_{ij} and t_{ij}. The variable s_{ij} defines whether Ω_j is active or not with respect to the route γ_i:

$$s_{ij} = \begin{cases} 0, \text{ if } \Omega_j \text{ not active} \\ 1, \text{ if } \Omega_j \text{ active} \end{cases} \quad \forall i = 1, \cdots, N; \ \forall j = 1, \cdots, M, \tag{3}$$

while t_{ij} defines the way an active obstacle Ω_j is avoided on route γ_i:

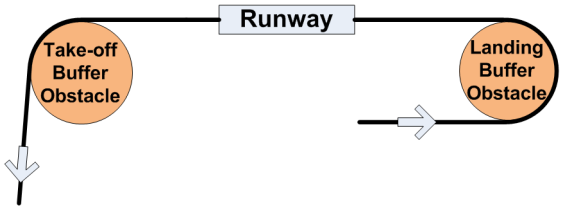


Fig. 3 Runway buffer obstacle.

$$t_{ij} = \begin{cases} 0, \text{ if turn counter-clockwise} \\ 1, \text{ if turn clockwise} \\ 2, \text{ if impose a level flight below obstacle } \Omega_j \end{cases} \qquad \forall i = 1, \cdots, N;\ \forall j = 1, \cdots, M. \tag{4}$$

Note that when an obstacle is not active ($s_{ij} = 0$), the value of t_{ij} is not applicable in the design. The proposed decision variables are based on the available obstacle avoidance strategies and take into account the considered route shape.

C. Constraints

TMA is one of the most complex types of airspace; many constraints have to be satisfied. These constraints fall into two main categories: operational constraints related to air traffic operations (such as obstacle avoidance and route separation), 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 following constraints are considered.

Obstacle avoidance

This constraint is considered and handled through the choice of the decision variables.

Route separation

The standard separation norm between pairwise aircraft in a TMA is 3NM in the horizontal plane or 1000ft in the vertical plane. Routes are designed to respect these separation criteria, as a consequence, aircraft following two different routes are automatically separated.

Runway alignment

A SID must join a route section smoothly after a take-off leg and a STAR must go directly to the corresponding runway before the FAF. To deal with the runway alignment constraint, the

notion of *buffer obstacles* is proposed, as shown in Fig. 3. Each route γ_i is associated with one buffer obstacle Ω_{bi} in a cylinder shape, defined by $(C_{bi}(x_{bi}, y_{bi}), r_{bi}, z_{bi_{inf}}, z_{bi_{sup}}, t_{\Omega_{bi}})$, where $t_{\Omega_{bi}}$ is the turn orientation on Ω_{bi} and the other parameters are as defined in Sec. II A. The values of these parameters are input data and ensure that the tangent segment connecting the buffer obstacle and the start point is parallel to the corresponding runway, so that the designed route joins the runway directly. A buffer obstacle is always active and the choice of its avoidance strategy is either counter-clockwise turn or clockwise turn.

RF related constraints

The first constraint related to RF is the minimum turn radius R_{min}. The turn radius depends on the bank angle Φ and ground speed V_{GS} [31, 32], and is calculated using the following equation:

$$R = \frac{V_{GS}^2}{g \cdot tan(\Phi)}, \tag{5}$$

where g is the acceleration of gravity. According to Eq. (5), for a fixed value of V_{GS}, the lowest radius of a flyable RF leg is given by $R_{min}(V_{GS}) = \frac{V_{GS}^2}{g \cdot tan(\Phi_{max})}$, where Φ_{max} is the maximum value of the bank angle. Consequently, in order to ensure that a RF leg is flyable for all aircraft following it, the radius of the RF leg has to be greater than $R_{min} = \frac{V_{GS,max}^2}{g \cdot tan(\Phi_{max})}$, where $V_{GS,max}$ is the maximum ground speed in a TMA. Taking $V_{GS,max} = 400$kt and $\Phi_{max} = 25°$ (according to [33]), the corresponding minimum radius R_{min} is equal to 5NM. To satisfy the constraint on minimum turn radius, a pre-processing is applied to the obstacles: for an obstacle with radius lower than R_{min}, its radius is increased to R_{min}. Note that, as aircraft ground speed is very low near the runway, the radius of a runway buffer obstacle may be lower than R_{min}. In order to avoid a very big turn, an upper bound on the RF leg radius is also imposed. Let R_{max} be the largest radius of a RF leg. According to [34], the value of R_{max} is set to be 13NM. In our pre-processing, all input obstacles are modeled as cylinders whose radius of lower and upper bases is lower than R_{max}. Another constraint related to RF is that RF legs must terminate at least 2NM prior to the FAF [31].

Monotonicity of the vertical profile

The vertical profile for a SID should be monotonically ascending and for a STAR should be

monotonously descending. That is, for $t_1, t_2 \in [0,1]$, if $t_1 \leq t_2$, Eq. (6) must be satisfied:

$$\begin{aligned} h_{i_{inf}}(t_1) &\leq h_{i_{inf}}(t_2), \ \ i = 1, \cdots, N \\ h_{i_{sup}}(t_1) &\leq h_{i_{sup}}(t_2), \ \ i = 1, \cdots, N. \end{aligned} \tag{6}$$

Level flight related constraints

Since the changes in flight levels have an influence on the fuel consumption and CO2 emissions, the number of level flights on each route is bounded above by N_{max}, usually set to 2. For each route γ_i:

$$\sum_{j=1}^{m} \max(t_{ij} - 1, 0) \leq N_{max}, \ \ i = 1, \cdots, N. \tag{7}$$

Moreover, 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, the following constraints are imposed: for an obstacle Ω_j, if $z_{j_{inf}} < H_{min}$, then no level flight is allowed below it, therefore $\forall i, t_{ij} \in \{0, 1\}$. In particular, for the obstacles connecting to the ground (*e.g.*, mountains, buildings), no level flight is allowed below them. They are avoided only by horizontal turns. The obstacles that can be avoided by level flight can be military areas and conflict zones between routes (to be presented in Sec. IV). Finally, to ensure the passenger comfort, the length of each level flight should not be too short, therefore a minimum length L_{min} for each level flight is required.

D. Objective function

To define our objective function, let us first define, for each route γ_i, a weighted sum L_{γ_i} of two terms: the length of the horizontal curve γ_{i_H} and the total length of level flights. More precisely:

$$L_{\gamma_i} = c_1 \left(\int_0^1 \|\gamma'_{i_H}(t)\|_2 \ \mathrm{d}t \right) + c_2 \ell_{i_{LF}}, \tag{8}$$

where $\ell_{i_{LF}}$ is the length of route sections corresponding to level flights projected onto the horizontal plane. The coefficients c_1 and c_2 are penalty parameters, whose values are user-defined and depend on the weight associated with the corresponding term. The objective is to minimize the sum of $L_{\gamma_i}, i = 1, \cdots, N$:

$$L = \sum_{i=1}^{N} L_{\gamma_i} \tag{9}$$

In summary, the optimization problem $(\mathcal{P})$ is:

$$(\mathcal{P}) \begin{cases} \min \ L = \sum_{i=1}^{N} L_{\gamma_i} \\ \text{s.t.} \quad \text{obstacle avoidance} \\ \qquad \text{route separation} \\ \qquad \text{runway alignment} \\ \qquad \text{RF related constraints} \\ \qquad \text{monotonicity for the vertical profile} \\ \qquad \text{level flights related constraints} \end{cases} \tag{10}$$

This is a combinatorial optimization problem as the decision variables take integer values. The numerous constraints to be considered make the SIDs/STARs design a very complex problem. Therefore, in Sec. III the simpler subproblem of designing a single optimal route is considered. The proposed solution approach for the design of multiple routes is presented in Sec. IV.

III. Designing one route

The strategy proposed in [21] is used to design an individual route γ_i avoiding obstacles in TMA in an optimal way. Note that all of the constraints presented in Sec. II C, except route separation, are considered. This is the basis of our proposed solution approach to build multiple routes, where the constraint of route separation is considered additionally. To design a single optimal route, a B&B method is applied. This is inspired by the approach presented in [12], where 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 whether an obstacle is avoided by turning clockwise or counter-clockwise. This branching strategy is extended to also allow obstacles to be avoided by imposing a level flight below the obstacle. Our branching strategy is illustrated in Fig. 4. More precisely, in order to branch on an obstacle Ω_{ij}, it is first set as active ($s_{ij} = 1$) or not ($s_{ij} = 0$); when it is active, three branches are developed in order to account for the 3 possibilities of avoiding it: counter-clockwise ($t_{ij} = 0$), clockwise ($t_{ij} = 1$) or imposing a level flight ($t_{ij} = 2$), as illustrated in Fig. 5. More precisely, the counter-clockwise and clockwise turn avoid the obstacle along its border (Figs 5a), 5b)), and the level flight is imposed under the lower basis of the obstacle (Fig 5c)).

Let SP be a subproblem in B&B for the generation of γ_i, where only some of the values of

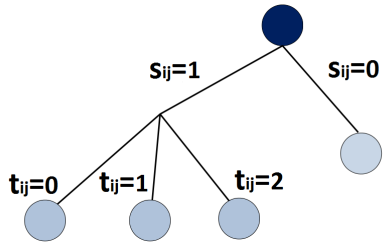
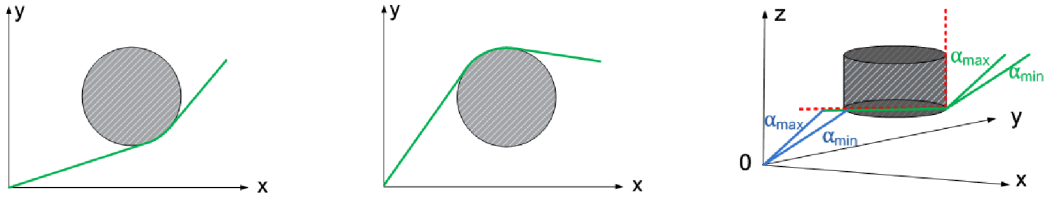


Fig. 4 Branch and Bound Branching Strategy.



a) $(s_{ij}, t_{ij}) = (1, 0)$**, 2D View** **b)** $(s_{ij}, t_{ij}) = (1, 1)$**, 2D View** **c)** $(s_{ij}, t_{ij}) = (1, 2)$**, 3D View**

Fig. 5 The routes associated with different values of the decision variables.

the decision variables are instantiated ($\{(s_{ij}, t_{ij})\}_{j\in J}, J \subset \{1, \cdots, M\}$). The values of the decision variables that are not yet determined are set as non-active ($\{(s_{ij} = 0)\}_{j\in\{1,\cdots,M\}\setminus J}$). A route γ_{SP} corresponding to these values is first computed (following the procedure explained later). If γ_{SP} avoids an active obstacle by an arc with a central angle larger than 180°, a new route $\tilde{\gamma}_{SP}$ is computed, by setting the concerned obstacle as non-active. The lower bound corresponding to SP, denoted as LB_{SP}, is computed by:

$$LB_{SP} = c_1 \left(\int_0^1 \|\tilde{\gamma}'_{SP_H}(t)\|_2 \, \mathrm{d}t \right) + c_2 L_{min} \sum_{j=1}^{M} \max(t_{ij} - 1, 0) \tag{11}$$

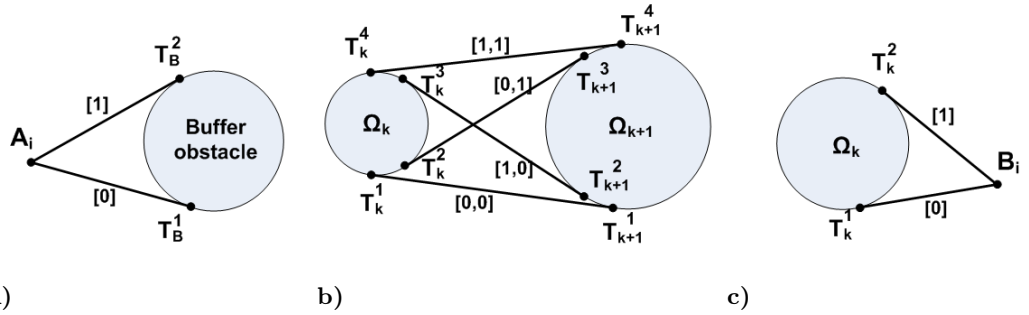
where $\tilde{\gamma}_{SP_H}$ is the horizontal profile of $\tilde{\gamma}_{SP}$, and $t_{ij}, j = 1, \cdots, M$ are related to the choices of the obstacle avoidance strategies in SP. The obtained lower bound is used to identify whether a branch requires further exploration. The pseudo-code of the B&B method for generating a single optimal route $\gamma_i, i \in \{1, \cdots, N\}$ is presented in Algorithm 1.

Algorithm 1 Generation of a single route using B&B

Require: start and end points (A_i, B_i), the altitude of the start point H_{A_i}, buffer obstacle Ω_{bi}, obstacles $\Omega_j, j = 1, \cdots, M$

Initialize: subproblems list $SPL = \emptyset$, current best length $L_{CB} = +\infty$
Build the direct route connecting A_i and B_i, denoted as γ_i^0
if γ_i^0 satisfies all constraints **then**
    γ_i^0 is the optimal route, $\gamma_i := \gamma_i^0$, $L_{CB} := L_{\gamma_i^0}$
else
    Develop new branches according to Fig. 4, add them to SPL
end if
while $SPL \neq \emptyset$ **do**
    Select 1 subproblem SP ($\{(s_{ij}, t_{ij})\}_{j\in J}, J \subset \{1, \cdots, M\}$) in SPL and remove it from SPL
    Build the corresponding horizontal curves γ_{SP_H} and $\tilde{\gamma}_{SP_H}$
    Build the corresponding vertical cones γ_{SP_V}
    Compute the lower bound LB_{SP} according to Eq. (11)
    if $LB_{SP} < L_{CB}$ **then**
        if γ_{SP} satisfies all constraints and γ_{SP} is touched by all active obstacles **then**
            Compute $L_{\gamma_{SP}}$ according to Eq. (8)
            if $L_{\gamma_{SP}} < L_{CB}$ **then**
                $\gamma_i := \gamma_{SP}$, $L_{CB} := L_{\gamma_{SP}}$
            else if SP has remaining obstacle to be branched on **then**
                Develop new branches according to Fig. 4, add them to SPL
            end if
        else if SP has remaining obstacle to be branched on **then**
            Develop new branches according to Fig. 4, add them to SPL
        end if
    end if
end while
return γ_i and L_{CB}

Given the values of $\{(s_{ij}, t_{ij})\}_{j\in J}, J \subset \{1, \cdots, M\}$, the corresponding horizontal curve $\gamma_{i_H}^J$ depends on the subset J. For simplicity, the index J is omitted and the route is denoted by γ_{i_H}.



a) **b)** **c)**

Fig. 6 Different scenarios for connecting tangent points. In the notations $[t_{i,k}]$ and $[t_{i,k}, t_{i,k+1}]$, $t_{i,k}$ indicates the turn direction to avoid obstacle Ω_k on route γ_i.

The route is built in the following way. First, the active obstacles (excluding the buffer obstacle) avoided by a turn ($t_{ij} = 0$ or 1), are numbered in 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 (the center of the two bases of obstacle Ω_j) onto the line (A_iB_i). Next, the horizontal curve γ_{i_H} is computed by tangentially connecting, the buffer obstacle and successive active ($s_{ij} = 1$) obstacles which are associated with counter-clockwise ($t_{ij} = 0$) or clockwise ($t_{ij} = 1$) turns, following the order determined above. The alternative scenarios for connecting tangent points are illustrated in Fig. 6.

After obtaining γ_{i_H}, the associated vertical profile γ_{i_V} is in the form of a cone bounded by two straight line segments whose slopes are $\alpha_{min,TO}$, $\alpha_{max,TO}$ in a SID case (respectively, $\alpha_{min,LD}$, $\alpha_{max,LD}$ in a STAR case) (Fig. 1). For an active obstacle Ω_j with $t_{ij} = 2$, before imposing a level flight, we check whether the cone intersects Ω_j both in the horizontal and vertical plane. We first check the intersection in the horizontal plane, as shown in Fig.7. The length of route sections from A_i to the two intersection points are denoted as d_1 and d_2. When an intersection exists in the horizontal plane, we continue to check in the vertical plane whether the cone intersects Ω_j. If neither of the two conditions:

$$\begin{aligned} h_{i_{inf}}(d_1) \geq z_{j_{sup}} \quad \text{(Fig. 8a))} \\ h_{i_{sup}}(d_2) \leq z_{j_{inf}} \quad \text{(Fig. 8b))} \end{aligned} \tag{12}$$

is satisfied, then the vertical intersection exists. If intersection occurs in both the horizontal and the vertical planes, then the corresponding level flight is imposed under the obstacle. Moreover, the vertical profiles of the lower and the upper bounds are computed separately. For each bound,

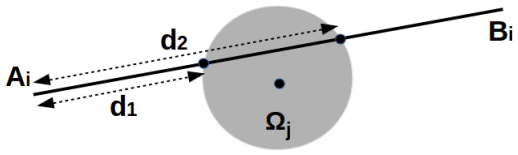
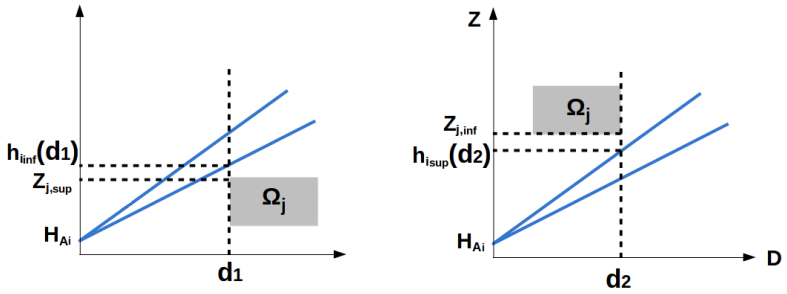


Fig. 7 Checking intersection between γ_{i_H} and Ω_j in the horizontal plane.



a) Checking intersection at d_1 **b) Checking intersection at d_2**

Fig. 8 Checking intersection between γ_{i_V} and Ω_j in the vertical plane.

the level flight starts at the point where the altitude reaches $z_{j_{inf}}$, and ends at the point where the corresponding length in the horizontal plane is d_2. Afterwards, the vertical profiles of the upper and the lower bounds continue to increase with the slopes α_{max} and α_{min} respectively. Note that if an active obstacle with $t_{ij} = 2$ is not intersected by the cone associated with the horizontal route, then the corresponding level flight cannot be imposed. In such a case, the route is infeasible regarding our definition of "active obstacle".

IV. Designing multiple routes

To design N conflict-free routes avoiding obstacles, where a *conflict* between two routes is defined as a violation of the minimum separation (3NM in the horizontal plane, or 1000ft in the vertical plane), a sequential *1-against-n* approach is proposed, based on the B&B method presented in Sec. III. Routes are generated sequentially in decreasing order of their traffic load. Previously built routes become obstacles for new routes. Note that, ordering the routes by traffic load provides better chance of finding near-optimal solutions, but does not guarantee optimality.

Each new route is initially generated using the B&B method proposed in Sec. III. Pairwise conflicts are then detected between the new and the previously generated routes. The method used

for conflict detection is presented in Sec. IV A. The detected conflicts are clustered into groups. Each group of conflicts is associated with a *fictitious obstacle*, modeled as a cylinder as in Sec. II A. The method used to cluster conflicts and create the fictitious obstacles is described in Sec. IV B. In the case when more than one group of conflicts exist, the corresponding fictitious obstacles are ordered according to their projection lengths onto the line connecting the start and end points of the route, as explained in Sec. III. In order to solve the conflict corresponding to the first fictitious obstacle $\Omega_f(C_f(x_f, y_f), r_f, z_{f_{inf}}, z_{f_{sup}})$, the natural solution is to perturb the considered route locally around this conflict zone. Again, three possible perturbations are proposed: turning counter-clockwise, turning clockwise, or imposing a level flight. Afterwards, the remaining route section is built by re-applying B&B, and then applying the conflict detection technique and route perturbation method to the newly generated section of the route. Thus, the new route is built progressively, and each of its sections is perturbed based on the optimal route structure given by the B&B method. Note that a conflict is not guaranteed to be solved by one of the available route perturbation strategies. It may happen that there are not enough degrees of freedom available to be able to solve the conflict. In this case, the proposed algorithm provides information on the unsolved conflict and on the length of involved route sections. A pseudo-code of the proposed B&B-based sequential approach is presented in Algorithm 2. The following subsections describe the key intermediate steps.

A. Conflict detection

In steps 6 and 15 of Algorithm 2, a conflict detection technique is applied on pairs of routes. Detecting conflicts between 3D-routes, especially when the vertical profiles are cones instead of curves, is not an easy problem. A two-steps scheme is proposed to deal with this problem. First, a check is performed in the horizontal plane, to see if any pairwise routes violate the 3NM separation. Then, for the route sections found to be in violation of horizontal separation constraint, a check is performed in the vertical plane, to see if they violate the 1000ft separation. If a route section violates both horizontal and vertical separation constraints, then it is in conflict.

Algorithm 2 Generation of multiple routes using the B&B-based approach

Require: routes start and end points, altitudes of the corresponding start points, and the associated buffer obstacles $\{(A_i, B_i), H_{A_i}, \Omega_{bi} | i = 1, \cdots, N\}$ (classified in an increasing order of traffic load), obstacles $\{\Omega_j | j = 1, \cdots, M\}$

Initialize: list of optimal routes $list = \emptyset$

Apply B&B to generate the optimal route γ_1 connecting (A_1, B_1) ▷ call Algorithm 1((A_1,B_1), H_{A_i}, Ω_{b1}, $\{\Omega_j | j = 1, \cdots, M\}$)

Add γ_1 to $list$

for i=2 to N **do**

    Apply B&B to generate an initial route γ_i^0 connecting (A_i, B_i) ▷ call Algorithm 1(A_1,B_1), H_{A_i} Ω_{bi}, $\{\Omega_j | j = 1, \cdots, M\}$)

    Detect pairwise conflicts between γ_i^0 and routes in $list$

    if no conflict detected **then**

        Set $\gamma_i := \gamma_i^0$, add γ_i to $list$

    else

        Set the current route $\gamma_i^{cur} := \gamma_i^0$

        while conflict detected **do**

            Cluster conflicts and associate a fictitious obstacle with each cluster

            Order the fictitious obstacles with respect to (A_i, B_i) in the same way as in Sec. III

            Perturb γ_i^{cur} locally around the conflict zone corresponding to the first fictitious obstacle Ω_f ▷ call Algorithm 3(γ_i^{cur}, Ω_f, $\{\Omega_j | j = 1, \cdots, M\}$, $list$)

            Detect pairwise conflicts between the remaining route section after the last perturbation on γ_i^{cur} and routes in $list$

        end while

        Apply the post-processing technique on γ_i^{cur}

        Set $\gamma_i := \gamma_i^{cur}$, add γ_i to $list$

    end if

end for

return $list$

Horizontal detection

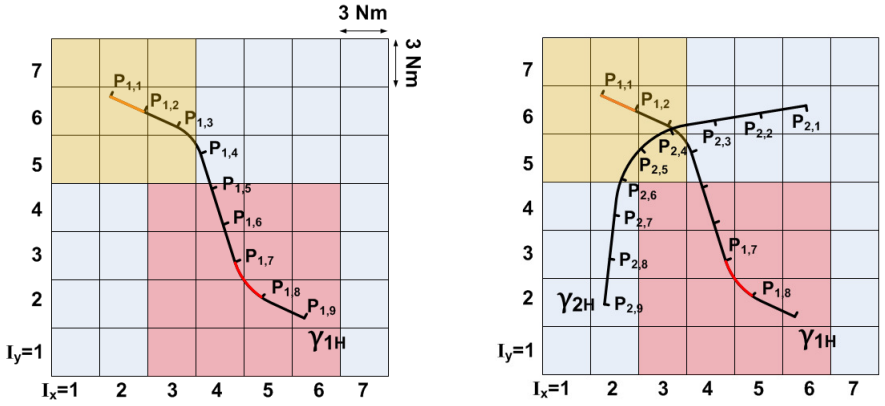
Route projections in the horizontal plane $\gamma_{i_H}, i = 1, \cdots, N$ are in the form of 2D-curves. A

2D-grid is used to cover all the horizontal curves. The idea of using a 2D-grid for conflict detection has been applied in [35]. The dimension of a cell on the 2D-grid is 3NM×3NM, as defined by the horizontal separation norm. Each cell is identified by two indices (I_x, I_y), where I_x is the index on the x-axis and I_y is the index on the y-axis, $I_x, I_y \in \mathbb{N}$.

Each horizontal curve is discretized by using a discretization step δt. In order to guarantee that in each cell occupied by a curve there is at least one discretization point taken on that curve in that cell, a post-processing to add discretization points is applied. Let $0 = t_0^i < t_1^i < t_2^i < \cdots < t_{N_i}^i = 1$ be a subdivision of $[0, 1]$. Each t_k^i corresponds to a discretization point $P_{i,k}$ on curve γ_{i_H}, where $P_{i,k} = \gamma_{i_H}(t_k^i)$. A *curve section* is defined as the section of a curve between two successive discretization points. Let $P_{i,k}$ and $P_{i,k+1}$ be two successive discretization points on curve γ_{i_H}, then the curve section between $P_{i,k}$ and $P_{i,k+1}$ is denoted by $\gamma_{i_{k,k+1}}$. An illustration of curve discretization on a 2D-grid is presented in Fig. 9a).

A *horizontal conflict* occurs when the minimum distance between two curve sections (in the form of arc or segment) on two different curves is less than 3NM. This minimum distance is computed analytically. For γ_{i_H}, the horizontal conflict is detected by successively considering the cells occupied by its curve sections. For each occupied cell, instead of checking the whole grid, only the neighboring non-empty cells are checked, since the horizontal conflict with other curves only occurs in the same or neighboring cells. In such a way, the efficiency of conflict detection is improved. More precisely, suppose that $P_{i,k}$ (respectively, $P_{i,k+1}$) is located in cell $({I_x}^{i,k}, {I_y}^{i,k})$ (respectively, $({I_x}^{i,k+1}, {I_y}^{i,k+1})$). Then the neighboring cells where potential horizontal conflict may occur are $\{(I_x, I_y) | \min({I_x}^{i,k}, {I_x}^{i,k+1}) - 1 \leq I_x \leq \max({I_x}^{i,k}, {I_x}^{i,k+1}) + 1, \min({I_y}^{i,k}, {I_y}^{i,k+1}) - 1 \leq I_y \leq \max({I_y}^{i,k}, {I_y}^{i,k+1}) + 1\}$.

For example, consider the curve γ_{1_H} in Fig. 9a). The discretization points on this curve are $P_{1,k},\ k = 1, \cdots, 9$. The first curve section on γ_{1_H} is $\gamma_{1_{1,2}}$. Both points are in cell $(2, 6)$. Thus the neighboring cells are $\{(I_x, I_y) | 1 \leq I_x \leq 3, 5 \leq I_y \leq 7\}$. When considering the curve section between points $P_{1,7}$ (in cell $(4, 3)$) and $P_{1,8}$ (in cell $(5, 2)$), the neighboring cells are $\{(I_x, I_y) | 3 \leq I_x \leq 6, 1 \leq I_y \leq 4\}$. Figure 9b) presents an example of horizontal conflict detection. Curve γ_{2_H} is discretized by points $P_{2,k}, k = 1, \cdots, 9$. Considering the curve section $\gamma_{1_{1,2}}$, all the neighboring cells are checked,



a) Horizontal curve discretization **b) Horizontal conflict detection**

Fig. 9 An example of horizontal conflict detection.

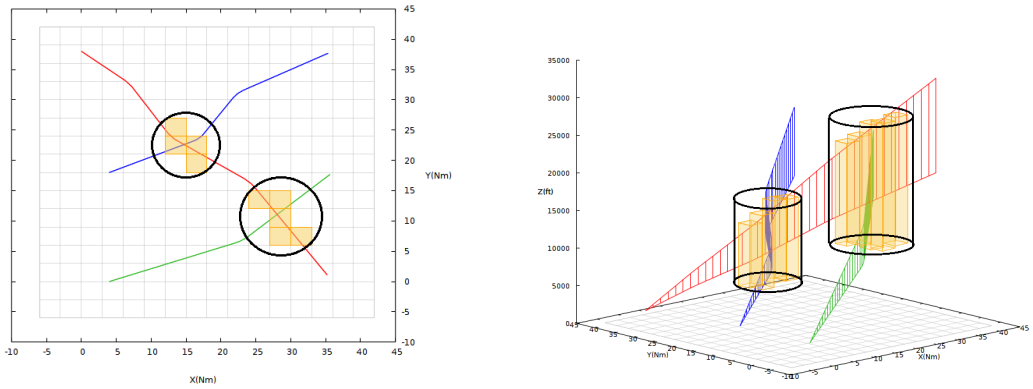
and three discretization points ($P_{2,4}$, $P_{2,5}$, $P_{2,6}$) on γ_{2_H} are found. In order to determine whether a horizontal conflict exists, the pairwise minimum distance between $\gamma_{1_{1,2}}$ and $\gamma_{2_{3,4}}$ (respectively, $\gamma_{2_{4,5}}$, $\gamma_{2_{5,6}}$, $\gamma_{2_{6,7}}$) are measured. By repeating this operation along γ_{1_H}, all the curve sections on γ_{1_H} that violate the horizontal separation constraint are found.

Vertical detection

If a horizontal conflict is detected, a further check in the vertical plane is needed. Denote $H_{inf}^{i,k}$ ($H_{sup}^{i,k}$) the lower (upper) bound of the cross section in the vertical plane at discretization point $P_{i,k}$ on route γ_i, where $H_{inf}^{i,k} = h_{i_{inf}}(t_k^i)$ and $H_{sup}^{i,k} = h_{i_{sup}}(t_k^i)$. Suppose that two curve sections, $\gamma_{i_{k,k+1}}$ and $\gamma_{i'_{l,l+1}}$, on curves γ_{i_H} and $\gamma_{i'_H}$ respectively ($i \neq i'$) have a horizontal conflict. These two curve sections are in *vertical conflict* when neither of the following conditions is satisfied:

$$\begin{aligned} \min\left(H_{inf}^{i,k}, H_{inf}^{i,k+1}\right) - \max\left(H_{sup}^{i',l}, H_{sup}^{i',l+1}\right) > 1000\text{ft} \\ \min\left(H_{inf}^{i',l}, H_{inf}^{i',l+1}\right) - \max\left(H_{sup}^{i,k}, H_{sup}^{i,k+1}\right) > 1000\text{ft} \end{aligned} \tag{13}$$

If one of these conditions is satisfied, which implies that two curve sections are separated vertically at their extremities, then they are separated along the sections. This is because the vertical profiles are monotonically increasing. Even though this detection method brings an additional margin in the separation, it has the advantage of being simple to implement. Finally, if two curve sections violate both horizontal and vertical separation constraints, they are in conflict.



a) Illustration in the horizontal plane **b) Illustration in 3D**

Fig. 10 An example of clustering conflict cells and creating fictitious obstacles.

B. Clustering conflicts and creating fictitious obstacles

After detecting pairwise conflicts between the current route and previously generated routes, each conflict zone is associated with a cylinder-shaped fictitious obstacle (step 12 in Algorithm 2), modeled using the method presented in Sec. II A. In order to generate fictitious obstacles, a *conflict cell* is defined as a cell containing at least one conflicting curve section on the current route. The minimum and maximum altitudes associated with a conflict cell (I_x, I_y) are denoted as $z_{inf}(I_x, I_y)$ and $z_{sup}(I_x, I_y)$ respectively. Let $I \subset \{1, \cdots, N\}$ be the subset denoting the indices of the previously generated routes, and let $\{\gamma_{i_{k,k+1}} | k \in K_i, K_i \subset \{0, \cdots, N_i\},\ i \in I\}$ be the set of curve sections in conflict with the current route in cell (I_x, I_y). Then $z_{inf}(I_x, I_y)$ and $z_{sup}(I_x, I_y)$ can be computed:

$$z_{inf}(I_x, I_y) = \min\{H_{inf}^{i,k} | k \in K_i,\ i \in I\} \tag{14}$$

$$z_{sup}(I_x, I_y) = \max\{H_{sup}^{i,k} | k \in K_i,\ i \in I\} \tag{15}$$

Hence, the geometries related to the conflicting route sections are transferred to the geometries of the corresponding conflict cells. Afterwards, the conflict cells are clustered into groups in such a way that two *adjacent* conflict cells are in the same group. Two cells (I_x^k, I_y^k) and (I_x^l, I_y^l) are adjacent when $|I_x^k - I_x^l| \leq 1$ and $|I_y^k - I_y^l| \leq 1$. Each group of conflict cells is associated with a fictitious obstacle, which is the smallest cylinder containing them. The altitude of the lower (upper) basis of the cylinder is the minimum (maximum) altitude associated with the conflict cells in the

corresponding group, as defined in Eq. (14) and Eq. (15). An example is illustrated in Fig. 10.

C. Perturbing γ_i^{cur}

Algorithm 3 Route perturbation

Require: the current route γ_i^{cur}, the first fictitious obstacle in the ordered list Ω_f, obstacles $\{\Omega_j | j = 1, \cdots, M\}$, the list of previously generated routes $list$

1: Re-generate the portion of route γ_i^{cur} involved in the conflict by perturbing it clockwise and counter-clockwise $\Rightarrow$ obtain γ_i^{cw} and γ_i^{ccw}
2: **if** both γ_i^{ccw} and γ_i^{cw} solve the corresponding conflict **then**
3: Set γ_i^{cur} equals to the one with the shorter length
4: **else if** only one of γ_i^{ccw} and γ_i^{cw} solves the corresponding conflict **then**
5: Set γ_i^{cur} equals to the one that solves the corresponding conflict
6: **else**
7: Re-generate the portion of route γ_i^{cur} involved in conflict by imposing a level flight $\Rightarrow$ obtain γ_i^{lf}
8: **if** γ_i^{lf} solves the corresponding conflict **then**
9: Set $\gamma_i^{cur} := \gamma_i^{lf}$
10: **else**
11: Set γ_i^{cur} equals to the one among γ_i^{ccw}, γ_i^{cw} and γ_i^{lf} which has the minimum length involved in conflict on the perturbed portion of route
12: **end if**
13: **end if**
14: **return** γ_i^{cur}

If one or more fictitious obstacles are generated in step 12 in Algorithm 2, implying that the current route is involved in at least one conflict zone, a route is locally perturbed around the conflict zone corresponding to the first fictitious obstacle $\Omega_f(C_f(x_f, y_f), r_f, z_{f_{inf}}, z_{f_{sup}})$ (step 14 in Algorithm 2). A fictitious obstacle is associated with two decision variables, s_{ij} and t_{ij}. When a fictitious obstacle is active ($s_{ij} = 1$), it can be avoided by three strategies: turn counter-clockwise ($t_{ij} = 0$), turn clockwise ($t_{ij} = 1$) and impose a level flight ($t_{ij} = 2$). Since the Continuous Climb Operations (CCO) and Continuous Descent Operations (CDO) are preferred in TMAs, in our route perturbation strategies horizontal perturbation has a priority over a vertical perturbation. If the horizontal perturbation is able to solve the considered conflict, then there is no need to try the

vertical perturbation. A pseudo-code of the route perturbation is presented in Algorithm 3. In the following, the different route perturbation strategies are presented in more detail.

1. Perturbing a route by counter-clockwise and clockwise turns

In the case of avoiding a fictitious obstacle by a turn ($t_{ij} = 0$ or 1), the perturbed route must be both smooth and flyable, as well as close to the current route γ_i^{cur}. Indeed, in the route perturbation process, the aim is not only to reduce conflicts, but also to keep the perturbed route as close as possible to the optimal route generated by B&B. To meet these objectives, for each fictitious obstacle, four *smoothing obstacles* tangent to both the considered fictitious obstacle and γ_i^{cur} are generated. The radius of the smoothing obstacles r_s ($r_s \geq R_{min}$) is a user-defined parameter and it is the same for all smoothing obstacles. Figure 11 illustrates the four smoothing obstacles given γ_i^{cur} and a fictitious obstacle.

Let γ_i^{ccw} and γ_i^{cw} denote the new routes obtained by perturbing the portion of route γ_i^{cur} involved in a conflict counter-clockwise and clockwise respectively. When the counter-clockwise avoidance strategy is chosen, smoothing obstacles 1 and 3 are taken to build γ_i^{ccw} and the turn direction on both smoothing obstacles is clockwise, as shown in Fig. 12a). The point B_i' is the tangent point of γ_i^{cur} and smoothing obstacle 3. The partially perturbed route $\gamma_i^{ccw,par}$ between the start point A_i and point B_i' is built in the following way:

- First, set A_i and B_i' as the start and end points of $\gamma_i^{ccw,par}$ respectively.
- Second, create a list of obstacles, containing all the active obstacles when building γ_i^{cur}
- Third, add the fictitious obstacle Ω_f to the list of obstacles and order the obstacles in the list according to their projection length to the line (A_iB_i'), as described in Sec. III.
- Then, for obstacles whose order index is lower than that of Ω_f, keep them as active and keep their avoidance strategies; for obstacles whose order index is higher than that of Ω_f, set them as not-considered.
- If an active real obstacle is avoided by a turn and it is intersected with the fictitious obstacle, also set it as not-considered.

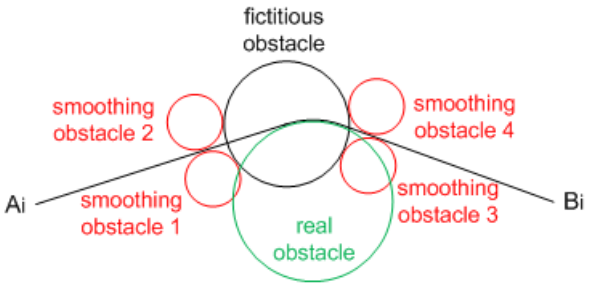
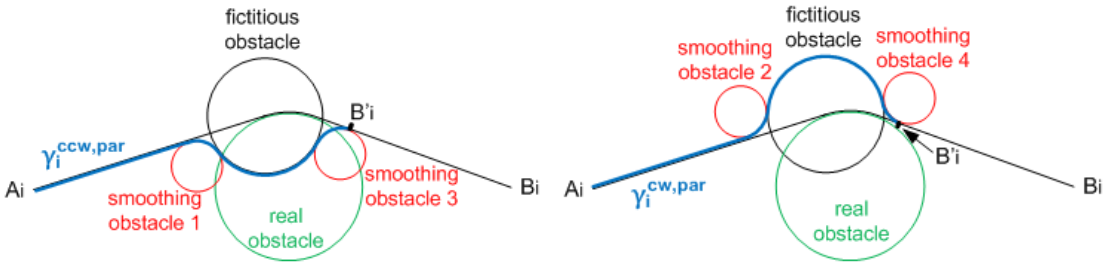


Fig. 11 Configuration of the smoothing obstacles



a) Turn counter-clockwise around the fictitious obstacle

b) Turn clockwise around the fictitious obstacle

Fig. 12 Route perturbation in the horizontal plane.

- Add smoothing obstacle 1 and 3 before and after the fictitious obstacle respectively.
- Finally, build the route in the same way as presented in Sec. III.

After the partial route has been constructed, pairwise conflicts are detected between $\gamma_i^{ccw,par}$ and previously generated routes. If any residual conflict is detected, the radius of the fictitious obstacle is iteratively increased by a user-defined value δ_R, until the conflict is solved or the radius reaches the maximum radius R_{max}. In the case when the radius of the fictitious obstacle reaches R_{max} and a residual conflict still exists, $\gamma_i^{ccw,par}$ is set to be the perturbed route portion satisfying all constraints except route separation and with the smallest length involved in conflicts obtained in the various iterations.

To complete the perturbed route γ_i^{ccw} (*i.e.* build the route section between B_i' and B_i), a list of obstacles is created, including all the real obstacles as well as the fictitious and smoothing obstacles created when perturbing the route. Then, in order to keep the perturbed route section between A_i

and B_i' unchanged in B&B, the indices of the obstacles as well as the values of the decision variables considered when building $\gamma_i^{ccw,par}$ are not changed. Afterwards, the other obstacles not-considered when building $\gamma_i^{ccw,par}$ are ordered according to their projection length on the line (A_iB_i). To build the remaining part of the route, the values of the decision variables for the non-active obstacles are determined in B&B. In the case when the clockwise avoidance strategy is chosen, smoothing obstacles 2 and 4 are taken to build $\gamma_i^{cw,par}$ and their turn direction is counter-clockwise, as presented in Fig. 12b). The approach for building γ_i^{cw} is similar to the one for building γ_i^{ccw}.

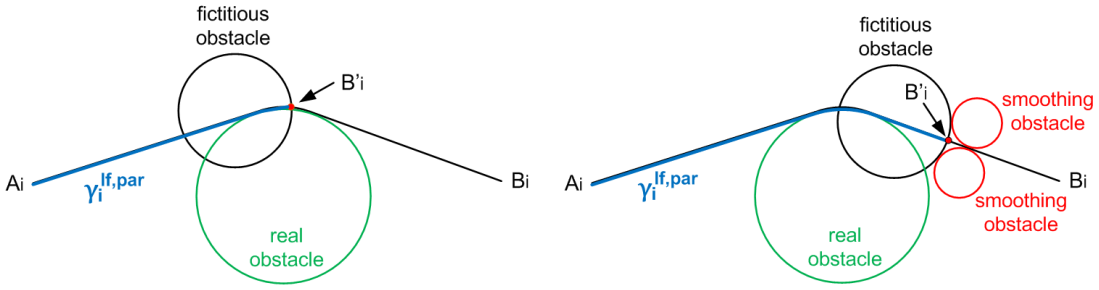
2. Perturbing a route by imposing a level flight

In the case of avoiding fictitious obstacles by imposing a level flight ($t_{ij} = 2$), there are two different scenarios, depending on the relative position between the fictitious obstacle and γ_i^{cur}. Each fictitious obstacle has two intersections with γ_i^{cur} in the horizontal plane, since the corresponding conflict zone is located around γ_i^{cur}. Denote the intersection closer to B_i as B_i'.

The first case occurs when B_i' is located on an arc of γ_i^{cur}, as shown in Fig. 13a). Then the partially perturbed route $\gamma_i^{lf,par}$ connecting A_i and B_i' is built in the following way:

- First, set A_i and B_i' as the start and end points of $\gamma_i^{lf,par}$ respectively.
- Second, create a list of obstacles, consisting of all the active obstacles when building γ_i^{cur}
- Then, add the fictitious obstacle to the list of obstacles and order them according to their projection length to line (A_iB_i').
- For the obstacles in the list whose order index is lower than that of the obstacle where the second intersection point is lying on (including the obstacle itself), keep their avoidance strategies; for the other obstacles, set them as not-considered.
- Finally, build the route in the same way as presented in Sec. III.

The second case occurs when B_i' is located on a segment of γ_i^{cur}. To maintain the horizontal form of the perturbed route, two smoothing obstacles are added, which are tangent to both the



a) Case 1: the second intersection point is located on an arc

b) Case 2: the second intersection point is located on a segment

Fig. 13 Route perturbation in the vertical plane.

fictitious obstacle and the line that the segment is lying on, as shown in Fig. 13b). Then the partially perturbed route $\gamma_i^{lf,par}$ connecting A_i and B_i' is built in the following way:

- First, set A_i and B_i' as the start and end points of $\gamma_i^{lf,par}$.
- Second, create a list of obstacles, consisting of all the active obstacles when building γ_i^{cur}
- Then, add the fictitious obstacle to the list of obstacles and order them according to their projection length to line (A_iB_i').
- For the obstacles whose order index is lower than that of the fictitious obstacle, keep their avoidance strategies; for the other obstacles, set them as not-considered.
- Finally, build the route in the same way as presented in Sec. III.

After building $\gamma_i^{lf,par}$, pairwise conflicts are detected between $\gamma_i^{lf,par}$ and the previous routes. If any residual conflict is detected, the altitude of the lower basis of the fictitious obstacle is iteratively decreased by a user-defined value δ_H, until the conflict is solved or the altitude of the lower basis reaches H_{min}. In the case when the altitude of the lower basis of the fictitious obstacle reaches H_{min} and a residual conflict still exists, $\gamma_i^{lf,par}$ is set to be the perturbed route portion satisfying all constraints except route separation, and with the smallest length involved in conflicts obtained in the iterations.

To complete the route section between B_i' and B_i, a list of obstacles is created, including all the real obstacles as well as the fictitious and smoothing obstacles created when perturbing the route.

These obstacles are then ordered according to their projection length on the line (A_iB_i). In order to keep the form of $\gamma_i^{lf,par}$ unchanged, the values of the decision variables for the obstacles whose order index is lower than that of the fictitious obstacle (including the fictitious obstacle itself) are kept. Afterwards, the methods for completing the remaining route section in the first case (the second intersection point is located on an arc) and in the second case (the second intersection point is located on a segment) are slightly different.

- In the first case, the values of the decision variables of the obstacle where the second intersection point is located is kept. The values of the decision variables for the other obstacles are determined in the modified-B&B method.

- In the second case, the values of the decision variables for the other obstacles (including the two smoothing obstacles) are determined in the modified-B&B method. Meanwhile, only counter-clockwise turn is allowed on smoothing obstacle 1 when it is active and only clockwise turn is allowed on smoothing obstacle 2 when it is active.

D. Post-processing technique (step 17 in Algorithm 2)

In the route perturbation process, each horizontal perturbation is realized by sequentially avoiding a smoothing obstacle, a fictitious obstacle and a second smoothing obstacle, as shown in Fig. 12. Thus the perturbed route section is composed by three successive arcs lying on the three obstacles. This kind of curve is flyable in reality, since a RF leg can join to another RF leg with an opposite turn direction and a different radius [3]. However, in order to simplify the operation for pilots, a post-processing technique on γ_i^{cur} is proposed. For a real SID/STAR, a horizontal perturbation occurs to avoid an obstacle or to avoid another route, afterwards the route usually connects directly to the next route section. Thus, in a SID (respectively, STAR) case, for each horizontal route perturbation, the second (respectively, first) smoothing obstacle is set as non-active and the values of the decision variables for the other obstacles are preserved and the route γ_i^{cur} is re-built. If no more conflict with the previously generated routes is induced by the post-processing, then the new γ_i^{cur} is accepted.

V. Simulation results

In this section some numerical results on the design of multiple routes are presented. The proposed methodology is first tested on an artificially generated problem. Then the design of 15 routes in the Paris CDG airport is presented. A comparison of the designed routes with the standard routes published in Jeppesen charts [36] is also presented. The coordinates are given in a Cartesian Coordinate System where the center is (47° N, 0° E). Tests were run on a Linux platform with a 2.4 GHz processor and 8 GB RAM. The input data, common to all the presented tests, are given in Table 3.

Table 3 Common input data

taking-off slopes $\alpha_{min,TO} = 7\%$ ($\sim 4°$), $\alpha_{max,TO} = 11\%$ ($\sim 6.3°$)
landing slopes $\alpha_{min,LD} = 1.6\%$ ($\sim 0.92°$), $\alpha_{max,LD} = 4.2\%$ ($\sim 2.4°$)
minimum and maximum radius of a RF leg $R_{min} = 5$NM and $R_{max} = 13$NM
maximum number of level flights on each route $N_{max} = 2$
minimum altitude of each level flight $H_{min} = 3500$ft
minimum length of each level flight $L_{min} = 5$NM
discretization step on the horizontal curves $\delta t = 0.01$
radius of smoothing obstacles $r_s = 5$NM
fictitious obstacle radius increment $\delta_R = 1$NM
fictitious obstacle lower basis altitude decrement $\delta_H = 1000$ft
weight coefficient $c_1 = 1$, $c_2 = 0$

A. Test 1, generation of 2 SIDs and 3 STARs ($N = 5$)

The results of tests carried out on an artificially generated problem are presented first. The routes to be built are denoted $\gamma_i, i = 1 \cdots, 5$, where γ_1 and γ_2 are SIDs, and γ_3, γ_4 and γ_5 are STARs. The routes are generated sequentially according to the increasing order of their indices i.

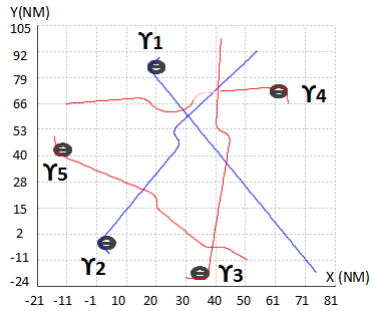
Case 1: 0 obstacles ($N = 5$, $M = 0$)

First, consider the case when there are no obstacles. The simulation result is presented in Fig. 14. Since there are no obstacles, route γ_1, which is generated first, takes the form of a straight line

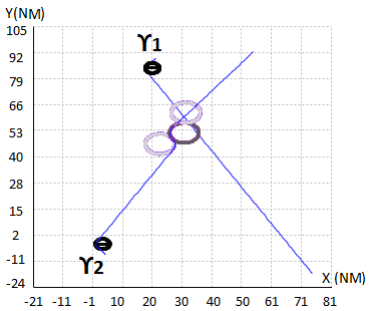
segment. Next, γ_2 is generated. It is perturbed clockwise to pass above γ_1 (Fig. 14b)). Afterwards, γ_3 is computed. It is perturbed counter-clockwise to avoid the conflict with γ_1 (it consequently passes below γ_1 (Fig. 14c))). Next, route γ_4 is generated. As it is in conflict with γ_3, and no perturbation in the horizontal plane is able to solve this conflict, a level flight is imposed to let γ_4 pass below γ_3. An illustration of the level flight on γ_4 in the vertical plane is shown in Fig. 14e). Another conflict occurs when γ_4 meets γ_1, thus a clockwise perturbation is executed on the route section that follows the level flight on γ_4 (Fig. 14d)). Finally, γ_5 is generated and two successive perturbations are performed to avoid the conflict with γ_2 and with γ_3 respectively. The proposed B&B-based approach solves all conflicts with a total simulation time of 2s.

Case 2: 9 obstacles ($N = 5$, $M = 9$)

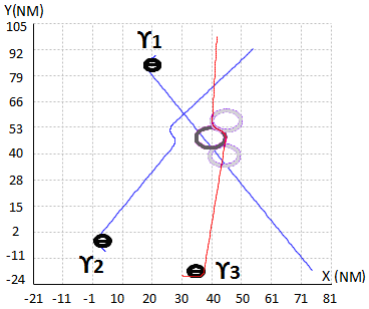
In this case, 9 obstacles are added to the previous configuration (case 1), to test our approach in a more complex situation. With the presence of obstacles, the space for route perturbation is reduced, thus the complexity of the problem is increased. The simulation result is presented in Fig. 15. The layout of the obstacles, as well as the optimal routes obtained are presented in Fig. 15a). Figure 15b) shows that both γ_1 and γ_2 circumnavigate the obstacles before they meet each other due to obstacle avoidance. When their paths cross, their altitudes are not the same as in case 1. Thus, route γ_2 is perturbed clockwise with a bigger turn radius compared to case 1. Due to the presence of obstacles, the γ_3 generated is also different to that generated in case 1. As the shapes of γ_1 and γ_2 are also changed, no conflict occurs between γ_3 and γ_1, nor between γ_3 and γ_2. Next, route γ_4 requires a longer level flight compared to case 1, so that it passes below both γ_3 and γ_1. The profile of γ_4 in the vertical plane is illustrated in Fig. 15e). Note that, in case 1, γ_4 avoids γ_1 by a clockwise turn, however this is not possible here, since the clockwise perturbation intersects an obstacle. Finally, γ_5 is generated and consists of two successive horizontal perturbations, as shown in Fig. 15f). Its shape is different from the one in case 1 (Fig. 14f)), since the post-processing in case 2 does not provide a better solution. The proposed B&B-based approach solves all conflicts with a total computing time of 1.1s.



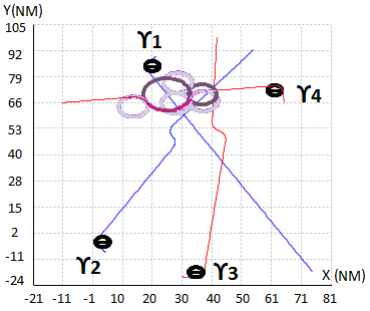
a) Optimal routes illustration



b) Perturbation of γ_2

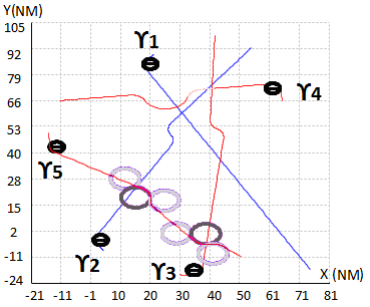


c) Perturbation of γ_3



d) Perturbation of γ_4

e) Perturbation of γ_4 (in the vertical plane)



f) Perturbation of γ_5

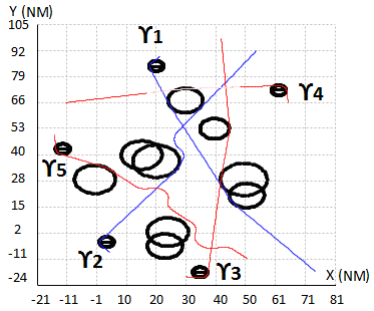
Fig. 14 Test 1: case 1, $N = 5$, $M = 0$, simulation results (runway buffer obstacles are striped, fictitious obstacles are in dark gray, and smoothing obstacles are in light gray.

B. Test 2, generation of multiple routes in the TMA of Paris CDG airport

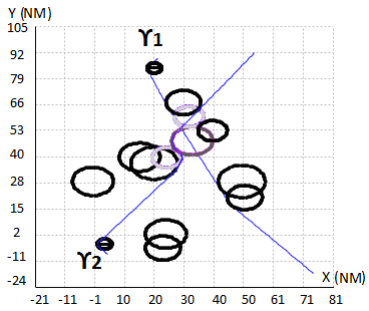
The proposed methodology is now tested on the design of multiple routes in a real TMA, namely the TMA of Paris CDG airport. The radar data of real traffic in this TMA during one day are used. The simulation result is then compared with the standard published SIDs and STARs.

Input data

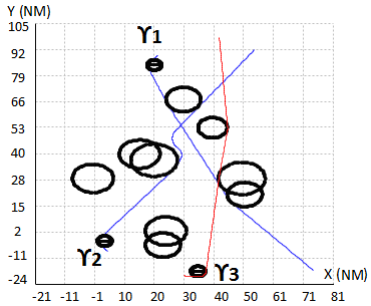
CDG airport has four parallel runways, as shown in Fig. 16a), where the numbering of the 8



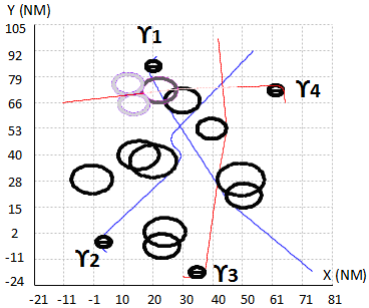
a) Optimal routes illustration



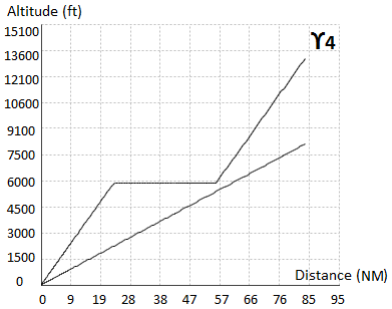
b) Perturbation of γ_2



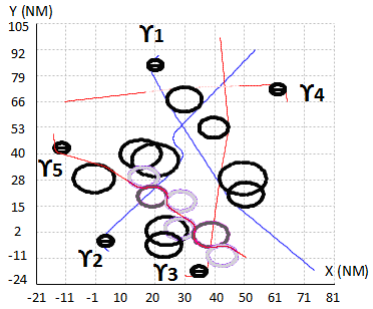
c) Perturbation of γ_3



d) Perturbation of γ_4



e) Perturbation of γ_4 (in the vertical plane)

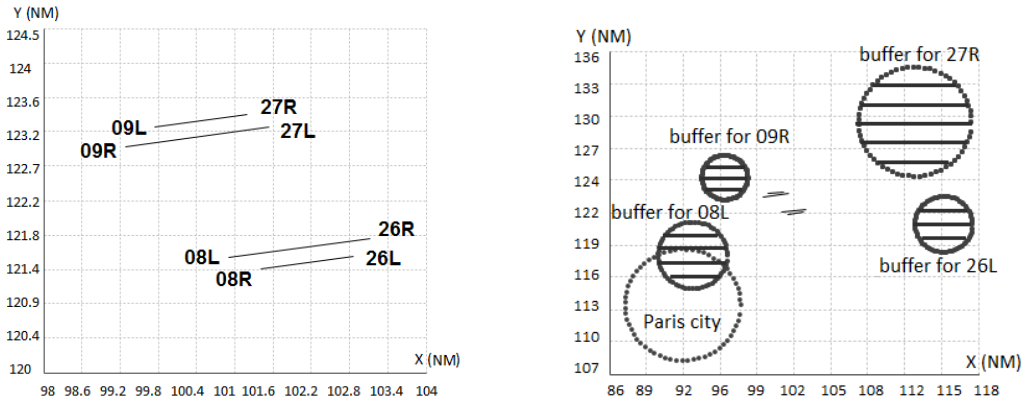


f) Perturbation of γ_5

Fig. 15 Test 1: case 2, $N = 5$, $M = 9$, simulation results (runway buffer obstacles are striped, fictitious obstacles are in dark gray, and smoothing obstacles are in light gray.

thresholds (two thresholds per runway) is also illustrated. In general, two runways are used for take-off and the other two are used for landing. The radar data used for testing correspond to a day when the runway thresholds used for take-off (respectively, landing) are 09R and 08L (respectively, 27R and 26L). The same thresholds are used in our simulation.

The radar data used are illustrated in Fig. 17, where the real traffic arriving at and departing from Paris CDG airport during one day are presented. The light gray areas represent departure



a) Four parallel runways **b) Buffer obstacles and a real obstacle**

Fig. 16 Paris CDG airport configuration.

flights, and the dark gray areas represent arrival flights. It can be seen that the departure and arrival routes are located alternately in order to decrease the interaction between routes. Observe that there are two main conflict zones. One occurs between the arrival flows from the north-west and the departure flows to the north, the other between the arrival flows from the south-west and the departure flows to the south. Based on the radar data, 15 principal routes (8 SIDs and 7 STARs) for jet aircraft are selected from the published Jeppesen charts [36]. These standard routes, as well as the related waypoints, are also presented in Fig. 17, where the gray segments represent SIDs and black ones represent STARs.

In order to compute the traffic load on each principal route, an exit TMA window is defined for each exit point on a SID. Similarly, an entry TMA window is defined for each entry point on a STAR. The traffic load on each principal route is calculated by summing the number of flights passing through the corresponding exit or entry window. In Fig. 17, the selected routes are numbered according to decreasing traffic load. The information on the selected routes, including the threshold used and traffic load (in percentage) are presented in Table 4.

In our simulation, the design of 15 routes (denoted as $\gamma_i, i = 1, \cdots, 15$) is considered. Their start and end points, as well as traffic loads correspond to that of the 15 selected standard principal routes. More precisely, in a SID case, the start and end points of γ_i are the same as that of the standard route i, while in a STAR case, the end point of γ_i corresponds to the TMA entry point

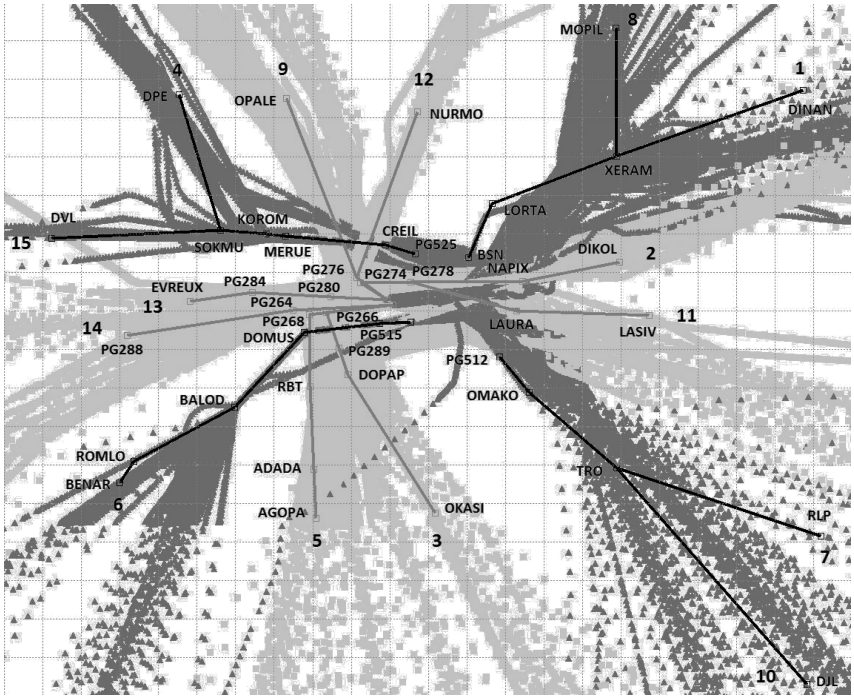


Fig. 17 One day radar data of Paris CDG airport, and principal SIDs and STARs. (The light gray areas represent the departure flights, and the dark gray ones are the arrival flights. The gray segments are the SIDs, and the black ones are the STARs.)

Table 4 Test 2: traffic load and waypoints of the selected standard routes

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
route type	STAR	SID	SID	STAR	SID	STAR	STAR	STAR	SID	STAR	SID	SID	SID	SID	STAR
threshold	27R	09R	08L	27R	08L	26L	26L	27R	09R	26L	09R	09R	09R	08L	27R
traffic (%)	12.77	10.82	10.72	8.7	7.44	7.33	7.16	6.97	5.63	5.61	4.9	4.76	3.38	2.33	1.48

on the standard route i, and the start point is the FAF of the corresponding runway threshold. As described in Sec. II, each route to be designed is associated with a buffer obstacle. The characteristics of the buffer obstacles are presented in Table 5. It is assumed that the routes using the same runway use the same buffer obstacle. The TMA of CDG airport is located in a relatively simple geographical environment, the only area to be avoided is Paris city, for the sake of noise abatement. This area is modeled as an obstacle. Thus $M = 1$ in our simulation, and the characteristics of this obstacle are $(x_1, y_1, r_1, z_{1_{inf}}, z_{1_{sup}}) = (92.43\text{NM}, 113.16\text{NM}, 5\text{NM}, 0, 50000\text{ft})$. The buffer obstacles as well as

Table 5 Test 2: runway thresholds and the corresponding buffer obstacles

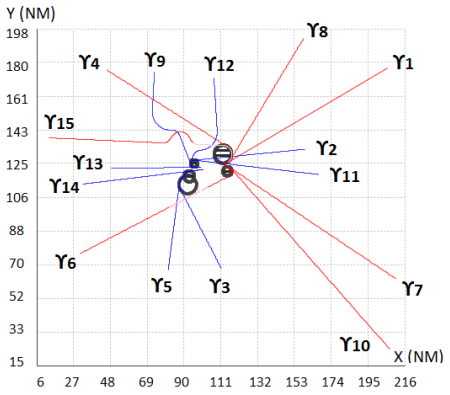
threshold	buffer obstacle		
	$(x_{bi}, y_{b_i}, r_{bi})$ (NM)	$(z_{bi_{inf}}, z_{bi_{sup}})$ (ft)	$t_{\Omega_{b_i}}$
09R	(96.07, 124.59, 2)	(0, 50000)	1
08L	(93.29, 117.61, 3)	(0, 50000)	0
27R	(112.55, 129.61, 4.9)	(0, 50000)	0
26L	(115.07, 120.4, 2.5)	(0, 50000)	1

the obstacle representing Paris city are illustrated in Fig. 16b).

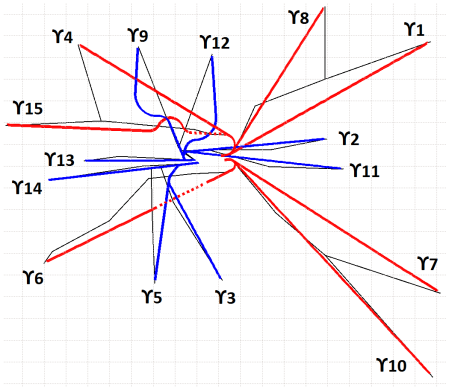
Simulation results

Routes are built sequentially according to decreasing order of traffic load. The simulation result is presented in Fig. 18. The presented results show that two main conflict zones occur in the simulation. One conflict zone is between departure routes γ_9, γ_{12} and arrival routes γ_4, γ_{15}, the other is between departure routes γ_3, γ_5 and arrival route γ_6. This is coherent with our analysis of the real radar data. For departure routes γ_9, γ_{12} and arrival routes γ_4, γ_{15} in the simulation, route γ_4 is the first of the four to be generated, since it has a higher traffic load. Routes γ_9 and γ_{12} are generated next. In order to avoid the conflict with γ_4, route γ_9 is perturbed clockwise and route γ_{12} is perturbed counter-clockwise. Route γ_{15} is the final route to be generated, and is initially in conflict with both γ_{12} and γ_9. To avoid γ_{12}, a level flight is imposed, and to avoid γ_9, a counter-clockwise turn is applied. These four routes are illustrated in 3D in Fig. 18c), and the vertical profile of γ_{15} is illustrated in Fig. 18d). For departure routes γ_3, γ_5 and arrival route γ_6, route γ_6 is generated after γ_3, γ_5, and a level flight is imposed on γ_6, so as to avoid conflict with previously generated routes. These three routes are illustrated in 3D in Fig. 18e), and the vertical profile of γ_6 is illustrated in Fig. 18f). The computing time for obtaining the 15 optimal routes is about 9.8s.

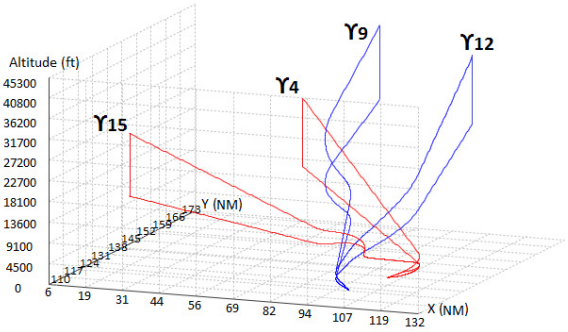
Figure. 18b) represents the comparison between the selected standard routes and our simulation results in the horizontal plane. The thin curves represent the selected standard SIDs and STARs, and the thick curves correspond to the optimal routes obtained using the proposed method, where



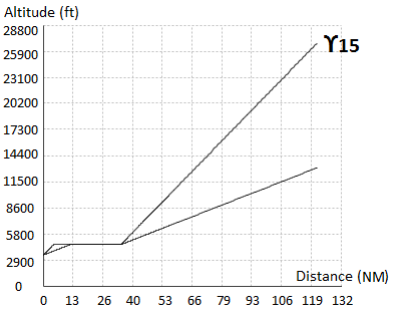
a) Illustration in the horizontal plane



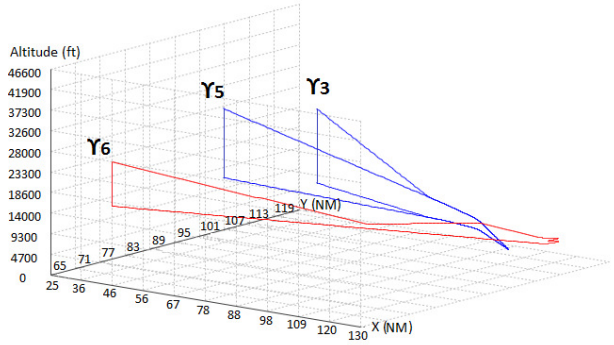
b) Comparison with standard routes



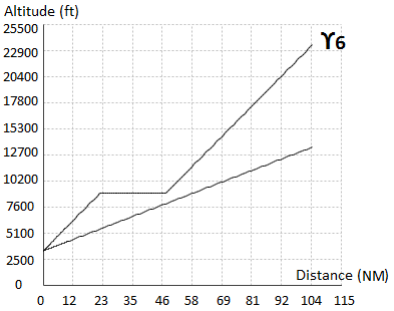
c) Illustration of γ_4, γ_9, γ_{12} and γ_{15}



d) Illustration of γ_{15} in the vertical plane



e) Illustration of γ_3, γ_5 and γ_6



f) Illustration of γ_6 in the vertical plane

Fig. 18 Test 2: simulation results.

the dashed sections represent level flights. The total length of the 15 selected standard routes is 1358.26NM, and it is reduced by around 45NM using the proposed B&B-based approach. Except routes γ_9, γ_{12} and γ_{15}, which are perturbed to avoid conflicts, the other routes obtained by the B&B-based approach are all shorter than the standard routes. This is especially true for the routes with relatively heavy traffic load, since they have a priority order in the design and therefore the

generated routes are straight line segments. The length of level flight on γ_6 is 25.7NM, longer than the corresponding level flight on the standard route, which is 20NM (according to [36]). The length of level flight on γ_{15} is 30Nm, again longer than the corresponding level flight on the standard route, which is about 25NM (according to the radar data). On average, CDG airport accommodates a total of 1300 flights per day (including both departures and arrivals). By using the optimal routes obtained in our simulation, the total flown distance is reduced by 5889NM (10906km) per day, and the total length of level flights is increased by 640NM (1185km) per day.

From an economic point of view, this can lead to considerable savings. Consider an A320 aircraft as an example. The average fuel consumption of one engine in the take-off and landing phases is about 6kg/km. Thus, the total fuel consumption is reduced by about 131ton per day, leading to about \$65,000 in the reduction of jet fuel costs (assuming a jet fuel price of 61.3\$/bbl (497.5\$/ton) [37]). Since the length of level flights is only slightly increased, and the affected traffic load is only 8.8%, the impact of the increase in the total length of level flights on fuel consumption is negligible. In addition, because aircraft flying on different routes are automatically separated, the workload for the controllers is reduced. This may lead to a reduction in the expenses related to ground services.

Readers interested by the detailed input data (including the coordinates of the start and end points, characteristics of the buffer obstacles, and traffic load of each route), or detailed numerical results (including the initial length of route sections involved in conflicts, the optimal route length, the length of level flight on each optimal route) of tests 1 and 2, may refer to [38] for more information. More simulation results on the design of a single or multiple routes are presented in [39].

VI. Conclusions

In this paper, a decision support methodology is proposed for the design of multiple Standard Instrument Departure routes (SIDs) and Standard Terminal Arrival Routes (STARs) in a Terminal Maneuvering Area (TMA) at a strategic level. The objective of the proposed approach is to design SIDs/STARs that minimize total route length, while taking into account operational and environmental constraints; in particular, obstacle avoidance and route separation constraints. A route is modeled as a curve formed by segments and arcs in the horizontal plane, associated with a cone

enveloping all ascent (or descent) aircraft profiles in the vertical plane. Three different ways to avoid an obstacle are proposed: turning clockwise, turning counter-clockwise, and imposing a level flight under the obstacle. These maneuvers are compatible with operational requirements and correspond to what is currently implemented.

The problem is modeled as an Integer Programming optimization problem. To design a single optimal route, a Branch and Bound (B&B) method is developed, whose branching strategies correspond to the three ways obstacles can be avoided. To design multiple routes, the proposed B&B method is used to sequentially generate routes, and a route-modification step is introduced to deal with conflicts between routes. The proposed methodology is first tested on an artificially generated problem, and then applied to the design of 15 routes in the TMA of Paris Charles-de-Gaulle (CDG) airport. The routes obtained are continuous and smooth. These routes result in a reduction in total route length compared with the published standard charts in the case of Paris CDG airport, which in turn results in a sizable reduction in jet fuel consumption.

Acknowledgments

This work has been partially supported by Civil Aviation University of China (CAUC), by National Natural Science Foundation of China (NNSFC) through grant NNSFC 61201085, by the Fundamental Research Funds for the Central University of China through grant number ZXH2012H005, and by French National Research Agency (ANR) through grant ANR 12-JS02-009-01 "ATOMIC". The authors would like to thank Serge Roux and Alana Moore for interesting discussions and feedback on the manuscript. The authors would also like to thank Gilles Baroin for his help in formatting the figures.

References

[1] International Civil Aviation Organization (ICAO), "Aircraft Operations (Doc 8168)," 2006.

[2] Eurocontrol, "EUROCONTROL Long-Term Forecast: IFR Flight Movements 2010-2030," Dec. 2010.

[3] Civil Aviation Safety Authority (CASA) Australia, "Performance Based Navigation (PBN) Operational Approval Handbook," Aug. 2010.

[4] Eurocontrol, "European Airspace Concept Handbook for PBN Implementation," June 2013.

[5] Gallina, P. and Gasparetto, A., "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, 2000, pp. 237–262. doi:10.1023/A:1008168615430.

[6] Souissi, O., Benatitallah, R., Duvivier, D., Artiba, A., Belanger, N., and Feyzeau, P., "Path planning: A 2013 survey," *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, Oct. 2013, pp. 1–8.

[7] Delahaye, D., Puechmorel, S., Tsiotras, P., and Feron, E., "Mathematical Models for Aircraft Trajectory Design: A Survey," *Air Traffic Management and Systems: Selected Papers of the 3rd ENRI International Workshop on ATM/CNS (EIWAC2013)*, 2014, pp. 205–247. doi:10.1007/978-4-431-54475-3_12.

[8] Lozano-Pérez, T. and Wesley, M. A., "An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles," *Communications of the ACM*, Vol. 22, No. 10, ACM, NY, USA, Oct. 1979, pp. 560–570. doi:10.1145/359156.359164.

[9] Liu, Y.-H. and Arimoto, S., "Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Obstacles," *International Journal of Robotics Research*, Vol. 11, No. 4, Sage Publications, Inc., Thousand Oaks, CA, USA, Aug. 1992, pp. 376–382. doi:10.1177/027836499201100409.

[10] Pocchiola, M. and Vegter, G., "Minimal tangent visibility graphs," *Computational Geometry: Theory and Applications*, Vol. 6, No. 5, 1996, pp. 303 – 314. doi:10.1016/0925-7721(95)00016-X.

[11] Kim, D. S., Yu, K., Cho, Y., Kim, D., and Yap, C., "Shortest paths for disc obstacles," *Computational Science and Its Applications – ICCSA 2004*, edited by A. Laganá, M. L. Gavrilova, V. Kumar, Y. Mun, C. J. K. Tan, and O. Gervasi, 2004, pp. 62–70. doi:10.1007/978-3-540-24767-8_7.

[12] Eele, A. and Richards, A., "Path planning with avoidance using nonlinear branch and bound optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 32, 2009, pp. 384–394. doi:10.2514/1.40034.

[13] Pierre, S., Delahaye, D., and Cafieri, S., "Aircraft Trajectory Planning with Dynamical Obstacles by Artificial Evolution and Convex Hull Generations," *2015 ENRI International Workshop on ATM/CNS (EIWAC2015)*, ENRI, Nov. 2015,

[14] Jimmy, K., Changkil, L., and Joseph S. B., M., "Turn-Constrained Route Planning for Avoiding Hazardous Weather," *Air Traffic Control Quarterly*, Vol. 14, No. 2, 2006, pp. 159–182.

[15] Sharir, M. and Schorr, A., "On Shortest Paths in Polyhedral Spaces," *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, 1984, pp. 144–153. doi:10.1145/800057.808676.

[16] Jiang, K., Seneviratne, L., and Earles, S., "Finding the 3D shortest path with visibility graph and minimum potential energy," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, July 1993, pp. 679–684. doi:10.1109/IROS.1993.583190.

[17] Canny, J. and Reif, J., "New lower bound techniques for robot motion planning problems," *28th Annual Symposium on Foundations of Computer Science (SFCS 1987)*, Oct. 1987, pp. 49–60. doi:10.1109/SFCS.1987.42.

[18] Gianazza, D., Durand, N., and Archambault, N., "Allocating 3D-trajectories to air traffic flows using A* and genetic algorithms," *Proceedings of the International Conference on Computational Intelligence for Modelling, Control, and Automation (CIMCA04)*, 2004,

[19] Pfeil, D. M., *Optimization of airport terminal-area air traffic operations under uncertain weather conditions*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2011.

[20] Zhou, J., Cafieri, S., Delahaye, D., and Sbihi, M., "Optimization of Arrival and Departure Routes in Terminal Maneuvering Area," *6th International Conference on Research in Air Transportation (ICRAT2014)*, May 2014,

[21] Zhou, J., Cafieri, S., Delahaye, D., and Sbihi, M., "Optimizing the design of a route in Terminal Maneuvering Area using Branch and Bound," *2015 ENRI International Workshop on ATM/CNS (EIWAC2015)*, Nov. 2015,

[22] Gianazza, D. and Durand, N., "Separating air traffic flows by allocating 3D-trajectories," *23rd Digital Avionics Systems Conference (DASC 2004)*, Vol. 1, Oct. 2004, pp. 2.D.4–21–13. doi:10.1109/DASC.2004.1391275.

[23] Gianazza, D. and Durand, N., "Assessment of the 3D-separation of Air Traffic Flows," *6th USA/ Europe Air Traffic Management Research and Development Seminar (ATM 2005)*, 2005,

[24] Polishchuk, V., "Generating Arrival Routes with Radius-to-Fix Functionalities," *7th International Conference on Research in Air Transportation (ICRAT 2016)*, 2016,

[25] Zhou, J., Cafieri, S., Delahaye, D., and Sbihi, M., "Optimal Design of SIDs/STARs in TMA Using Simulated Annealing," *35th Digital Avionics Systems Conference (DASC2016)*, 2016,

[26] Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, Vol. 1, Springer-Verlag New York, Inc., Secaucus, NJ, USA, Dec. 1959, pp. 269–271. doi:10.1007/BF01386390.

[27] Bellman, R., "On a Routing Problem," *Quarterly of Applied Mathematics*, Vol. 16, No. 1, April 1958, pp. 87–90.

[28] Lawler, E. L. and Wood, D. E., "Branch-And-Bound Methods: A Survey," *Operations Research*, Vol. 14, No. 4, INFORMS, 1966, pp. 699–719.

[29] Hart, P., Nilsson, N., and Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, July 1968, pp. 100–107. doi:10.1109/TSSC.1968.300136.

[30] Dreo, J., Petrowski, P., Siarry, P., and Taillard, E., *Metaheuristics for hard optimization*, Springer-Verlag Berlin Heidelberg, Berlin, 2005, pp. 1–19.

[31] Federal Aviation Administration (FAA), "The United States Standard for Area Navigation (RNAV) (Order 8260.54A)," 2007.

[32] Federal Aviation Administration (FAA), "United States Standard for Performance Based Navigation (PBN) Instrument Procedure Design (Order 8260.58A)," 2016.

[33] Eurocontrol, "User Manual for the Base of Aircraft Data (BADA) Revision 3.9," April 2011.

[34] Garmin, "Garmin Radius to Fix Leg Project Report (in support of FAA memorandum of agreement no.DTFAWA-11-A-80009)," 2013.

[35] Chaimatanan, S., Delahaye, D., and Mongeau, M., "A Hybrid Metaheuristic Optimization Algorithm for Strategic Planning of 4D Aircraft Trajectories at the Continental Scale," *IEEE Computational Intelligence Magazine*, Vol. 9, No. 4, Nov 2014, pp. 46–61. doi:10.1109/MCI.2014.2350951.

[36] Jeppesen, "LFPG (Charles-De-Gaulle)," 2011.

[37] International Air Transport Association (IATA), "Fuel Price Analysis," `http://www.iata.org/publications/economics/fuel-monitor/Pages/price-analysis.aspx`.

[38] Zhou, J., `http://www.recherche.enac.fr/~cafieri/SID-STAR-data.pdf`.

[39] Zhou, J., *Optimisation des procédures de départ et d'arrivée dans une zone terminale*, Ph.D. thesis, Université Paul Sabatier, Toulouse, 2017.