

Light Propagation algorithm for aircraft trajectory planning

Nour Elhouda Dougui, Daniel Delahaye, Stéphane Puechmorel, Marcel
Mongeau

► **To cite this version:**

Nour Elhouda Dougui, Daniel Delahaye, Stéphane Puechmorel, Marcel Mongeau. Light Propagation algorithm for aircraft trajectory planning. ACC 2011, American Control Conference, Jun 2011, San Francisco, United States. pp 2143 - 2147. hal-00938501

HAL Id: hal-00938501

<https://hal-enac.archives-ouvertes.fr/hal-00938501>

Submitted on 6 Mar 2014

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

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

Light Propagation Algorithm for Aircraft Trajectory Planning

Nour Elhouda Dougui and Daniel Delahaye and Stéphane Puechmorel and Marcel Mongeau

Abstract—The need for increasing air traffic capacity motivates 4D trajectory planning concept. In order to generate conflict-free 4D trajectories, we introduce a new concept based on light propagation modeling algorithm. This algorithm is a wavefront propagation method that yields a natural solution for the path planning problem specifically in the case of air traffic congestion.

I. INTRODUCTION

During recent years, several methods have been proposed to find an optimal solution that could solve conflicts in air traffic. The aim of these methods is to find for each aircraft, an optimal 4D trajectory that avoids conflicts with other aircraft, reaches the destination point and optimizes a cost function which depends on the travel duration and on the cost index (a coefficient that takes into account fuel consumption). There are mainly two classes of methods to address this problem: genetic algorithms [1] and navigation-function based approach [2]. Each one provides only a partial solution to the problem.

The first one, genetic algorithms, consists in generating a new population of aircraft trajectories from a base population using three basic operators: selection, mutation and crossover in order to improve the cost function. This process is iterated until the cost function is no longer improved. The state space is a set of finite maneuvers, which are straight lines, turning points (changing an aircraft heading and then bringing it back on its initial trajectory) and offsets (inducing a lateral shift from the initial trajectory). Those maneuvers are the ones used by air traffic controller. Genetic algorithms generate trajectories with feasible operational maneuvers and with velocities within bounded ranges. They can reach asymptotically optimal solution, but for a given computing time, a feasible (conflict-free) solution is not guarantee.

The second method, based on navigation functions, consists in using an electrostatic modeling of the problem: an electron (which has a negative charge) is subject to an electric field, and is attracted by a positive charge which represents the goal and is pushed away by negative charges which represent obstacles. Thus, the electron is going to move

towards the goal and steered by the resultant electric fields. The aircraft (a virtual electron) is represented by a point in 3D space. If a mathematical function of potential fields can be built to model adequately the destination charge and the distribution of the obstacle charges, then the virtual forces applied on such virtual electron, initially positioned at the departure point, can be computed. This produces a trajectory which connects the departure point with the destination while avoiding obstacles (the other aircraft).

Navigation functions have already demonstrated their effectiveness in motion planning with guaranteed collision avoidance and convergence towards the goal configuration (reach the destination point with the right orientation). However, they do not take into account the constraints imposed by ATM, such as bounded speed, smooth trajectory and time constraints. The objective of our approach, based on an optical analogy, is to find for each aircraft a feasible (relevant to ATM constraints) optimal 4D trajectory, avoiding conflicts and which minimizes a criterion based on a local metric.

In the next section, we present our method. Numerical results are presented in section III.

II. LIGHT PROPAGATION ALGORITHM

Fermat (1601-1665) developed his *Principle of Least Action* [6]: *The path of a light ray connecting two points is the one for which the time of transit, not the length, is a minimum.* Based on this principle, we introduce an optimal path planning algorithm, the *Light Propagation algorithm* (LPA), which compute smooth geodesic trajectories in environments with static or dynamic obstacles. This algorithm mimic light propagation between a starting point towards a destination point for which obstacles are modeled by high-index areas. By controlling the index landscape, it is possible the ensure that the computed trajectories meet the speed constraints and are at specified minimum distance from obstacles. LPA is adjusted from the aircraft point of view. It is assumed that aircraft knows the surrounding aircraft trajectories (trajectories of other aircraft is a given input of our algorithm).

Assume that objective function is an application associating a positive real value to a curve of class C^1 of \mathbb{R}^3 . Such value is computed by integrating a local metric along the curve. We can thus represent length, travel time or the cost associated with a trajectory by a suitable choice of local metric. Determining an optimal trajectory will therefore reduce to search a geodesic which is the shortest path between two points on the space regarding the local metric.

In this algorithm, we use the well-known fact that a light ray trajectory is a geodesic when considering the

Nour Elhouda Dougui is with the Applied Mathematics Laboratory, French Civil Aviation University, 31055 Toulouse, France nour@recherche.enac.fr

Daniel Delahaye is with the Applied Mathematics Laboratory, French Civil Aviation University, 31055 Toulouse, France delahaye@recherche.enac.fr

Stéphane Puechmorel is with the Applied Mathematics Laboratory, French Civil Aviation University, 31055 Toulouse, France puechmor@recherche.enac.fr

Marcel Mongeau is with the Toulouse Institute of Mathematics UMR 5219, Paul Sabatier University, F-31062 Toulouse cedex 9, France mongeau@math.univ-toulouse.fr

environment refractive index as a local metric. To represent congestion areas and conflicts in air traffic management, we consider the refractive index as a measure of congestion or traffic complexity. We select a barrier index value in the prohibited areas and in the protection volumes surrounding each aircraft. The light will be slowed down in congested areas, but despite this, it can pass through. However, it will be completely blocked by aircraft protection volumes, which ensures conflict free-situations. We compute the environment index associated to a given congested area using a model based on Lyapunov exponents [3].

We seek to find numerically the path followed by light between two points in 4D (3D+time) space provided with a refractive index map. For this, we introduce a light source at the departure point. Then, we simulate the light propagation from this source using the wave propagation theory of light proposed by the Dutch scientist Christiaan Huygens in 1678. The Huygens principle [7] can be stated as follows: any point on a wavefront can be considered as the source of tiny wavelets that propagate forward at the same speed as the wave, and the new wavefront is the envelope of all the wavelets (that is to say, the tangent to these wavelets) as shown in Figure 1.

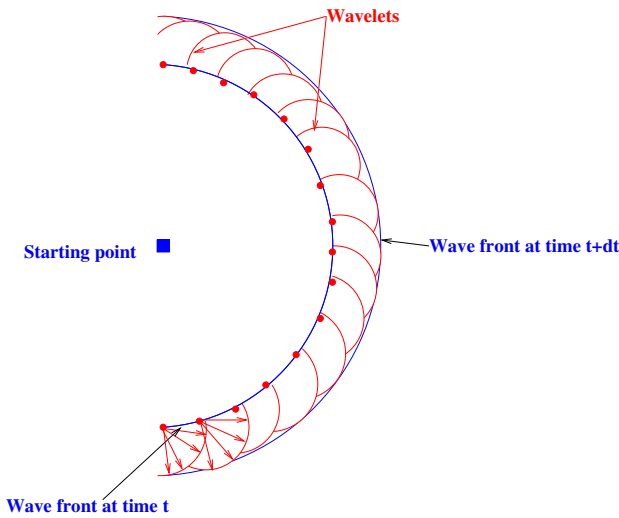


Fig. 1. Huygens principle used to determine the wave front at time $t + dt$ from the wave front at time t

LPA is then a wavefront propagation algorithm in 4D. This wavefront is issued from the starting point and is discretized in space: we launch several light rays in various directions from this point. It is propagated in time dimension using time step dt . The path of the first ray that reaches the arrival point corresponds to an approximation of a geodesic.

A. Branch-and-bound Algorithm

We implement the LPA within a *branch-and-bound algorithm* (B&B) [8], a classical framework for solving discrete optimization problems. B&B represents the set of all possible solutions by the root of an enumeration tree. Procedures to obtain lower and upper bounds for the optimal value of our

objective function (trajectory travel time) are first applied to the root. If these two bounds are equals, then the optimal solution is found, and the algorithm stops. Otherwise, the solution set is partitioned into several sub-problems (new children nodes). The method is then applied recursively on the corresponding sub-problems, generating a tree. If an optimal solution is found for a sub-problem, it is feasible but not necessarily optimal for the original problem. Based on the bound, feasible solutions are used to eliminate sets of partial solutions, reducing the size searching tree (cutting branches). The search goes on until all the nodes are explored or eliminated.

B. Branch and Bound applied to the Light Modeling Algorithm

For the implementation of our light propagation case, we compute an approximate lower bound for a given node as follows. We first compute a duration, "*TimeToDest*", for the remaining time to reach the destination. This duration is a weighted sum of two terms given by Formula (1), where α is a weighting parameter. The first one, "*integTime*", is the time to reach destination considering the refractive index along the direct straight line route. The second one, "*maxSpeedTime*", is the time needed to reach destination in direct route with the maximum speed assuming unit refractive index.

$$TimeToDest := \alpha * integTime + (1 - \alpha) * maxSpeedTime \quad (1)$$

The approximate lower bound (approxLB) that we are proposing is then the summation of *TimeToDest* and the time that was required to reach the current node from the origin *TimeToNode* (see Figure2):

$$approxLB := TimeToNode + TimeToDest$$

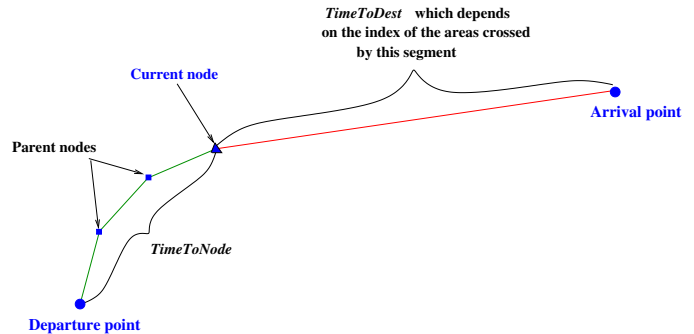


Fig. 2. The lower bound compute. This bound (for current node) is computed by adding the time required to reach such a node and the time to reach the arrival point from this node using direct trajectory (straight line). Those times are computed by using the encountered index value

Branching, in our context is based on what will allow to simulate the wavefront propagation. Indeed, we throw light beams in straight lines from the current node. We restrict the emission of light beams to directions in the half space between the current node and the arrival point (common aircraft practice will not consider other directions except in

the neighborhood of the origin or destination airport). These beams are launched in all such directions with discretization angle steps given by $d\theta$ and $d\phi$. A beam propagates in one direction with a velocity that depends on the refractive indices of the media through which it passes, reaching a son node of the current node after a time step dt . These algorithmic parameters $d\theta$, $d\phi$ and dt are set by the user. All nodes that have the same depth in the resulting tree will represent the same wavefront.

Next, LPA browses the search tree to find a minimal-time trajectory (an approximate geodesic). This can be done in different ways. We choose a strategy whose priority is to find quickly a feasible solution with depth-first search (DFS): choose a node for which children have not yet been generated, with deepest level in the search tree. The memory requirement in terms of number of sub-problems (stored at any given time) is bounded above by the number of levels in the search tree multiplied by the maximum number of children of any node, which is in our context a manageable number. The drawback of such approach is that nodes which are far from being optimal, may yield large amount of unnecessary bounding computations. In order to avoid such drawback in our case, DFS is combined with the following selection strategy: choose the node that has the best lower bound among the nodes at the same level in the search tree. In other words, we combine DFS as the overall principle with best first search (BFS) as a secondary selection criterion.

The main steps of the algorithm are as follows:

1. Set $TrajSolution := \emptyset$. Set $upperBound := \infty$.
2. $LaunchRays(Departure\ point)$.
3. While there is still unexplored nodes in the tree choose a node N according to the DFS and the BFS as described above. Then :
 - If $distance(N, destination\ point) \leq \epsilon$ and value of node $N \leq upperBound$ then $TrajSolution := Set\ of\ ascendant\ points\ that\ lead\ to\ N$ and $upperBound := value\ of\ node\ N$.
 - $LaunchRays(N)$.

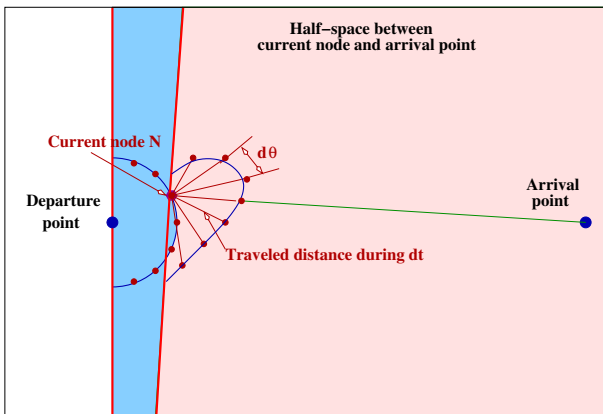


Fig. 3. Launching rays from current node N in 2D+time.

The sub-procedure " $LaunchRays(N)$ " for a node N is defined as below (see Figure 3):

- i. Discretize the half space between node N and the destination point with a time step dt and angle steps $d\theta$ and $d\phi$.
- ii. Determine new child nodes and their values using the following rule:
For any light beam, if it goes into a region with index n , its velocity inside this region is $v = \frac{c}{n}$, where c is the light speed in vacuum.
- iii. Remove node N from the tree and add its children.

In the following section, we will see the numerical results of a simplified version of the algorithm (2D), first with a static refractive index and then with a dynamic refractive index.

III. NUMERICAL RESULTS

In all our simulations, we work on a 3.2 GHz machine running under Debian Linux operating system, 1024 KB of RAM. The software has been developed in JAVA.

We use a coordinate system that is scaled with separation standards. Thus, we use an (x,y) grid with a standard horizontal separation (5 Nm¹) unit. The index map used is a square of (40*40) standard horizontal separation.

In the sub-procedure " $LaunchRays(N)$ ", we fix the time step dt to the duration needed to fly a half horizontal standard separation distance and the sampling angle $d\theta$ to $\frac{\pi}{10}$. We chose empirically the coefficient $\alpha := 0.9$ in the formula (1).

A. Results in 2D

In order to validate our methodology, we first test LPA in \mathbf{R}^2 space (no time involved) equipped with a static refractive index map. The goal is to find geodesics between two given points in \mathbf{R}^2 .

We test several refractive index functions. For instance, the index function used in Figure 4 is $\sum_{i=0}^4 e^{-\frac{((x-a_i)^2+(y-b_i)^2)}{k}}$ with different values of a_i , b_i and k in each of the four cases. This index function is continuous, with high values represented in red and low values in blue.

All four solution trajectories, computed by LPA and displayed on Figure 4, are found in less than 5 seconds of CPU time. As it can be seen, each trajectory produced by LPA is a geodesic approximation that avoids high index areas and passes through low-value "valleys", as expected. In Figure 4(b), the trajectory goes through a relatively congested area instead of bypassing it completely through the blue area above. This behavior can be explained by the fact that direct path, although it slows down the aircraft is more advantageous than a long detour with a higher aircraft velocity.

Trajectories produced by LPA guarantee a speed lower bound, which is crucial in our ATM applications. Moreover, these trajectories are, by construction, sequences of segments and arcs, that can be managed by the FMS.

Although the test problems we consider here are academic, LPA can be applied to real-world aircraft trajectories in pre-tactical planning to avoid bad-weather or congested areas

¹One Nautical Mile (Nm)= 1852 meters

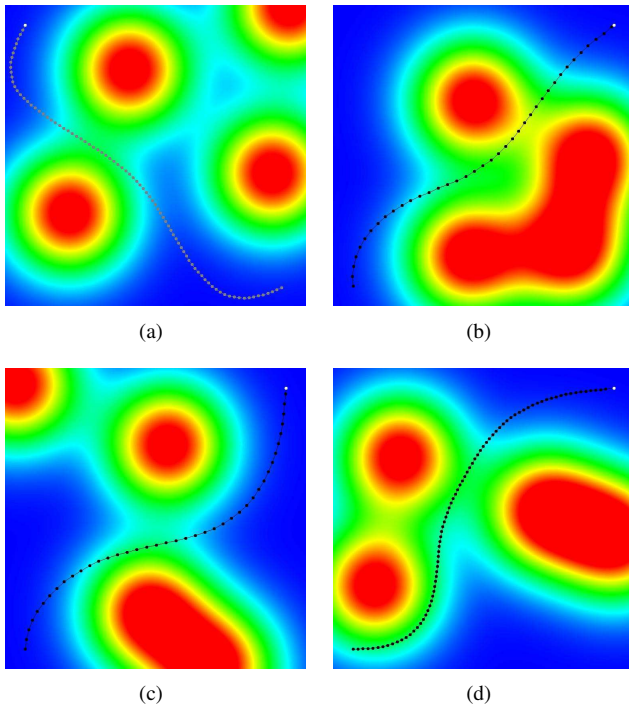


Fig. 4. Resulting trajectories in \mathbf{R}^2 space flying from bottom to top.

which are regarded as static obstacles that aircraft try to avoid, but can cross with high penalty. The associated time horizon is a few hours before entering such areas.

B. Results in 2D + time

Here, we deal with a situation of conflict resolution involving several aircraft.

The algorithm controls sequentially several aircraft trajectories by selecting aircraft according to a priority rule (in the following examples such rule has not been implemented and aircraft are selected according to their lexicographic order). The first aircraft supposes there is no other aircraft in the airspace. The second considers the first trajectory as a constraint and has to maintain separation with this latter aircraft. The third aircraft considers the two first trajectories as constraints. This process is repeated for the remaining aircraft.

The refractive index takes into account the protection zones of other aircraft which are represented by disks whose radius is the standard distance separation. The index inside a given disk has a very high value N and it is equal to 1 outside it. The index function of a space point \vec{Y} is then given by the following formula :

$$\begin{aligned} \|\vec{X}_i - \vec{Y}\| &:= \alpha_i. \\ \begin{cases} \text{if } \alpha_i \leq R & n(\vec{Y}) := N \quad N \gg 1. \\ \text{if } \alpha_i \geq R & n(\vec{Y}) := 1 \end{cases} \end{aligned} \quad (2)$$

where R is the standard separation distance. As it can be seen on equation 2, the index is symbolized by a barrier function which separates two constant index zones (in the disk and out of the disk). The light beams moving out of disks, will

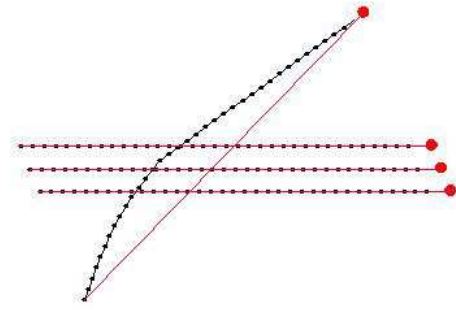


Fig. 5. In this situation, one aircraft coming from south-west going to north-east is crossing trajectories of three aircraft going to the east. The four aircraft fly at the same speed (450 kts) with the same altitude.

then have a constant velocity which is critical from the ATM point of view ². The ones entering separation disks will be destroyed by the branch and bound algorithm (no feasible solution).

In order to test our algorithm, three traffic situations have been created. For each figure, the protection zone will be shown by a red disk on each trajectory. The original trajectories are straight lines shown in red on the figures and the black ones are the trajectories produced by the algorithm.

The first situation shows an aircraft crossing a flow of three others aircraft going in the same direction (see figure 5). Initially, the crossing aircraft is in conflict with the three other ones. As we can see, the algorithm solves conflicts by moving the crossing aircraft in the west direction. For this example the crossing aircraft has been considered as the last aircraft (meaning that it considers the three other ones as constraints).

The second test presents an artificial situation involving 7 aircraft located on a circle of radius 100 Nm (see figure 6). Those aircraft move toward the circle center at the same speed inducing a 7 aircraft conflict. The aircraft being addressed sequentially, the produced conflict free trajectories are not symmetrical. Moreover, they are fully relevant from the air traffic management point of view (smooth).

Finally, a random situation involving 4 aircraft is shown on figure 7. Two aircraft trajectories have been changed in order to reach a conflict free situation.

In all these cases, the solution is found in less than 30 s of CPU and the trajectories generated by the algorithm avoid conflicts while remaining close enough to the original trajectories.

²In case of conflict resolution (a few minutes before the conflict), velocity change can not be used for resolution due to the lack of time for producing needed separation.

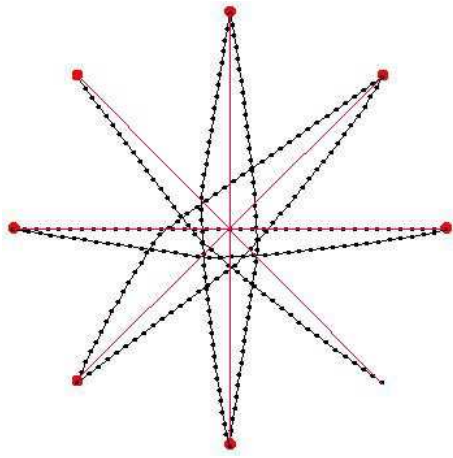


Fig. 6. In this situation, seven aircraft are converging towards the same point. As it can be seen on the figure, the first aircraft (going from west to east) maintains its original trajectory; the other ones undergo some changes.

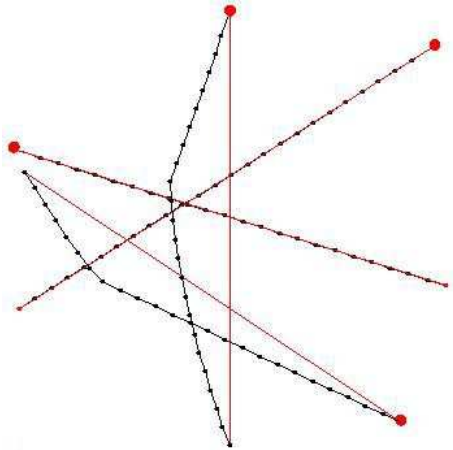


Fig. 7. In this figure, 4 aircraft are moving originally in straight lines producing 3 conflicts. The algorithm successfully addresses this situation by moving only two aircraft.

IV. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper a light-model algorithm has been developed to synthesize near optimal 4D trajectories which are geodesic approximations. A first version (2D space) enables congestion or weather avoidance in a pre-tactical framework (some hours before entering the area). A second version (2D + time) has been developed in order to address conflict resolution problem in a tactical framework (few minutes in advance). Both algorithms have been successfully applied to realistic 2D benchmarks for which result trajectories can be flown by the Flight Management System.

In the tactical framework, our light-model algorithm solves conflicts in a reasonable time (30s) which is a critical factor in such ATM application. Moreover, aircraft fly such trajectory at constant speed insuring that solutions will be built by heading changes (like it is done in the operational system).

In both cases, it is possible to ensure a minimum aircraft speed (higher than stall speed) by fixing the value of the maximum index (out of the separation disk for the tactical case).

B. Future Works

This algorithm is currently adapted in order to address a full day of augmented traffic over France which is stored in a database (the number of flight plans has been double compared to nowadays traffic). In a first step, the first 30 minutes trajectory segments are extracted from the database. Then, conflicts are detected and aggregated in clusters which are transferred to our solver. The conflict free trajectories are then updated in the database. The aircraft fly the first 20 minutes and the next 30 minutes trajectories segments are then extracted for the next step (based on a moving time window with no empty overlaps).

Furthermore, for tactical version of such algorithm, a non sequential implementation will be developed in order to improve equity between aircraft. The fact to remove sequential processing, induce a high combinatorial (NP_Hard) which imply the use of stochastic operator.

REFERENCES

- [1] N. Durand. *Optimisation de trajectoires pour la rsolution de conflits en route*. PhD thesis, ENSEEIHT, Institut National Polytechnique de Toulouse, France, 1996.
- [2] G. Roussos, G. Chaloulos, K. Kyriakopoulos and J. Lygeros. *Control of multiple non-holonomic air vehicles under wind uncertainty using model predictive control and decentralized navigation function*, IEEE Conference on Decision and Control, December, 2008.
- [3] S. Puechmorel and D. Delahaye. *Dynamical systems complexity with a view towards air traffic management applications*, IEEE Conference on Decision and Control, 2009.
- [4] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization. Algorithms and complexity*, Prentice-Hall, 1982.
- [5] D. Delahaye, S. Puechmorel. *Air traffic complexity: towards intrinsic matrices*, 3rd USA/Europe Air Traffic Management R& D Seminar Napoli, 2000.
- [6] Giancoli, Douglas C. *Physics for Scientists and Engineers with Modern Physics*, Prentice hall, 1989.
- [7] C. Huygens. *Traité de la lumière*, 1690.
- [8] E. Balas and P. Toth. *Branch and bound methods in the traveling salesman problem*, John Wiley & Sons, 361-401, 1985.