Mathematical Models for Aircraft Trajectory Design : A Survey
EIWAC 2013 Tokyo

D. Delahaye and S.Puechmorel and P.Tsiotras and E.Feron

Applied Mathematics Laboratory (MAIAA)
French Civil Aviation University
Toulouse, France
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, USA

February, 21 2013

Agenda

- Some Trajectory Models

Agenda

- Some Trajectory Models
- Strategic Trajectory Design

Agenda

- Some Trajectory Models
- Strategic Trajectory Design
- Pre-Tactical Trajectory Design

Agenda

- Some Trajectory Models
- Strategic Trajectory Design
- Pre-Tactical Trajectory Design
- Tactical Trajectory Design

Agenda

- Some Trajectory Models
- Strategic Trajectory Design
- Pre-Tactical Trajectory Design
- Tactical Trajectory Design
- Emergency Trajectory Design

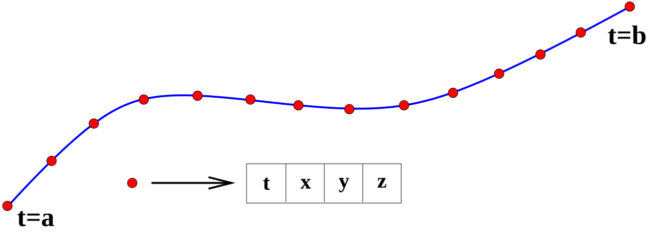
Agenda

- Some Trajectory Models
- Strategic Trajectory Design
- Pre-Tactical Trajectory Design
- Tactical Trajectory Design
- Emergency Trajectory Design

Trajectory Models

- Aircraft Trajectory Features
- Dimension Reduction Approaches
- Front Propagation Approaches
- Optimal Control Approaches

Classical representation

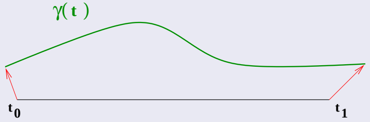


Trajectory data is expressed as an ordered list of plots (no aircraft dynamics in such representation)

Trajectories as functional data

Trajectories are infinite dimension mathematical objects

Trajectories as mappings

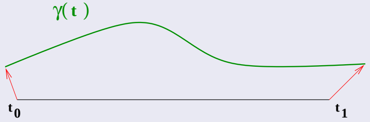


- Intuitive approach : a trajectory maps a bounded time interval $[t_0, t_1]$ to the state space ($\mathbb{R}^3$ or $\mathbb{R}^6$).

Trajectories as functional data

Trajectories are infinite dimension mathematical objects

Trajectories as mappings

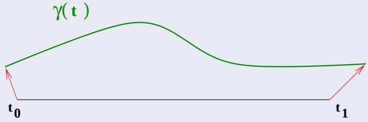


- Intuitive approach : a trajectory maps a bounded time interval $[t_0, t_1]$ to the state space ($\mathbb{R}^3$ or $\mathbb{R}^6$).
- Smoothness assumptions are made for trajectories (C^2).

Trajectories as functional data

Trajectories are infinite dimension mathematical objects

Trajectories as mappings



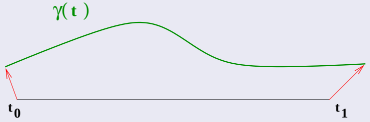
- Intuitive approach : a trajectory maps a bounded time interval $[t_0, t_1]$ to the state space ($\mathbb{R}^3$ or $\mathbb{R}^6$).
- Smoothness assumptions are made for trajectories (C^2).

Trajectories as shapes

Trajectories as functional data

Trajectories are infinite dimension mathematical objects

Trajectories as mappings



- Intuitive approach : a trajectory maps a bounded time interval $[t_0, t_1]$ to the state space ($\mathbb{R}^3$ or $\mathbb{R}^6$).
- Smoothness assumptions are made for trajectories (C^2).

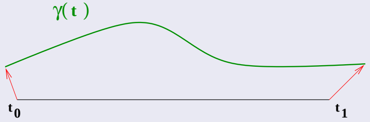
Trajectories as shapes

- The paths flown by aircraft are considered as curves in $\mathbb{R}^3$.

Trajectories as functional data

Trajectories are infinite dimension mathematical objects

Trajectories as mappings



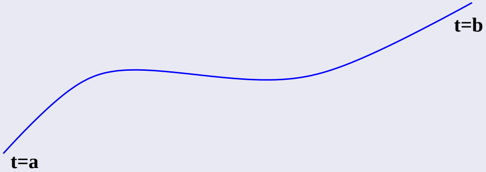
- Intuitive approach : a trajectory maps a bounded time interval $[t_0, t_1]$ to the state space ($\mathbb{R}^3$ or $\mathbb{R}^6$).
- Smoothness assumptions are made for trajectories (C^2).

Trajectories as shapes

- The paths flown by aircraft are considered as curves in $\mathbb{R}^3$.
- Such time independant trajectories are called *shapes*.

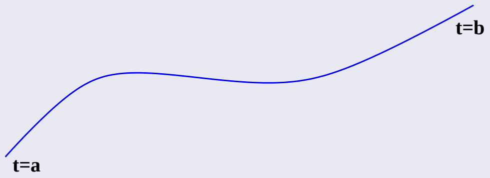
Aircraft Trajectories Features

Notations



Aircraft Trajectories Features

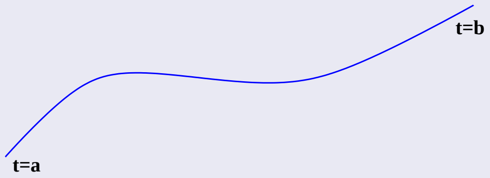
Notations



- Trajectory $\vec{\gamma} : \vec{\gamma}[a, b] \rightarrow E$ ($[a, b]$ time interval, $E : \mathbb{R}^3$ or $\mathbb{R}^6$)

Aircraft Trajectories Features

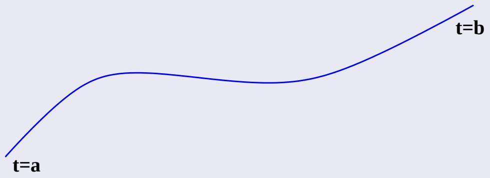
Notations



- Trajectory $\vec{\gamma} : \vec{\gamma}[a, b] \rightarrow E$ ($[a, b]$ time interval, $E : \mathbb{R}^3$ or $\mathbb{R}^6$)
- Trajectory length $l(\vec{\gamma}) = \int_a^b \|\vec{\gamma}'(t)\| dt$

Aircraft Trajectories Features

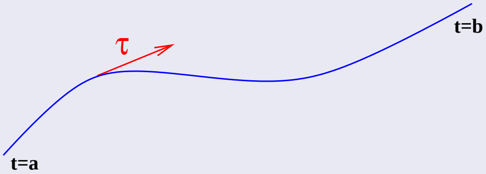
Notations



- Trajectory $\vec{\gamma} : \vec{\gamma}[a, b] \rightarrow E$ ($[a, b]$ time interval, $E : \mathbb{R}^3$ or $\mathbb{R}^6$)
- Trajectory length $l(\vec{\gamma}) = \int_a^b \|\vec{\gamma}'(t)\| dt$
- Parametrization by arclength : $s(a, b) \rightarrow (0, l(\vec{\gamma}))$
  $s(t) = \int_a^t \|\vec{\gamma}'(x)\| dx$ ($s'(t) = \|\vec{\gamma}'(t)\| > 0 \ \forall t \in (a, b)$)

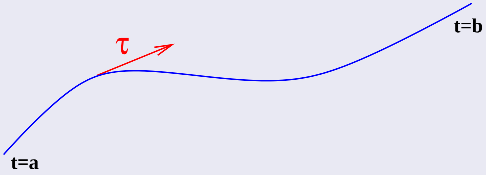
Aircraft Trajectories Feature

Unit tangent vector



Aircraft Trajectories Feature

Unit tangent vector



- $\vec{\tau}(s) = \vec{\gamma}'(s)$

Aircraft Trajectories Feature

Curvature

- $K(s) = \|\vec{\gamma}''(s)\| = \frac{\|\vec{\gamma}'(t) \wedge \vec{\gamma}''(t)\|}{\|\vec{\gamma}'(t)\|^3}$

Aircraft Trajectories Feature

Curvature

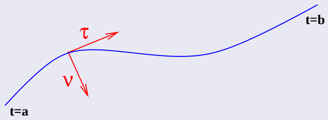
- $K(s) = \|\vec{\gamma}''(s)\| = \frac{\|\vec{\gamma}'(t) \wedge \vec{\gamma}''(t)\|}{\|\vec{\gamma}'(t)\|^3}$
- Aircraft trajectories have piecewise constant curvature.

Aircraft Trajectories Feature

Curvature

- $K(s) = \|\vec{\gamma}''(s)\| = \frac{\|\vec{\gamma}'(t) \wedge \vec{\gamma}''(t)\|}{\|\vec{\gamma}'(t)\|^3}$
- Aircraft trajectories have piecewise constant curvature.

Unit normal vector

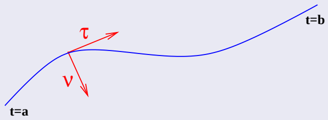


Aircraft Trajectories Feature

Curvature

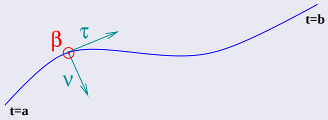
- $K(s) = \|\vec{\gamma}''(s)\| = \frac{\|\vec{\gamma}'(t) \wedge \vec{\gamma}''(t)\|}{\|\vec{\gamma}'(t)\|^3}$
- Aircraft trajectories have piecewise constant curvature.

Unit normal vector



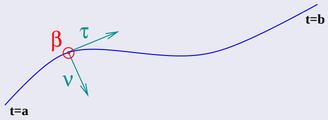
- $\vec{\nu}(s) = \frac{\vec{\gamma}''(s)}{K(s)}$

Torsion



- $\vec{\beta}(s) = \vec{\tau}(s) \wedge \vec{\nu}(s) \quad \vec{\beta}'(s) = T(s).\vec{\nu}(s)$

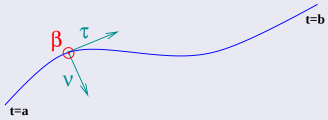
Torsion



- $\vec{\beta}(s) = \vec{\tau}(s) \wedge \vec{\nu}(s) \quad \vec{\beta}'(s) = T(s).\vec{\nu}(s)$
- The real number $T(s)$ is called the torsion of the curve at s and represents an obstruction for the curve to be planar.

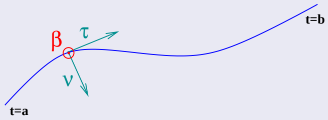
Aircraft Trajectories Feature

Torsion



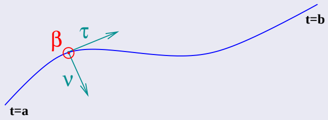
- $\vec{\beta}(s) = \vec{\tau}(s) \wedge \vec{\nu}(s) \quad \vec{\beta}'(s) = T(s).\vec{\nu}(s)$
- The real number $T(s)$ is called the torsion of the curve at s and represents an obstruction for the curve to be planar.
- $T(t) = -\frac{det(\vec{\gamma}'(t),\vec{\gamma}''(t),\vec{\gamma}'''(t))}{\|\vec{\gamma}'(t)\wedge\vec{\gamma}''(t)\|^2}$

Torsion



- $\vec{\beta}(s) = \vec{\tau}(s) \wedge \vec{\nu}(s) \quad \vec{\beta}'(s) = T(s).\vec{\nu}(s)$
- The real number $T(s)$ is called the torsion of the curve at s and represents an obstruction for the curve to be planar.
- $T(t) = -\frac{det(\vec{\gamma}'(t),\vec{\gamma}''(t),\vec{\gamma}'''(t))}{\|\vec{\gamma}'(t)\wedge\vec{\gamma}''(t)\|^2}$
- Aircraft have piecewise constant torsion mainly in terminal area.

Torsion



- $\vec{\beta}(s) = \vec{\tau}(s) \wedge \vec{\nu}(s) \quad \vec{\beta}'(s) = T(s).\vec{\nu}(s)$
- The real number $T(s)$ is called the torsion of the curve at s and represents an obstruction for the curve to be planar.
- $T(t) = -\frac{det(\vec{\gamma}'(t),\vec{\gamma}''(t),\vec{\gamma}'''(t))}{\|\vec{\gamma}'(t)\wedge\vec{\gamma}''(t)\|^2}$
- Aircraft have piecewise constant torsion mainly in terminal area.
- All the previous derivations rely on the fact that the first three derivatives of the trajectory are available.

Trajectory Models

- Aircraft Trajectory Features
- Dimension Reduction Approaches
- Front Propagation Approaches
- Optimal Control Approaches

Explicit vs Implicit

Explicit

$$y = f(x)$$

Example 2D line $y = a.x + b$
A curve may not have an explicit representation

Implicit

$$f(x, y) = 0$$

Example 2D circle $x^2 + y^2 - r^2 = 0$

Parametric Form

Expresses the value of each spatial variables for points in terms of an independent parameter u.

$$\vec{p}(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix}$$

Parametric Polynomial Curve

Consider a curve

$$\vec{p}(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix}$$

A polynomial parametric curve of degree n is of the form :

$$\vec{p}(u) = \sum_{k=0}^{n} \vec{c}_k . u^k$$

where each $\vec{c}_k$ has independent x, y, z components : $\vec{c}_k = [c_{kx}, c_{ky}, c_{kz}]^T$

Advantages of the Parametric Polynomial Curve

- Just needs to save a few control points
- Local control of shape
- Smoothness and continuity
- Ability to evaluate derivatives
- Stability
- Ease of rendering

Lagrangian Interpolation

Given $n+1$ real numbers y_i,$0 \leq i \leq n$, and $n+1$ distinct real numbers $x_0 < x_1 < ... < x_n$, *Lagrange polynomial of degree* n associated with $\{x_i\}$ and $\{y_i\}$ is a polynomial of degree n solving the interpolation problem :

$$p_n(x_i) = y_i, \quad 0 \leq i \leq n$$

Solution :

$$L_n(x) = \sum_{i=0}^{n} f(x_i) l_i(x)$$

where

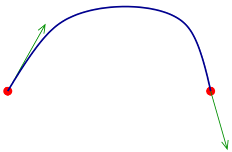
$$l_i(x) = \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

Hermite Interpolation

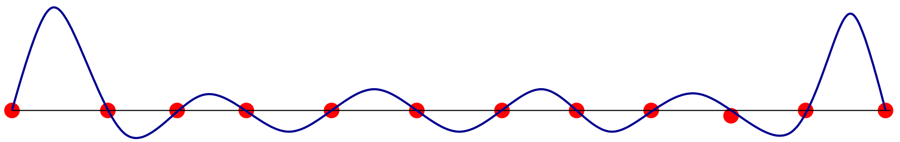
Hermite interpolation generalizes Lagrange interpolation by fitting a polynomial to a function f that not only interpolates f at each knot but also interpolates a given number of consecutive derivatives of f at each knot.

$$\left[\frac{\partial^j H(x)}{\partial x^j}\right]_{x=x_i} = \left[\frac{\partial^j f(x)}{\partial x^j}\right]_{x=x_i}$$

for all $j = 0, 1, ..., m$ and $i = 1, 2, ..., k$



Runge phenomenon

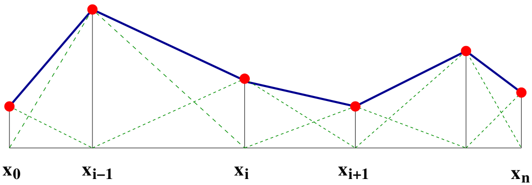


Interpolation with high degree polynomial is risky...

Solution : Piecewise interpolation

Piecewise Linear Interpolation

The simplest one



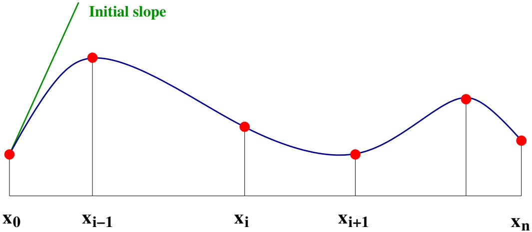
Given $n+1$ real numbers y_i,$0 \leq i \leq n$, and $n+1$ distinct real numbers $x_0 < x_1 < ... < x_n$, we consider the n linear curves $l_i(x) = a_i x + b_i$ on the intervals $[x_i, x_{i+1}]$ for $i = 0, ...n-1$.

- each $l_i(x)$ has to connect two points $\{(x_i, y_i),(x_{i+1}, y_{i+1})\}$

$$y_i = a_i x_i + b_i x_i \quad y_{i+1} = a_i x_{i+1} + b_i x_{i+1}$$

The resulting curves is not derivative.

Piecewise Quadratic Interpolation



Piecewise Quadratic Interpolation

We consider the n quadratic curves $q_i(x) = a_i x^2 + b_i x + c_i$ on the intervals $[x_i, x_{i+1}]$ for $i = 0, ...n - 1$.

- Each $q_i(x)$ has to connect two points $((x_i, y_i),(x_{i+1}, y_{i+1})$

$$y_i = a_i x_i^2 + b_i x_i + c_i$$

$$y_{i+1} = a_i x_{i+1}^2 + b_i x_{i+1} + c_i$$

Piecewise Quadratic Interpolation

We consider the n quadratic curves $q_i(x) = a_i x^2 + b_i x + c_i$ on the intervals $[x_i, x_{i+1}]$ for $i = 0, ...n - 1$.

- Each $q_i(x)$ has to connect two points $((x_i, y_i),(x_{i+1}, y_{i+1})$

$$y_i = a_i x_i^2 + b_i x_i + c_i$$

$$y_{i+1} = a_i x_{i+1}^2 + b_i x_{i+1} + c_i$$

- On each point the derivative of the previous quadratic has to be equal to the derivative of the next one.

$$2a_i + b_i = 2a_{i-1} + b_{i-1}$$

Piecewise Quadratic Interpolation

We consider the n quadratic curves $q_i(x) = a_i x^2 + b_i x + c_i$ on the intervals $[x_i, x_{i+1}]$ for $i = 0, ...n - 1$.

- Each $q_i(x)$ has to connect two points $((x_i, y_i),(x_{i+1}, y_{i+1})$

$$y_i = a_i x_i^2 + b_i x_i + c_i$$

$$y_{i+1} = a_i x_{i+1}^2 + b_i x_{i+1} + c_i$$

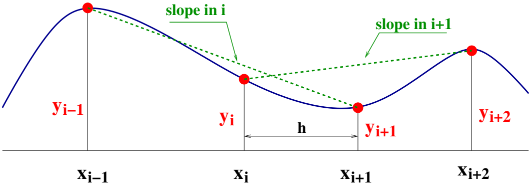
- On each point the derivative of the previous quadratic has to be equal to the derivative of the next one.

$$2a_i + b_i = 2a_{i-1} + b_{i-1}$$

- For the first segment the term $2a_{i-1} + b_{i-1}$ is arbitrarily chosen. (this affects the rest of the curve).

Piecewise Cubic Interpolation

Also called Hermite Cubic Interpolation



$$C_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

$$C_i(x_i) = y_i \qquad C_i(x_{i+1}) = y_{i+1}$$

$$C_i'(x_i) = y_i' = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \qquad C_i'(x_{i+1}) = y_{i+1}' = \frac{y_{i+2} - y_i}{x_{i+2} - x_i}$$

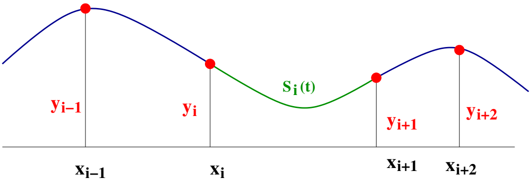
- Moving a point do not affect all the curve
- The curve is C^1 but not C^2.

Curvature radius

$$R = \frac{1 + \left(\frac{df(x)}{dx}\right)^{\frac{3}{2}}}{\left| \left(\frac{d^2 f(x)}{dx^2}\right) \right|}$$

In order to have a continuous curverture one must force curves to be C^2.

- Piecewise cubic interpolation
- Developped by General Motor in the 1950s.



$$S_i(x_i) = y_i \qquad S_i(x_{i+1}) = y_{i+1}$$
$$S_i'(x_i) = S_{i-1}'(x_{i+1}) \qquad S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$$
$$S_i''(x_i) = S_{i-1}''(x_{i+1}) \qquad S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$$

Cubic Spline Interpolation

$S_i(x)$ for $x \in [x_i, x_{i+1}]$

$$\begin{aligned} S_i(x) &= \tfrac{\sigma_i}{6}.\tfrac{(x_{i+1}-x)^3}{x_{i+1}-x_i} + \tfrac{\sigma_{i+1}}{6}.\tfrac{(x-x_i)^3}{x_{i+1}-x_i} \\ &+ y_i.\tfrac{x_{i+1}-x}{x_{i+1}-x_i} - \tfrac{\sigma_i}{6}.(x_{i+1}-x_i)(x_{i+1}-x) \\ &+ y_{i+1}.\tfrac{x-x_i}{x_{i+1}-x_i} - \tfrac{\sigma_{i+1}}{6}.(x_{i+1}-x_i)(x-x_i) \end{aligned}$$

where

$$\sigma_i = \frac{d^2 S_i(x)}{dx^2}$$

Such spline is also called natural spline because it represents the curve of a metal spline constrained to interpolate some given points.

Bézier Approximation Curve

- Bézier curves were first developped by automobile designers to describe the shape of exterior car panels in the 1960s and 70s.

Bézier Approximation Curve

- Bézier curves were first developed by automobile designers to describe the shape of exterior car panels in the 1960s and 70s.
- Given points $\vec{P}_0$ and $\vec{P}_1$, a linear Bézier curve is simply a straight line between those two points. The curve is given by

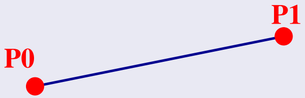
$$B(t) = \vec{P}_0 + t(\vec{P}_1 - \vec{P}_0) = (1-t)\vec{P}_0 + t\vec{P}_1 \ , \ t \in [0,1]$$

Bézier Approximation Curve

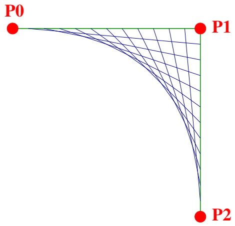
- Bézier curves were first developped by automobile designers to describe the shape of exterior car panels in the 1960s and 70s.
- Given points $\vec{P}_0$ and $\vec{P}_1$, a linear Bézier curve is simply a straight line between those two points. The curve is given by

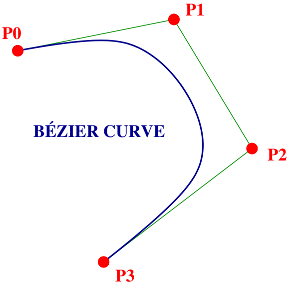
$$B(t) = \vec{P}_0 + t(\vec{P}_1 - \vec{P}_0) = (1-t)\vec{P}_0 + t\vec{P}_1 \ , \ t \in [0,1]$$

Bézier Curve with 2 points



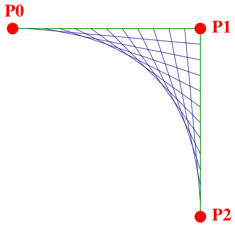
Cubic Bézier curves

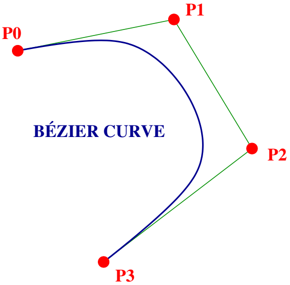




- Four points $\vec{P}_0$, $\vec{P}_1$, $\vec{P}_2$ and $\vec{P}_3$ in the plane or in higher-dimensional space define a cubic Bézier curve.

Cubic Bézier curves





- Four points $\vec{P}_0$, $\vec{P}_1$, $\vec{P}_2$ and $\vec{P}_3$ in the plane or in higher-dimensional space define a cubic Bézier curve.
- The curve starts at $\vec{P}_0$ going towards $\vec{P}_1$ and arrives at $\vec{P}_3$ coming from the direction of $\vec{P}_2$. Usually, it will not pass through $\vec{P}_1$ or $\vec{P}_2$; these points are only there to provide directional information.

- The polygon formed by connecting the Bézier points with lines, starting with $\vec{P}_0$ and finishing with $\vec{P}_n$, is called the Bézier polygon (or control polygon).

Cubic Bézier curves

- The polygon formed by connecting the Bézier points with lines, starting with $\vec{P}_0$ and finishing with $\vec{P}_n$, is called the Bézier polygon (or control polygon).
- The convex hull of the Bézier polygon contains the Bézier curve.

Cubic Bézier curves

- The polygon formed by connecting the Bézier points with lines, starting with $\vec{P}_0$ and finishing with $\vec{P}_n$, is called the Bézier polygon (or control polygon).
- The convex hull of the Bézier polygon contains the Bézier curve.
- The start (end) of the curve is tangent to the first (last) section of the Bézier polygon.

Cubic Bézier curves

The explicit form of the curve is :

$$B(t) = (1-t)^3\vec{P}_0 + 3(1-t)^2 t\vec{P}_1 + 3(1-t)t^2\vec{P}_2 + t^3\vec{P}_3 \ , \ t \in [0,1].$$

$$B(t) = \sum_{i=0}^{n} b_{i,n}(t)\vec{P}_i, \quad t \in [0,1]$$

where the polynomials

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots n$$

are known as Bernstein basis polynomials of degree n.

A Bézier curve defined with $n+1$ control points is of degree n.

So if there are many points one has to manipulate polynoms with high degree ⇒ Basis-Splines

B-Splines

Powerful tool for generating curves with many control points, B stands for basis.

- A single B-spline can specify a long complicated curve

B-Splines

Powerful tool for generating curves with many control points, B stands for basis.

- A single B-spline can specify a long complicated curve
- B-splines can be designed with sharp bends and even "corners"

B-Splines

Powerful tool for generating curves with many control points, B stands for basis.

- A single B-spline can specify a long complicated curve
- B-splines can be designed with sharp bends and even "corners"
- B-Spline interpolation is preferred over polynomial interpolation because the interpolation error can be made small even when using low degree polynomials for the spline.

B-Splines

Powerful tool for generating curves with many control points, B stands for basis.

- A single B-spline can specify a long complicated curve
- B-splines can be designed with sharp bends and even "corners"
- B-Spline interpolation is preferred over polynomial interpolation because the interpolation error can be made small even when using low degree polynomials for the spline.
- Spline interpolation avoids the problem of Runge's phenomenon which occurs when interpolating between equidistant points with high degree polynomials.

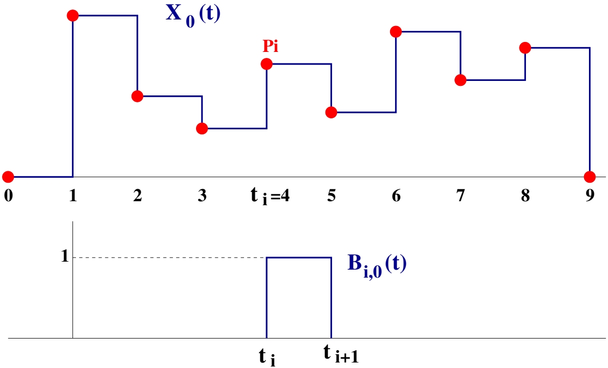
We consider a node vector $\vec{T} = \{t_0, t_1, ..., t_n\}$ with $t_0 \leq t_1 \leq, ..., \leq t_n$ and n points $\vec{P}_i$.
One want to build a curve $\vec{X}_0(t)$ such that

$$\vec{X}_0(t_i) = \vec{P}_i$$

$\Rightarrow \vec{X}_0(t) = \vec{P}_i \ \forall t \in [t_i, t_{i+1}]$

$$\vec{X}_0(t) = \sum_i B_{i,0}(t).\vec{P}_i$$

Uniform B-Splines of Degree Zero

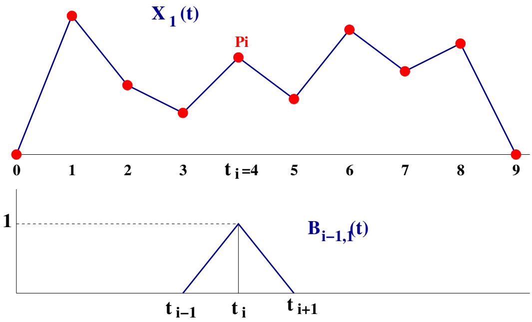


Uniform B-Splines of Degree One

We are searching for a piecewise linear approximation :

$$\vec{X}_1(t) = \left(1 - \frac{t - t_i}{t_{i+1} - t_i}\right)\vec{P}_{i-1} + \left(1 - \frac{t - t_i}{t_{i+1} - t_i}\right)\vec{P}_i \quad \forall t \in [t_i, t_{i+1}]$$

$$\vec{X}_1(t) = \sum_i B_{i,1}(t).\vec{P}_i$$



Uniform B-Splines of Degree Three

- Developped at Boeing in the 70s.

Uniform B-Splines of Degree Three

- Developped at Boeing in the 70s.
- One of the simplest and most useful cases of B-splines

Uniform B-Splines of Degree Three

- Developped at Boeing in the 70s.
- One of the simplest and most useful cases of B-splines
- Degree 3 B-Spline with $n+1$ control points :

$$\vec{X}_3(t) = \sum_{i=0}^{n} B_{i,3}(t).\vec{P}_i \;\; 3 \leq t \leq n+1$$

Uniform B-Splines of Degree Three

- Developped at Boeing in the 70s.
- One of the simplest and most useful cases of B-splines
- Degree 3 B-Spline with $n+1$ control points :

$$\vec{X}_3(t) = \sum_{i=0}^{n} B_{i,3}(t).\vec{P}_i \;\; 3 \leq t \leq n+1$$

- For degree 3,
  $B_{i,3}(t) = 0$ if $t \leq t_i$ or $t \geq t_{i+4}$ So

$$\vec{X}_3(t) = \sum_{i=j-3}^{j} B_{i,3}(t).\vec{P}_i \;\; t \in [j, j+1], \; 3 \leq j \leq n$$

When a single control point P_i is moved, only the portion of the curve $\vec{X}_3(t)$ with $t_i < t < t_{i+4}$ is changed $\Rightarrow$ local control.

The basis functions have the following properties :

- They are translates of each other i.e $B_{i,3}(t) = B_{0,3}(t-i)$

Uniform B-Splines of Degree Three

The basis functions have the following properties :

- They are translates of each other i.e $B_{i,3}(t) = B_{0,3}(t-i)$
- They are piecewise degree three polynomial

Uniform B-Splines of Degree Three

The basis functions have the following properties :

- They are translates of each other i.e $B_{i,3}(t) = B_{0,3}(t-i)$
- They are piecewise degree three polynomial
- Partition of unity $\sum_i B_i(t) = 1$ for $3 \leq t \leq n+1$

Uniform B-Splines of Degree Three

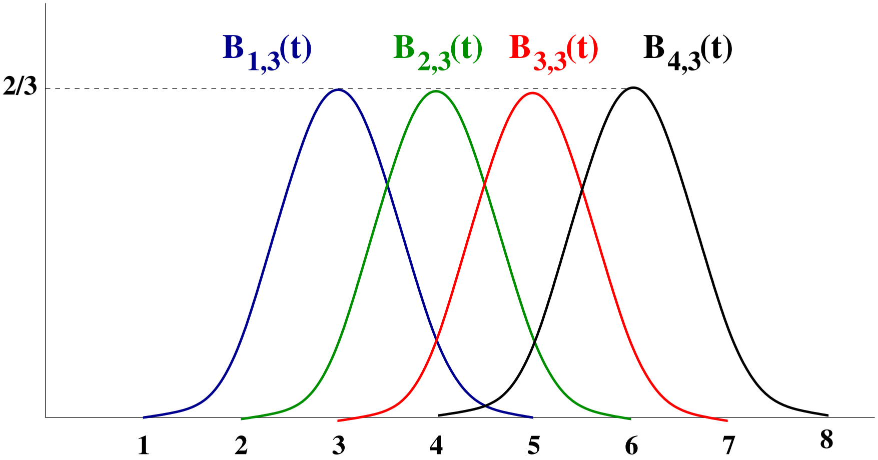
The basis functions have the following properties :

- They are translates of each other i.e $B_{i,3}(t) = B_{0,3}(t - i)$
- They are piecewise degree three polynomial
- Partition of unity $\sum_i B_i(t) = 1$ for $3 \leq t \leq n + 1$
- The functions $\vec{X}_i(t)$ are of degree 3 for any set of control points

Uniform B-Splines of Degree Three

$$B_{i-2,3}(t) = \frac{1}{h} \begin{cases} (t - t_{i-2})^3 \quad \text{if } t \in [t_{i-2}, t_{i-1}] \\ h^3 + 3h^2(t - t_{i-1}) + 3h(t - t_{i-1})^2 - 3(t - t_{i-1})^3 \\ \text{if } t \in [t_{i-1}, t_i] \\ h^3 + 3h^2(t_{i+1} - t) + 3h(t_{i+1} - t)^2 - 3(t_{i+1} - t)^3 \\ \text{if } t \in [t_i, t_{i+1}] \\ (t_{i+2} - t)^3 \quad \text{if } t \in [t_{i+1}, t_{i+2}] \\ 0 \text{ otherwise} \end{cases}$$

Uniform B-Splines of Degree Three

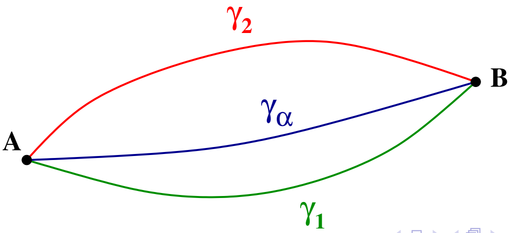


Homotopy Trajectory Design

If we consider two (or more) references trajectories $(\gamma_1(t), \gamma_2(t))$joining the same origine destination pair (past flown trajectories may be considered), one can create a new trajectory $\gamma(\alpha, t)$ by using an homotopy :

$$\gamma(\alpha, t) = \begin{cases} \gamma(0, t) = \gamma_1(t) \\ \gamma(1, t) = \gamma_2(t) \end{cases}$$

$$\gamma(\alpha, t) = (1 - \alpha)\gamma_1(t) + \alpha\gamma_2(t)$$



Functionnal Principal Component Analysis

Functionnal Principal Component Analysis

- Used for Stochastic Signal Compression (movies, image, voice)

Functionnal Principal Component Analysis

- Used for Stochastic Signal Compression (movies, image, voice)
- The goal of principal component analysis is to compute the most meaninfugful basis to re-express a noisy data set (maximize SNR,minimize redundancy).

Functionnal Principal Component Analysis

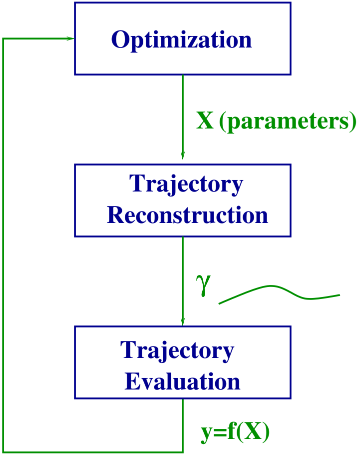
- Used for Stochastic Signal Compression (movies, image, voice)
- The goal of principal component analysis is to compute the most meaninfugful basis to re-express a noisy data set (maximize SNR,minimize redundancy).
- If speed is suitable one must work in Sobolev space

Functionnal Principal Component Analysis

- Used for Stochastic Signal Compression (movies, image, voice)
- The goal of principal component analysis is to compute the most meaninfugful basis to re-express a noisy data set (maximize SNR,minimize redundancy).
- If speed is suitable one must work in Sobolev space
- Extraction of the Probability Density Function of PCA coefficients in order to be able to randomly generate "flyable trajectories".

Optimization Approach

All the previous representations may be used in the following process



Trajectory Models

- Aircraft Trajectory Features
- Dimension Reduction Approaches
- Front Propagation Approaches
- Optimal Control Approaches

Propagating front methods : General principle

Methods introduced by J.A. Sethian.

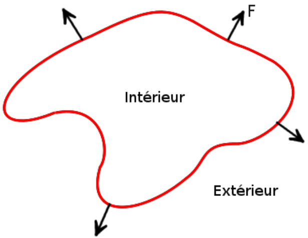


FIGURE: Curve propagating with speed F in normal direction.

Goal :

Track the motion of a front as it evolves.

How ?

We caracterize the position of the front by the computation of the arrival time $u(x, y)$ at each point (x, y).

$\Rightarrow$ Map of isocost.

Propagating front methods

Fast Marching :

→ **Isotropic problem**

The speed of propagation F is the same in any directions, it only depends on the position.

Ordered Upwind :

→ **Anisotropic problem**

The speed of propagation depends on position and direction of the propagation.

Fast Marching Method

Statement of the problem in the case of optimal path planning :
(J.A. Sethian, 1998)

Let $u(x)$ be the time where the front crosses the point x.

Computation of $u \rightarrow$ Solving the Eikonal equation :

$$\begin{cases} |\nabla u(x)| F(x) = 1 \text{ in } \Omega, & F(x) > 0 \\ \Gamma(u) = \{x | u(x) = u_0\}, \end{cases}$$

where x is the position and F is the propagation speed.

Fast Marching Method

Statement of the problem in the case of optimal path planning :
(J.A. Sethian, 1998)

Let $u(x)$ be the time where the front crosses the point x.

Computation of $u \rightarrow$ Solving the Eikonal equation :

$$\begin{cases} |\nabla u(x)|F(x) = 1 \text{ in } \Omega, & F(x) > 0 \\ \Gamma(u) = \{x | u(x) = u_0\}, & \end{cases}$$

where x is the position and F is the propagation speed.

To plan the optimal path $\gamma(t)$ (back traking) :

$$\frac{d\gamma(t)}{dt} = -\frac{\nabla u}{||\nabla u||}$$

Numerical solving : Godonov Scheme

The principal idea is to construct the solution using only upwind values. For this, we divide all the mesh points in **three sets** :

- **Accepted** : Set of points where the solution is known ;
- **Considered** : Set of points which are adjacent to at least one *Accepted* point ;
- **Far** : Set of points where we do not have yet any information about the solution.

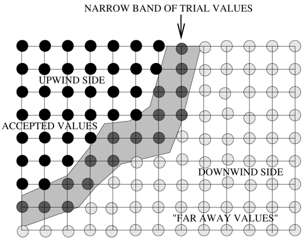


FIGURE: Construction of the algorithm

Fast Marching Algorithm

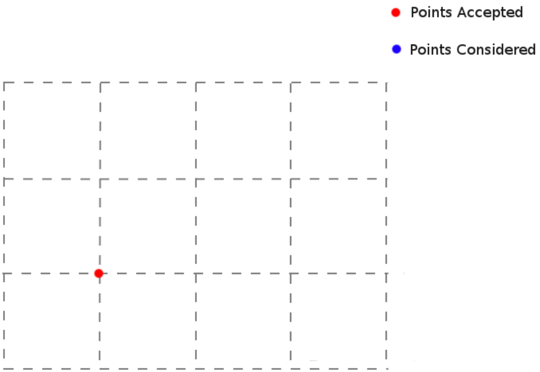


FIGURE: Step 1 : **Initialization**

Fast Marching Algorithm

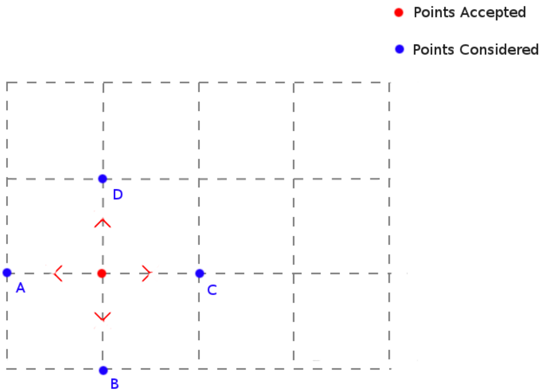


FIGURE: Step 2 : **Transfering** → *Considered*

Fast Marching Algorithm

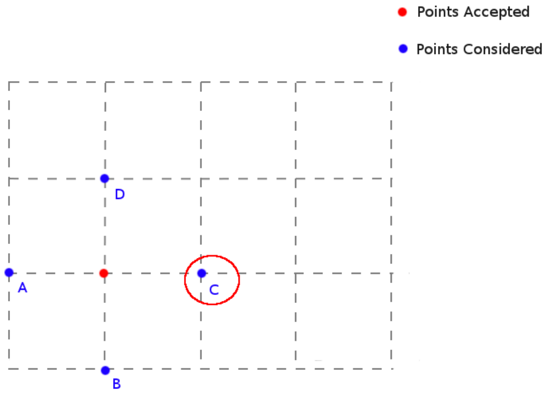


FIGURE: Step 3 : **Looking for** the smallest value $u(x_i)$

Fast Marching Algorithm

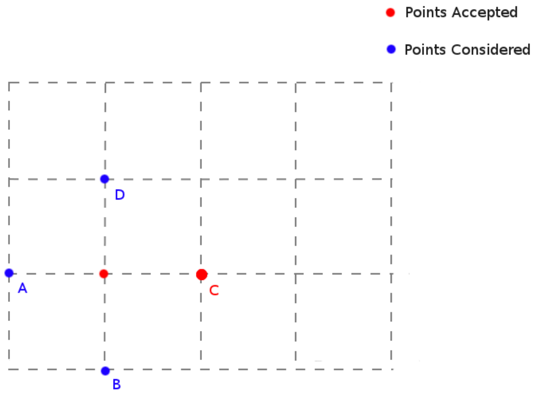


FIGURE: Step 4 : **Transfering** → *Accepted*

Fast Marching Algorithm

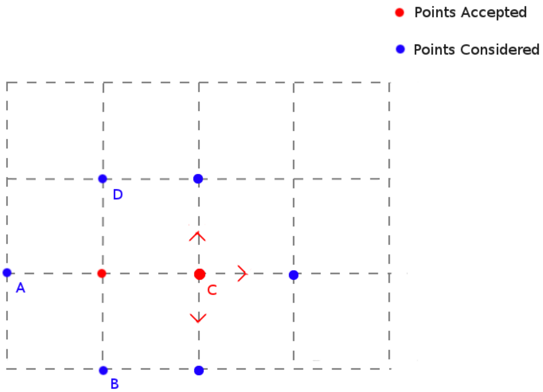


FIGURE: Step 5 : **Transfering** → *Considered*

Fast Marching Algorithm

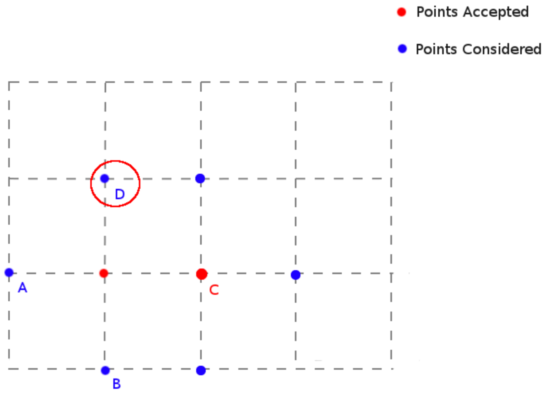


FIGURE: Step 6 : **Looking for** the smallest value $u(x_i)$

Fast Marching Algorithm

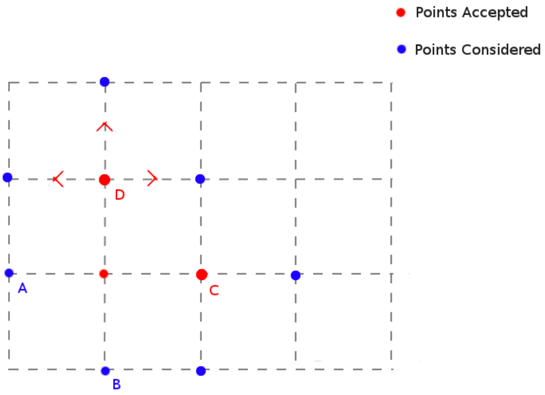


FIGURE: Step 7 : **Transfering** $\rightarrow$ *Considered*

Fast Marching Algorithm

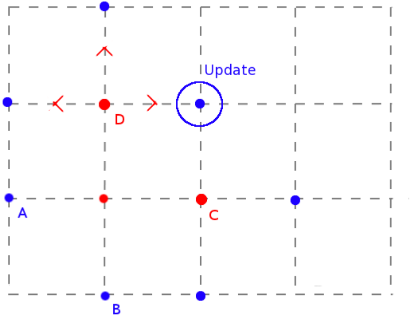


FIGURE: Step 8 : **Recomputing the value** $u(x_i)$

Fast Marching Algorithm

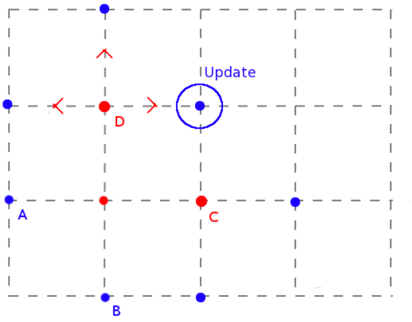


FIGURE: Step 8 : **Recomputing the value** $u(x_i)$

Trajectory Models

- Aircraft Trajectory Features
- Dimension Reduction Approaches
- Front Propagation Approaches
- Optimal Control Approaches

Optimal Control for Trajectory Generation

- Mainly used for time-parameterized of shapes.

Optimal Control for Trajectory Generation

- Mainly used for time-parameterized of shapes.
- Generating *time-parameterized* paths necessitates the incorporation of the aircraft dynamics.

Optimal Control for Trajectory Generation

- Mainly used for time-parameterized of shapes.
- Generating *time-parameterized* paths necessitates the incorporation of the aircraft dynamics.
- The objective of optimal control theory is to determine the control input(s) that will cause a process to satisfy the physical constraints, while, at the same time, minimize (or maximize) some performance criterion.

Optimal Control for Trajectory Generation

- Mainly used for time-parameterized of shapes.
- Generating *time-parameterized* paths necessitates the incorporation of the aircraft dynamics.
- The objective of optimal control theory is to determine the control input(s) that will cause a process to satisfy the physical constraints, while, at the same time, minimize (or maximize) some performance criterion.
- Feasibility of the trajectories is automatically ensured using this approach.

Optimal Control for Trajectory Generation

Given initial conditions x_0, final conditions $x_f \in \mathcal{X}$, and an initial time $t_0 \geq 0$, determine the final time $t_f > t_0$, the control input $u(t) \in \mathcal{U}$ and the corresponding state history $x(t)$ for $t \in [t_0, t_f]$ which minimize the cost function

$$J(x, u) = \int_{t_0}^{t_f} L(x(t), u(t)) \, \mathrm{d}t,$$

where $x(t)$ and $u(t)$ satisfy, for all $t \in [t_0, t_f]$ the differential and algebraic constraints.

$$\begin{cases} \dot{x}(t) - f(x(t), u(t)) = 0, \\ C(x(t), u(t)) \leq 0. \end{cases}$$

Optimal Control for Trajectory Generation

- Optimal control has its roots in the theory of calculus of variations, which originated in the 17th century by Fermat, Newton, Liebniz,etc...

Optimal Control for Trajectory Generation

- Optimal control has its roots in the theory of calculus of variations, which originated in the 17th century by Fermat, Newton, Liebniz,etc...
- It was not until the middle of the 20th century when the Soviet mathematician Pontryagin developed a complete theory that could handle such problem.

Optimal Control for Trajectory Generation

- Pontryagin's celebrated Maximum Principle states that the optimal control for the solution of the problem is given as the pointwise minimum of the so-called Hamiltonian function, that is :

$$u_{\mathrm{opt}} = \mathrm{argmin}_{u \in U} H(t, x, \lambda, u)$$

where $H(t, x, \lambda, u) = L(x, u) + \lambda^T f(x, u)$ is the Hamiltonian, and λ are the co-states, computed from

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial x}(x(t), \lambda(t), u(t)). \quad (1)$$

subject to certain boundary (transversality) conditions on $\lambda(t_f)$.

Numerical solution

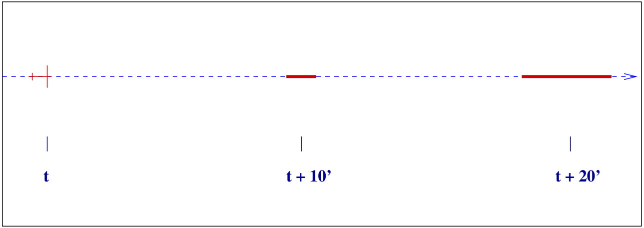
Agenda

- Some Trajectory Models
- Strategic Trajectory Design
- Pre-Tactical Trajectory Design
- Tactical Trajectory Design
- Emergency Trajectory Design

Continental Strategic Planning

- Before take-off
- Trajectory design for large segment (full trajectory)
- Action on time and space
- Large scale (30000-50000 aircraft)
- Continental or Oceanic
- Macroscopic congestion criterium
- One must take into account uncertainties

Uncertainties



Trajectory prediction limitation Factors

1. Wind ($\vec{V} = \vec{T} + \vec{W}$)
2. Temperature, pressure (engine trust, drag $d = \frac{1}{2}.c_x.\rho.S.v^2$)
3. Weight

On-board trajectory prediction

FMS in open loop : $+-15$Nm after one hour flight.

How much can we reduce congestion in the French Airspace ?

Optimization Approach

EUROCONTROL

How much can we reduce congestion in the French Airspace ?

- Approach based on optimization

What are our state space variables ?

- 2D Route + departure times ($\simeq$ 7000 flights).

How much can we reduce congestion in the French Airspace ?

- Approach based on optimization

What are our state space variables ?

- 2D Route + departure times ($\simeq$ 7000 flights).

What is our objective ?

- Airspace congestion minimization

How much can we reduce congestion in the French Airspace ?

- Approach based on optimization

What are our state space variables ?

- 2D Route + departure times ($\simeq$ 7000 flights).

What is our objective ?

- Airspace congestion minimization

What are the constraints ?

- Extra distance $\leq$ 10%
- Time shift have to be limited ($+-$ 45 minutes)
- The optimization process has to take into account flight connexions (hubs) and equity between airline.

Mathematical Modeling

A pair of decision variable (δ_i, r_i) is associated with each flight n.

$\delta_i \in \Delta_n \;\; r_i \in R_n$

$$\Delta_n = -\delta_m, -\delta_m + 1, ..., -1, 0, 1, ..., \delta_p - 1, \delta_p$$
$$R_n = r_0, r_1, r_2, ..., r_{max}$$

$(0, r_0)$: airline choice.

State point :

$$X = \begin{bmatrix} \delta_1 & \delta_2 & \dots & \delta_k & \dots & \delta_N \\ r_1 & r_2 & \dots & r_k & \dots & r_N \end{bmatrix}$$

Objective function

Congestion Minimization

$$\min y(X) = \min \sum_{k=1}^{k=P} \left((\sum_{t \in T} \widetilde{W}_{S_k}^t)^{\phi} \times (\max_{t \in T} \widetilde{W}_{S_k}^t)^{\varphi} \right)$$

$\max_{t \in T} \widetilde{W}_{S_k}^t$: is the maximum reported congestion.

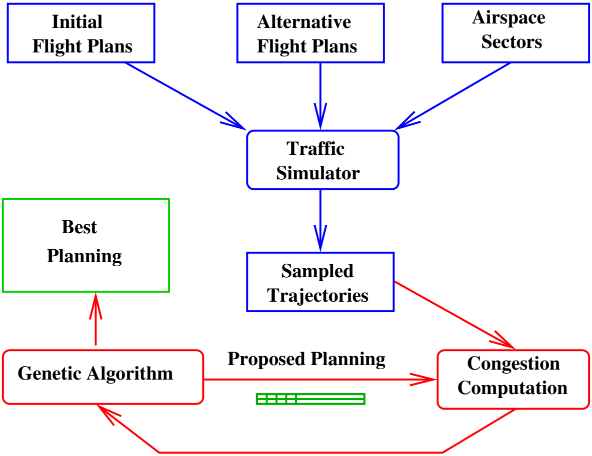
$\sum_{t \in T} \widetilde{W}_{S_k}^t$: is the sector cumulated congestion.

P is the number of elementary sectors, ϕ and φ are weight factors

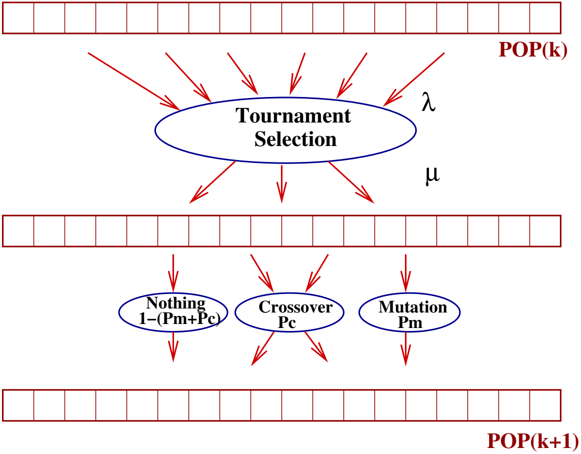
$$\max y_1(X) = \frac{y(X_{ref})}{y(X)}$$

($y_1 = 2$ means that the congestion has been divided by 2)

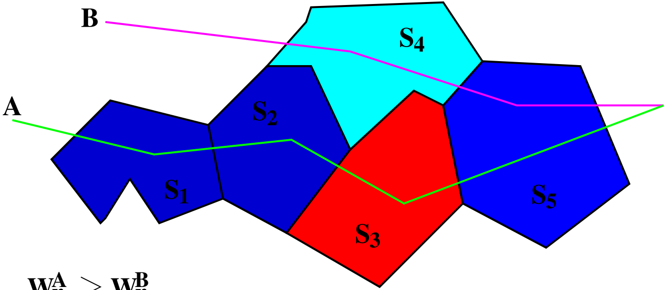
Simulation process



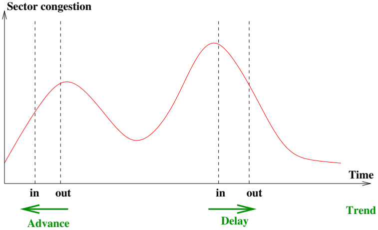
Genetic Algorithm



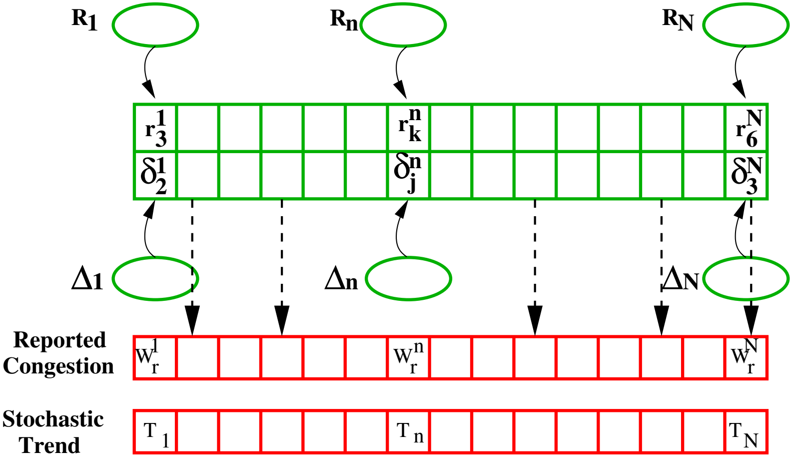
A Posteriori information



$W_r^A > W_r^B$



State space

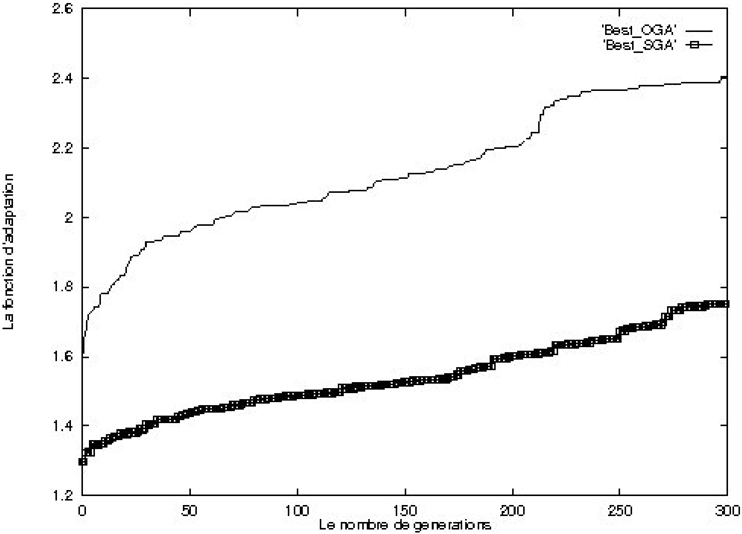

R1
Rn
RN
r 1 3
r n k
r N 6
δ 1 2
δ n j
δ N 3
Δ1
Δn
ΔN
Reported Congestion
W 1 r
W n r
W N r
Stochastic Trend
T 1
T n
T N

Test Features and Parameters

- **One day of traffic 6381 flights (june, 21 1996)**
- **89 elementary sectors with dynamic capacity**
- **Pop size : 50**
- **Generation number : 300**
- $\phi = 0.9$ **and** $\varphi = 0.1$
- **Max time shift : + or - 45 mn**
- **Alternative route with 10% extradistance**
- **6 computation hours on Pentium 1Ghz**

Evolution of best planning with generations

One day of traffic with ≃ 7000 flights optimized with GA



Multi-objective extension

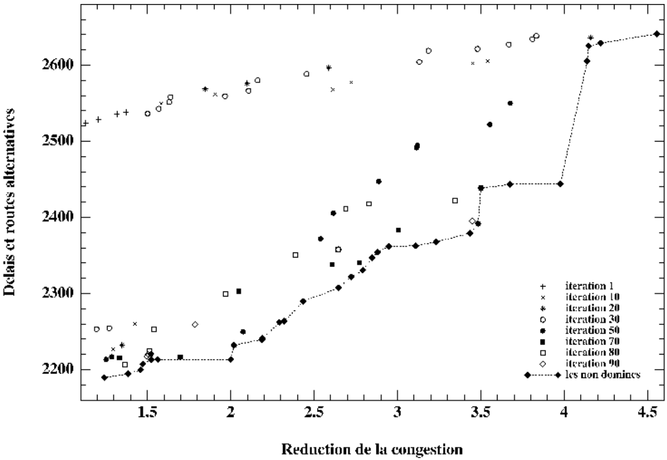
Delays and extra-distances minimization

- Delay on the ground : $\delta_s(i) = |t(i) - t_0(i)|$
- Delay on board : $\delta_r(i) = 3 * (T_r(i) - T_{r_0}(i))$
- Total delay : $\delta(i) = \delta_s(i) + \delta_r(i)$

$$\min y_2 = \sum_{i=1}^{N} \delta(i)^2$$

(the square insure equity)

Multi-objective extension



Strategic Conflict Free Planning

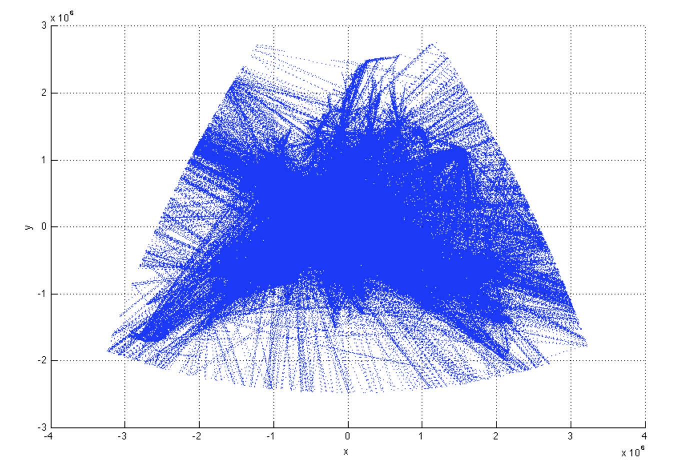
Optimization Approach

FP7 4D-CO project

Strategic Conflict Free Planning

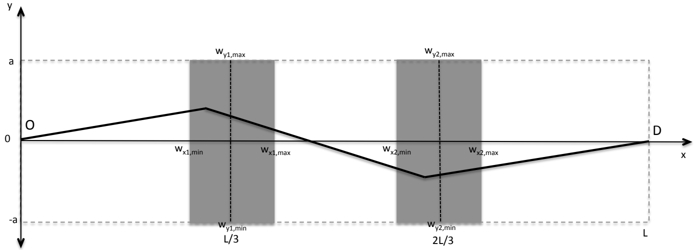
Consider the traffic over Europe ($\simeq$ 36000 flights)

Picture of Europe Traffic for One Day



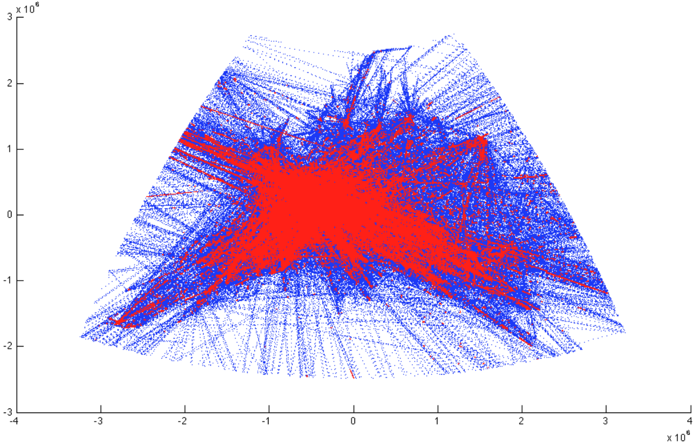
Strategic Conflict Free Planning

- We propose to design a gate-to-gate conflict free planning by adding waypoints and/or by shifting the time on departure.
- Departure and arrival segments are added to En-Route segments.
- Optimal altitude profiles have been used.
- Time shift : +- 30 minutes.
- Waypoint constraints : max 10% extra distance



Strategic Conflict Free Planning

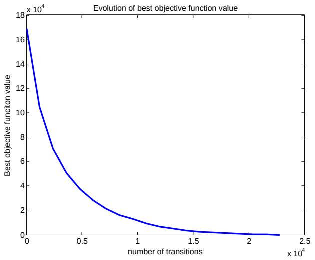
Direct route planning induces $\simeq$ 400000 interactions between trajectoires.



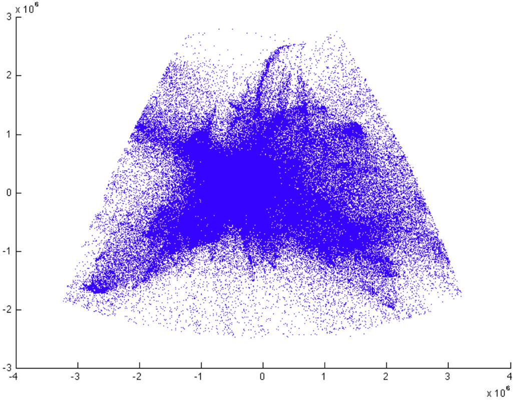
Strategic Conflict Free Planning

- This problem is NP_Hard
- One point of the state space requests 2GO memory space.

⇒ Simulated Annealing (20 minutes computing 2.4 Ghz intel CPU)



Strategic Conflict Free Planning



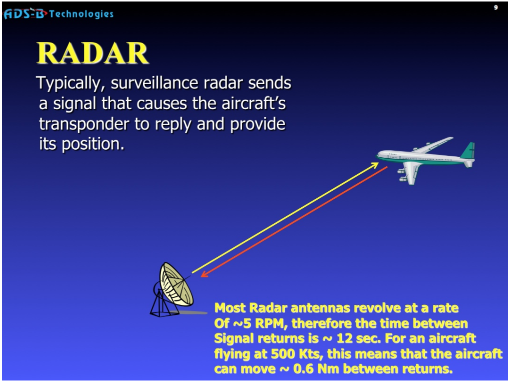
Oceanic Strategic Planning

Optimization Approach

ENAC

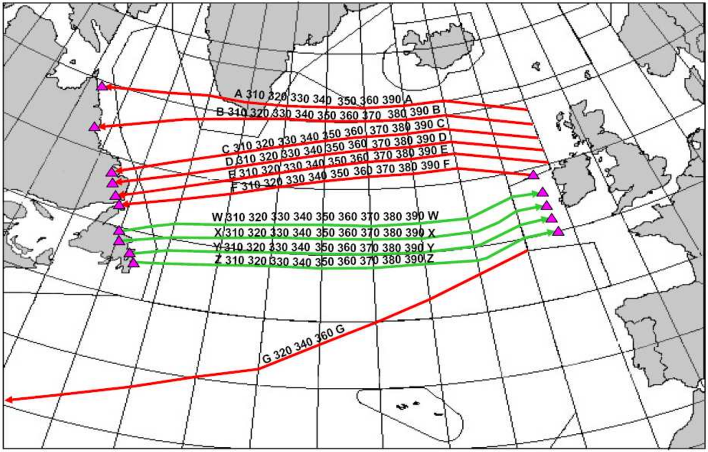
Oceanic Strategic Planning

- Continental Airspace ⇒ Radar

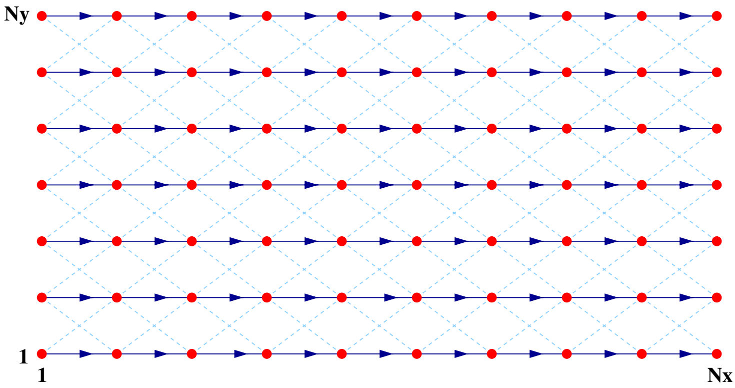


- Oceanic Airspace ⇒ Procedures based on oceanic tracks network

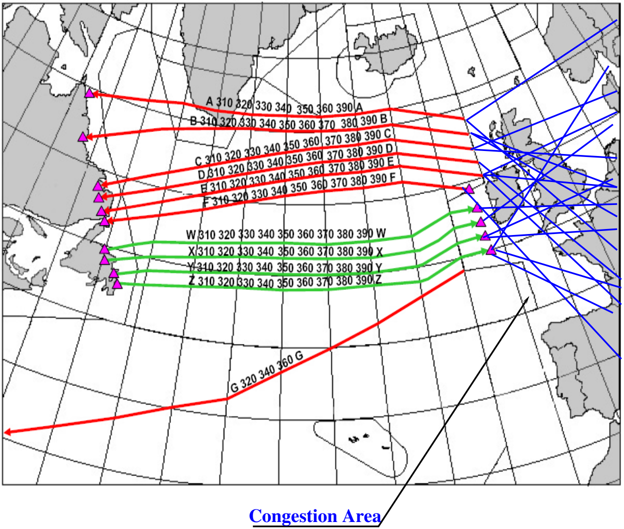
How It Works Today ?



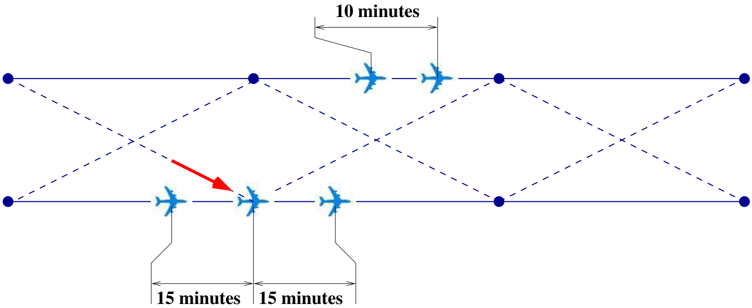
Oceanic Network Structure



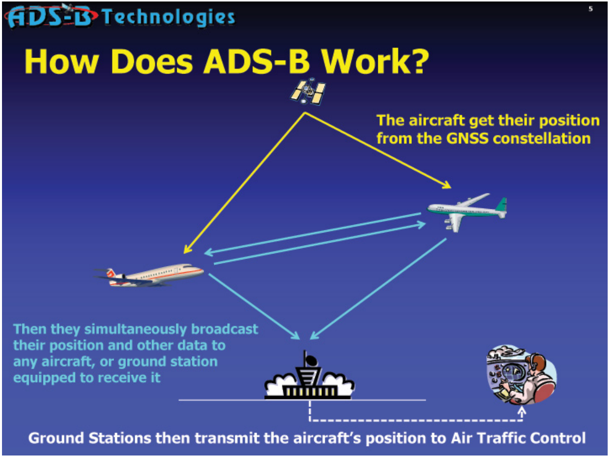
Network Limitation



Time Constraint for Oceanic Traffic

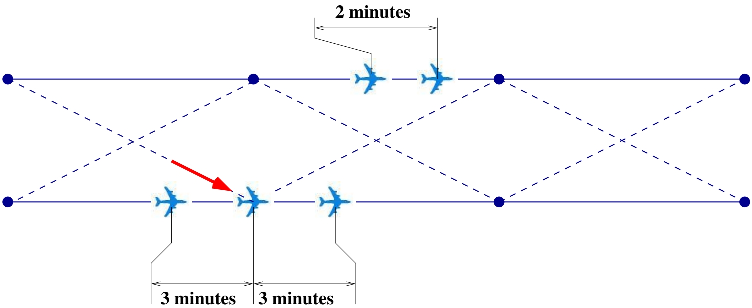


Automatic Dependent Surveillance-Broadcast



One measure every second

Time Constraint with ADSB



This new system increases the number of valid track changes and the maximum number of aircraft on the same track (wind optimal).

The model

- **Data** : For each flight $f \in \mathcal{F}$ we know

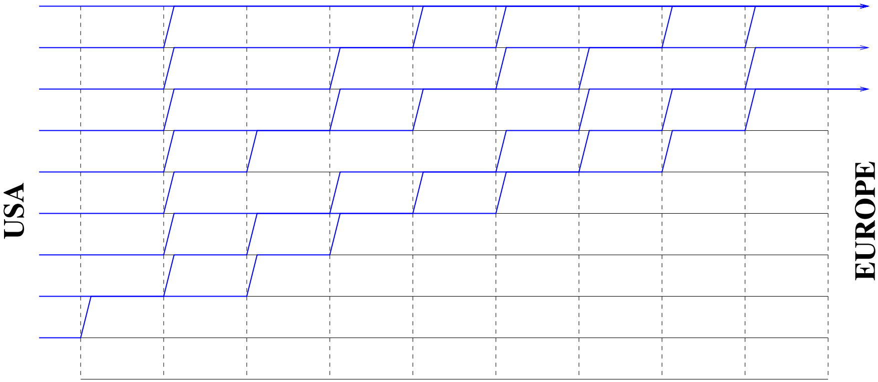
$Track^f_{in}$	the entry track
$Track^f_{out}$	the exit track
t^f_{in}	time of entrance in the track
FL^f_{in}	the input flight level
FL^f_{out}	the output flight level

- **Variables**

$$x_i^f = \begin{cases} 1 & \text{if flight } f \text{ changes track at waypoint } i \\ 0 & \text{otherwise} \end{cases}$$

δ^f : time shift at track entry : $t^f_{in} + \delta^f$

Altitude Profiles



Altitude profiles will be considered as constraints.

The model

- **Constraints**

$$\sum_{i=1}^{N_X-1} x_i^f = |Track_{out}^f - Track_{in}^f|$$

$$z_i^f = \begin{cases} 1 & \text{if flight } f \text{ changes flight level at waypoint } i \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{i=1}^{N_X-1} z_i^f = |FL_{out}^f - FL_{in}^f|$$

- **Objective function**
  Number of conflicts on nodes (Cf_n) and links (Cf_l).

Induced Combinatorics

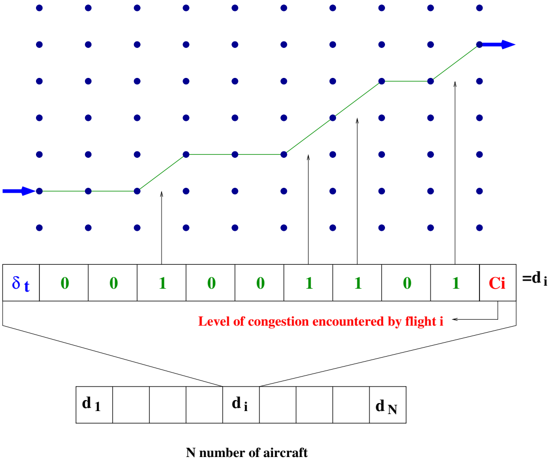
For each flight f we have the following

1. about 6 possible slots per flight.
2. an average of 4 track changes which have to be spread among the 10 waypoint positions ($=$ 210 options per flight)
3. the total number of options is about 1260.

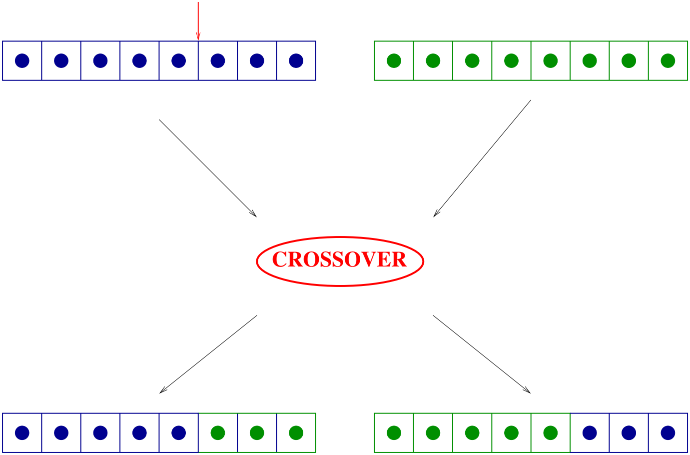
For 500 flights we have 1260^{500} options.

No separability $\Rightarrow$ Heuristic approach (EA)

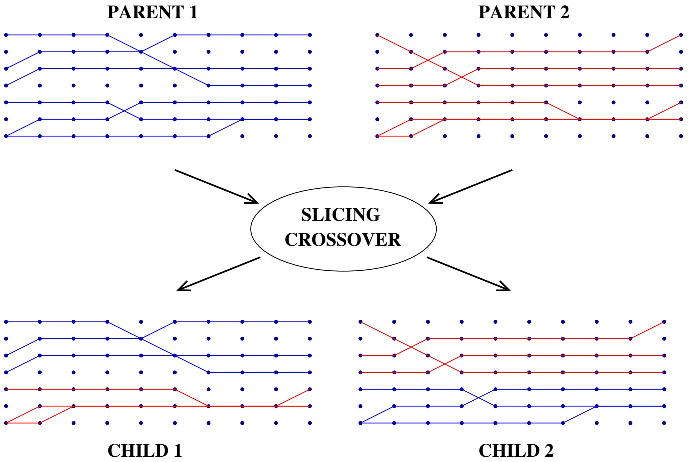
Coding


δ t
0
0
1
0
0
1
1
0
1
Ci
=d i
Level of congestion encountered by flight i
d 1
d i
d N
N number of aircraft

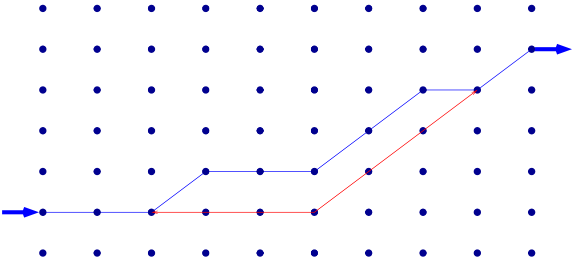
Slicing Crossover



Slicing Crossover



Mutation



δ_t	0	0	1	0	0	1	1	0	1	Ci

δ_t	0	0	0	0	0	1	1	1	1	Ci

Fitness Computation

Each aircraft trajectory is computed on the track network based on ;

- Altitude profile
- Aircraft speed
- Track changes decision variables
- Time delay at network entry (Max +/- 6x5=30 minutes)

Based on such simulation, we compute the conflicts on nodes (Cf_n) and on links (Cf_l).

$$fitness = \frac{1}{0.01 + Cf_n} + \frac{1}{0.01 + Cf_l}$$

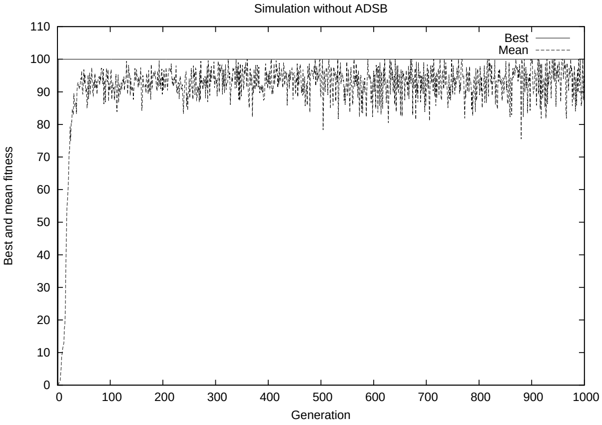
Test Framework

- 387 aircraft trajectories from August 4th 2006 (USA $\rightarrow$ Europe traffic)

Evolutionary Algorithm parameters

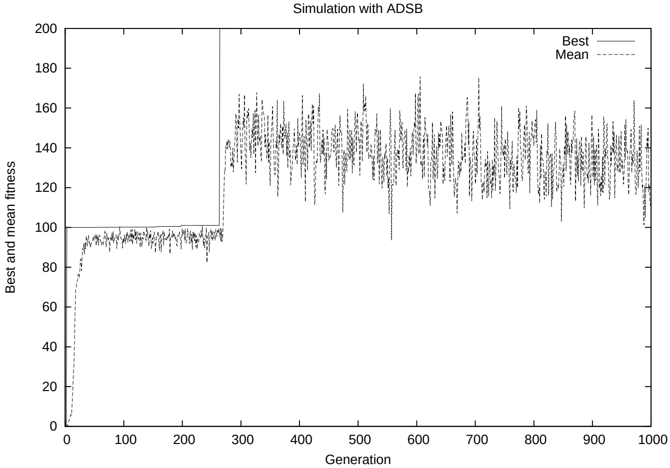
Pop size	500
Genration number	1000
Selection	$(\lambda = 6, \mu = 2)$
Proba Cross	0.5
Proba Mut	0.1

Results for Standard System



Remaining conflicts on nodes : 609 (initially 1515)

Results with ADSB Equiped Aircraft



Agenda

- Some Trajectory Models
- Strategic Trajectory Design
- Pre-Tactical Trajectory Design
- Tactical Trajectory Design
- Emergency Trajectory Design

Pre-Tactical Planning

After take-off (1, 2 hours planning)

Features

- 2D route design and speed control (state space)

Pre-Tactical Planning

After take-off (1, 2 hours planning)

Features

- 2D route design and speed control (state space)
- Congestion or weather areas avoidance (objective)

Wind Optimal Trajectory Design

Front Propagation Approach

Cap Gemini

What are our objectives ?

Currently

Using predefined air routes.

What are our objectives ?

Currently

Using predefined air routes.

$\Rightarrow$ **Proposed approach : Wind optimal route design.**

What are our objectives ?

Currently

Using predefined air routes.

⇒ **Proposed approach : Wind optimal route design.**

⇒ New problem :

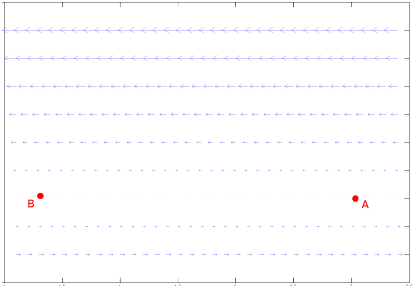
Optimization of aircraft trajectories based on weather conditions (wind) which avoid congestion areas (or bad weather phenomena, etc ...)

The optimization is based on Travel Time and (or) Fuel Consumption.

Statement of problem

Inputs

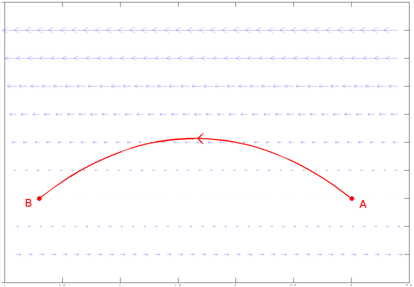
- Start point A,
  End point B ;
- Constant aircraft speed ;
- Wind forecast ;
- Areas to avoid.



Statement of problem

Inputs

- Start point A,
  End point B ;
- Constant aircraft speed ;
- Wind forecast ;
- Areas to avoid.



⇒ **Goal :** Connect the point A to the point B in order to minimize the travel time.

Adaptation of the Fast Marching Method

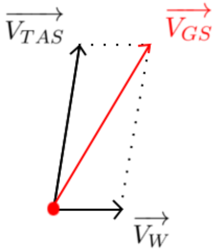


FIGURE: Speed

$$\boxed{\overrightarrow{V_{GS}} = \overrightarrow{V_{TAS}} + \overrightarrow{V_W}}$$

with :

- V_{TAS} (True Airspeed) : speed of the aircraft relative to the airmass in which it is flying ;
- V_W (Wind Speed) ;
- V_{GS} (Ground Speed).

Adaptation of the Fast Marching Method

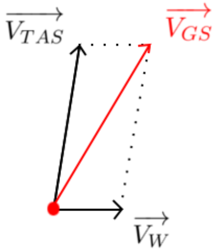


FIGURE: Speed

$$\boxed{\overrightarrow{V_{GS}} = \overrightarrow{V_{TAS}} + \overrightarrow{V_W}}$$

with :

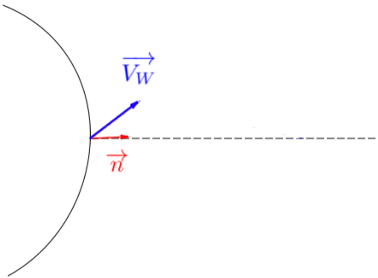
- V_{TAS} (True Airspeed) : speed of the aircraft relative to the airmass in which it is flying ;
- V_W (Wind Speed) ;
- V_{GS} (Ground Speed).

$\Rightarrow$ **The aircraft ground speed is function of the direction !**

$\Rightarrow$ **Anisotropic problem.**

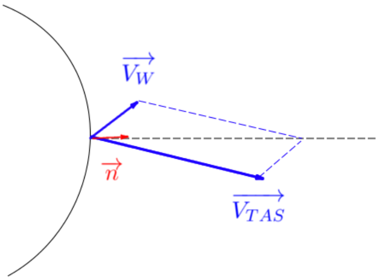
Calculation of the speed function : $F = ||\overrightarrow{F}||$

Calculation of the aircraft speed in the normal direction.



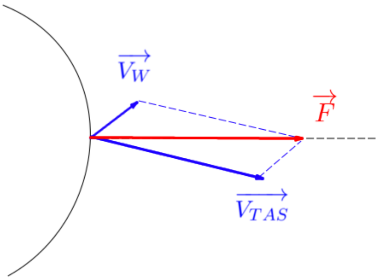
Calculation of the speed function : $F = || \overrightarrow{F} ||$

Calculation of the aircraft speed in the normal direction.



Calculation of the speed function : $F = || \overrightarrow{F} ||$

Calculation of the aircraft speed in the normal direction.

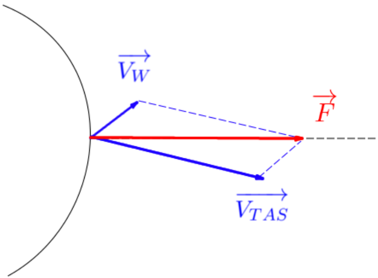


Calculation of the speed function : $F = ||\overrightarrow{F}||$

Calculation of the aircraft speed in the normal direction.

Calculation of the cost u :

$$\|\nabla u\| = \frac{1}{||\overrightarrow{F}||}$$

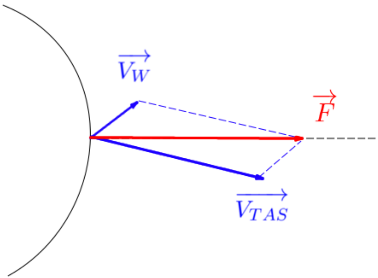


Calculation of the speed function : $F = ||\overrightarrow{F}||$

Calculation of the aircraft speed in the normal direction.

Calculation of the cost u :

$$\|\nabla u\| = \frac{1}{||\overrightarrow{F}||}$$



To plan the optimal path :

$$\frac{dX}{dt} = -\overrightarrow{V_W} - V_{TAS}\frac{\nabla u}{||\nabla u||}$$

Taking into account obstacles and weather conditions

$$\|\nabla u(x)\| = \frac{1}{F(x)}$$

$\Rightarrow$ Change of the propagation speed according to obstacles :

$$\|\nabla u(x)\| = \frac{1}{((1-\alpha(x))F(x))}$$

with $\alpha(x) \in [0; \alpha_0]$ and $0 \leqslant \alpha_0 < 1$.

Interpretation :

$\alpha(x) = \alpha_0$: forbidden areas
$\alpha(x) = 0$: free areas
$0 \leq \alpha(x) \leq \alpha_0$ penalized areas

Example with obstacles

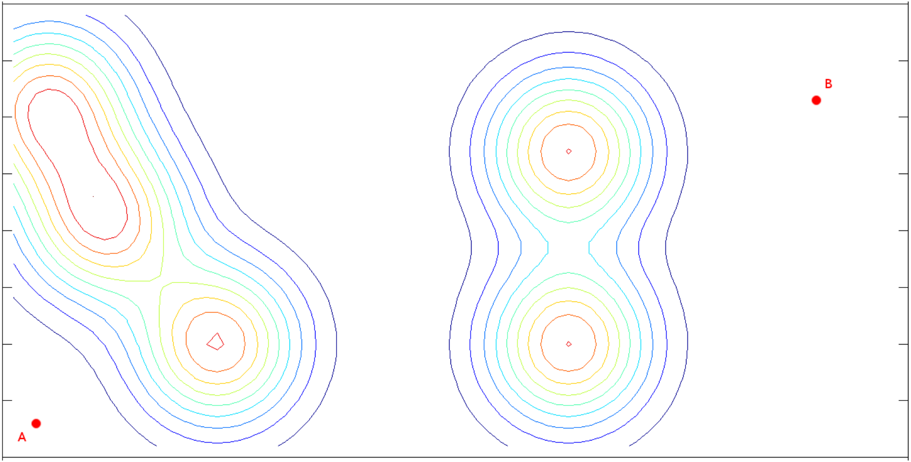


FIGURE: **Obstacles** (Forbidden areas then coefficient decreasing to 0.)

Example with obstacles

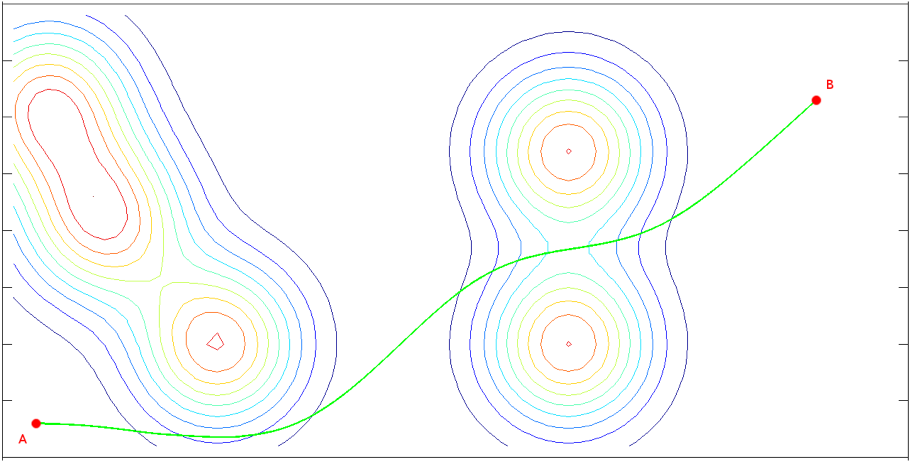


FIGURE: Optimal trajectory (green) without wind

Example with obstacles

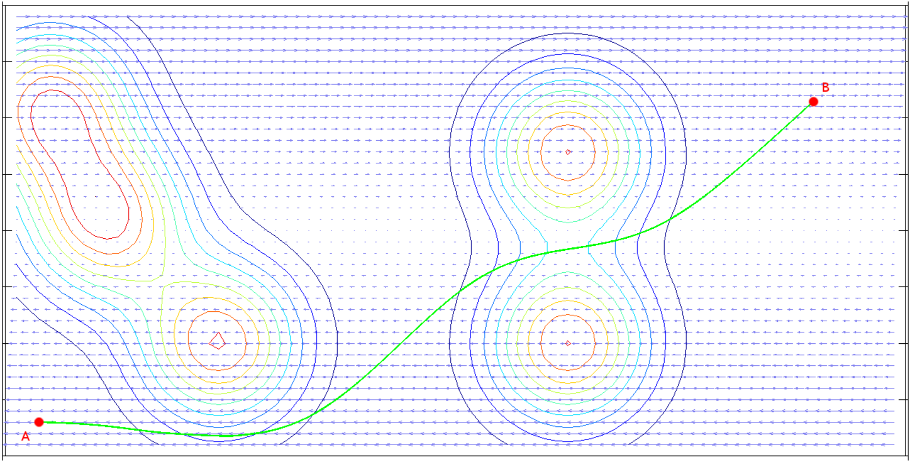


FIGURE: Wind

Example with obstacles

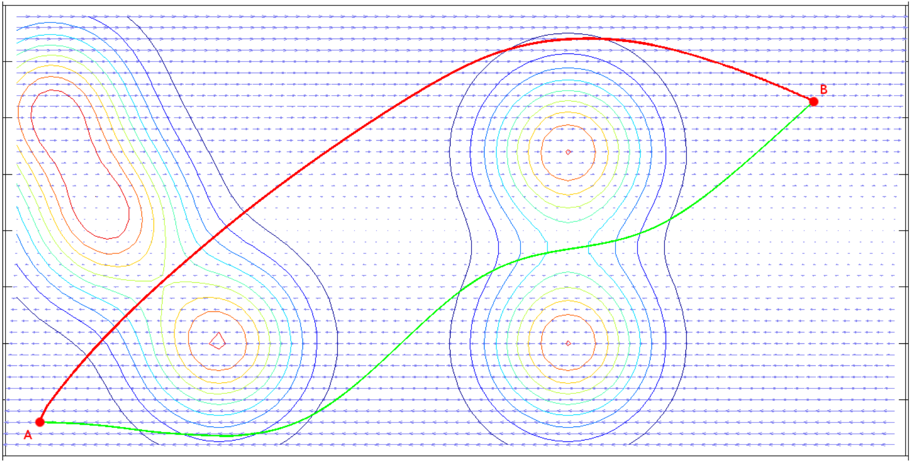


FIGURE: Optimal trajectories : with wind and without wind.

Wave Propagation Algorithm for Trajectory Design

- Aircraft Trajectory Design in a Wind Field
- Light Propagation Algorithm AIRBUS FMS Division

The light propagation method

The light propagation analogy

- Light follows Geodesic in time thereby avoiding areas of high index.

The light propagation method

The light propagation analogy

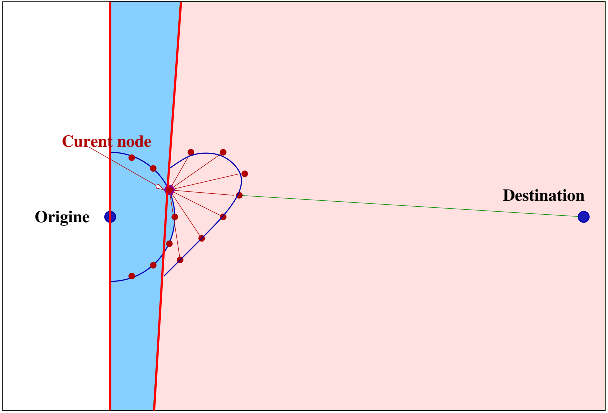
- Light follows Geodesic in time thereby avoiding areas of high index.
- Light propagation is controlled by the Descarte law.

The light propagation method

The light propagation analogy

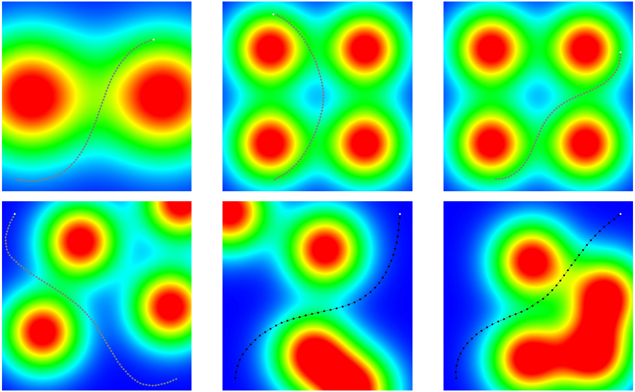
- Light follows Geodesic in time thereby avoiding areas of high index.
- Light propagation is controlled by the Descarte law.
- Trajectory planning can be achieved by computing wavefronts.

Principles of the light propagation method



Geodesic computation (A^* like algorithm or Triangle mesh algorithm)

Experimental results



Agenda

- Some Trajectory Models
- Strategic Trajectory Design
- Pre-Tactical Trajectory Design
- Tactical Trajectory Design
- Emergency Trajectory Design

Tactical Planning

After take-off (horizon : 20 minutes))

Features

- 2D Route design (state space)

Tactical Planning

After take-off (horizon : 20 minutes))

Features

- 2D Route design (state space)
- Collision avoidance (objective)

Tactical Planning

After take-off (horizon : 20 minutes))

Features

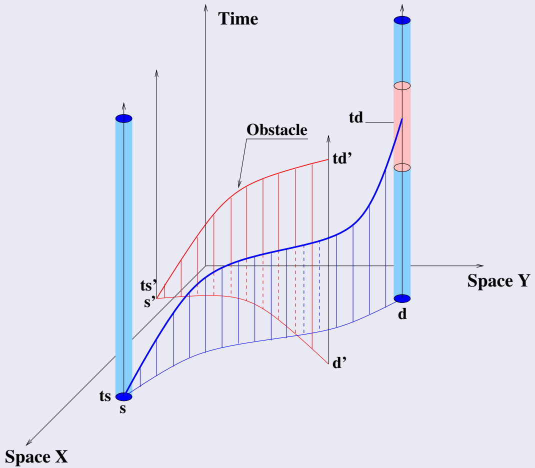
- 2D Route design (state space)
- Collision avoidance (objective)
- One must bring a proof for such algorithms

Tactical Trajectory Design

- Time extension of light Propagation Algorithm
- Approach based on B-Splines
- Approach based biharmonic navigation functions

Approach Based on LPA

Time extension for dynamic obstacles



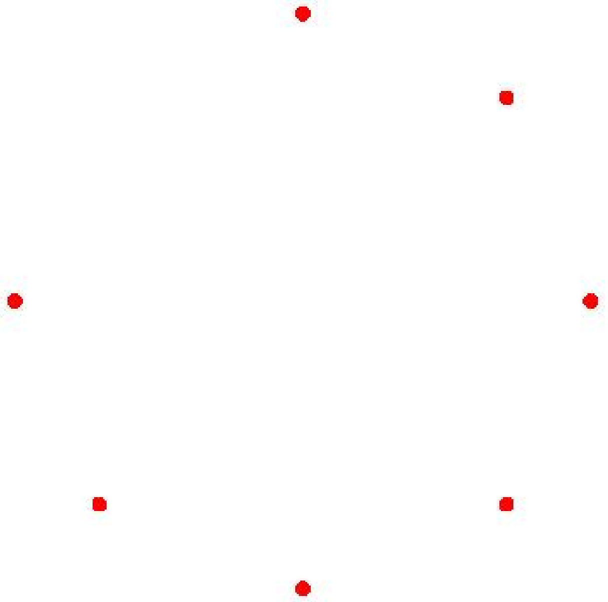
Light has to propagate one way in time dimension

Experimental results

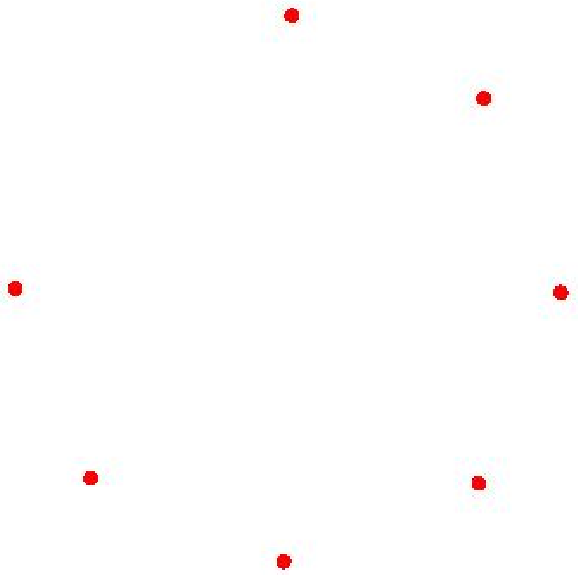
A 2D + time algorithm version

- The algorithm sequentially control conflicting aircraft.
- The aircraft are represented by high index discs of radius the standard separation.

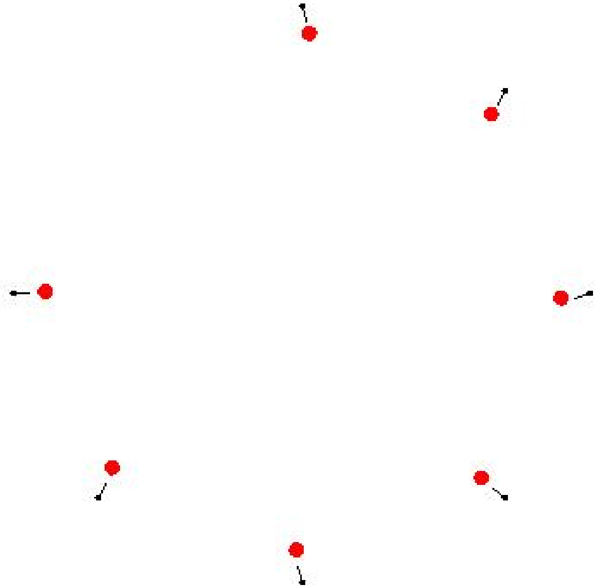
7 Conflicting Aircraft



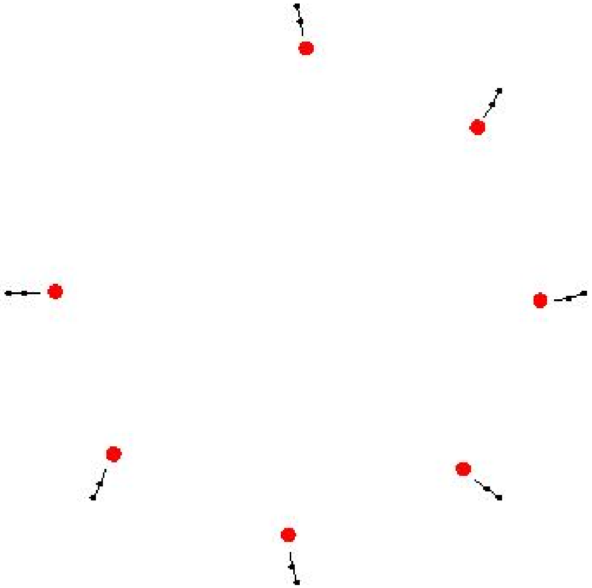
7 Conflicting Aircraft



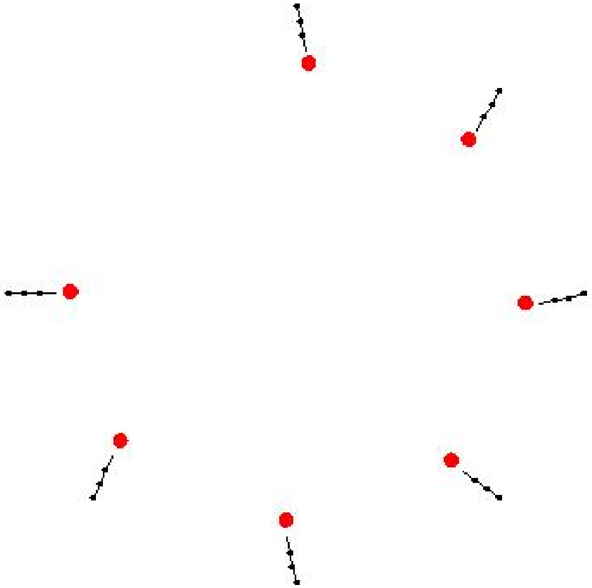
7 Conflicting Aircraft



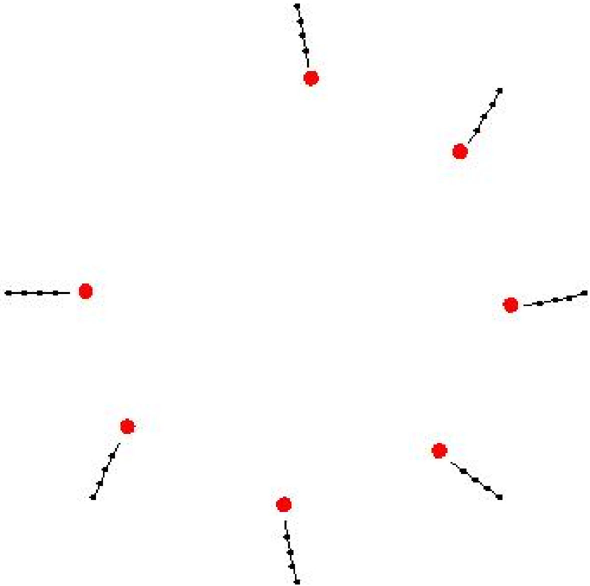
7 Conflicting Aircraft



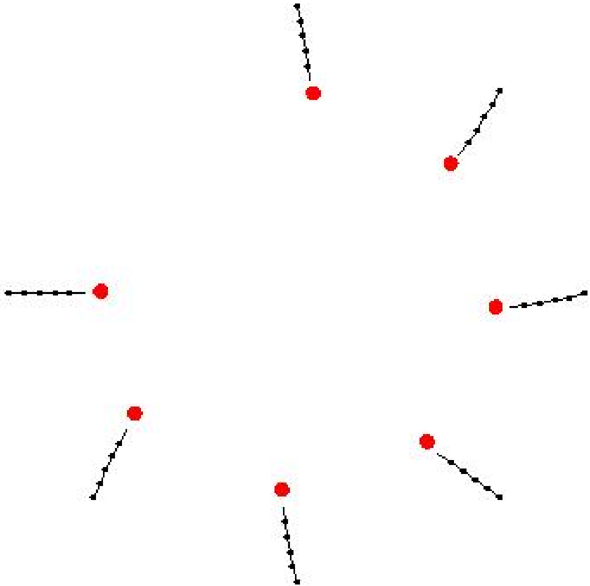
7 Conflicting Aircraft



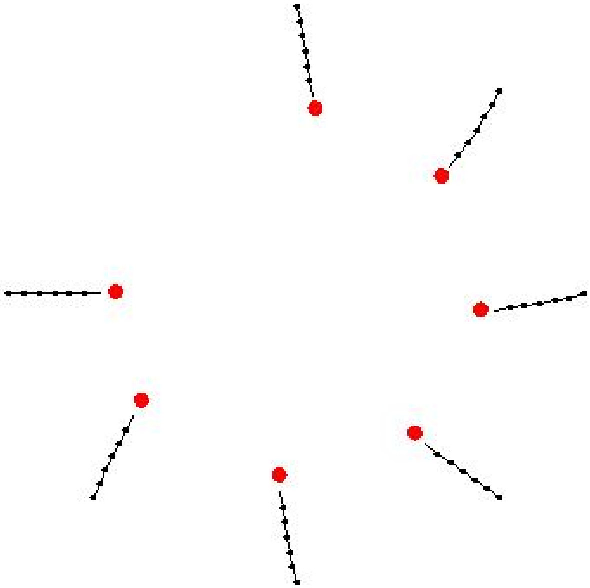
7 Conflicting Aircraft



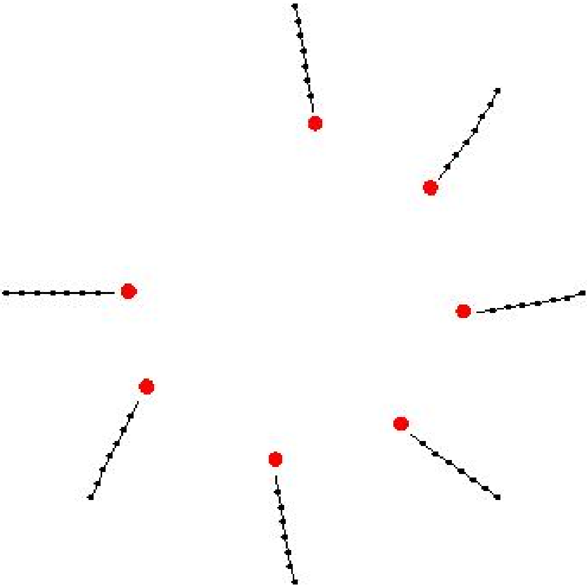
7 Conflicting Aircraft



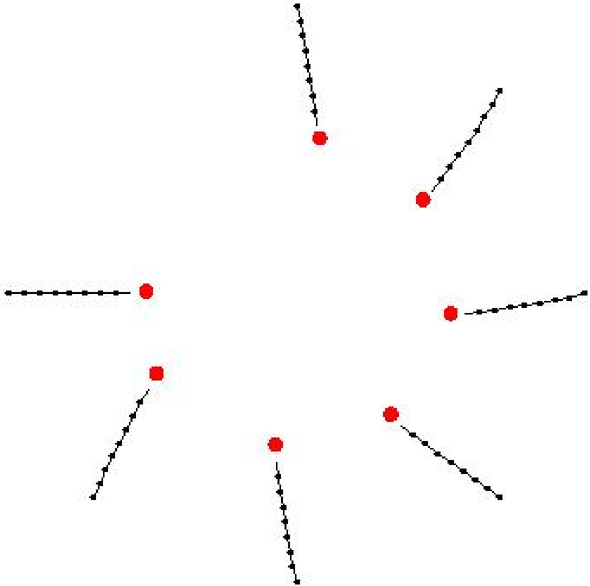
7 Conflicting Aircraft



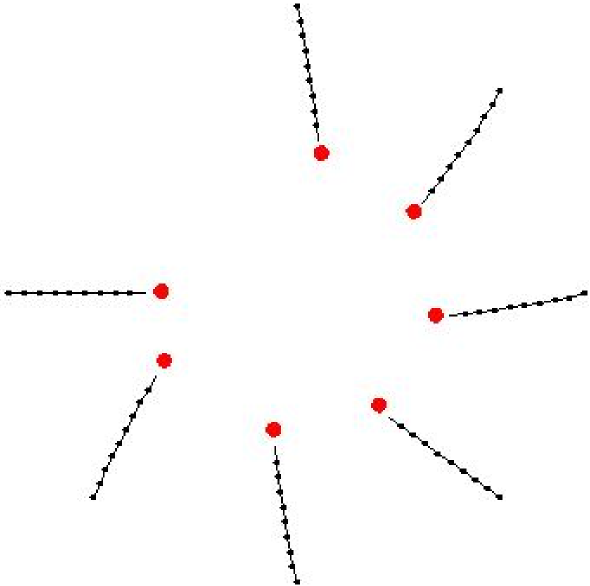
7 Conflicting Aircraft



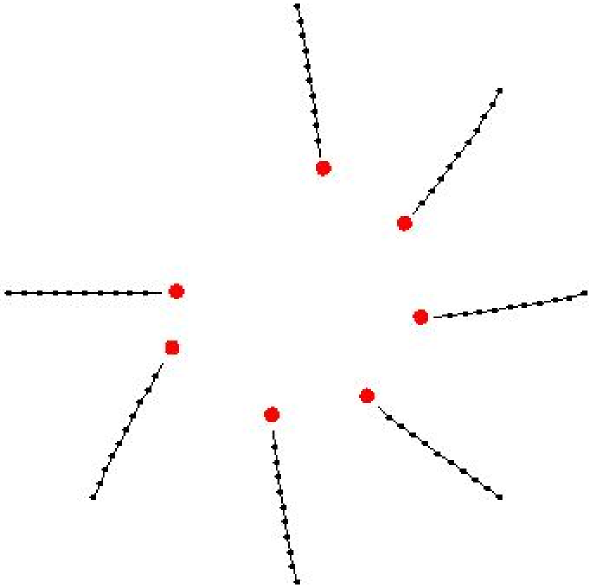
7 Conflicting Aircraft



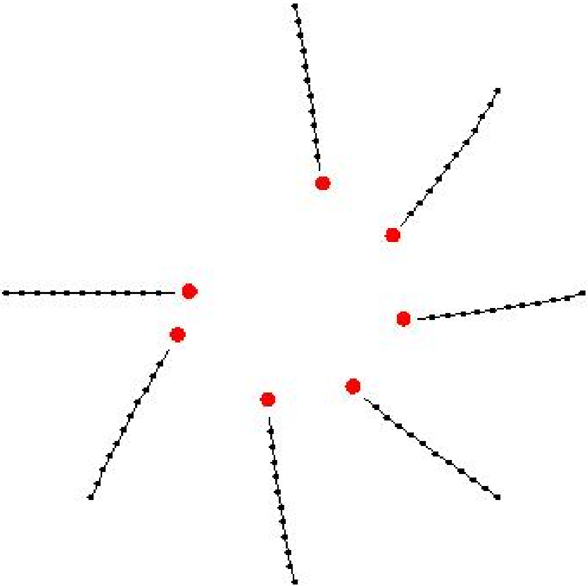
7 Conflicting Aircraft



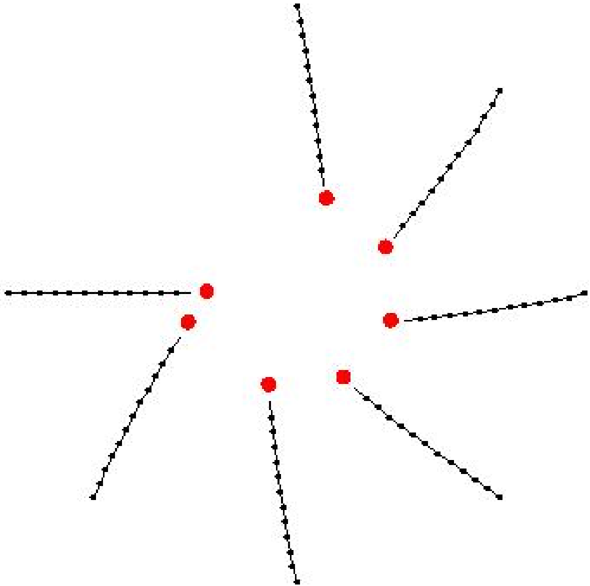
7 Conflicting Aircraft



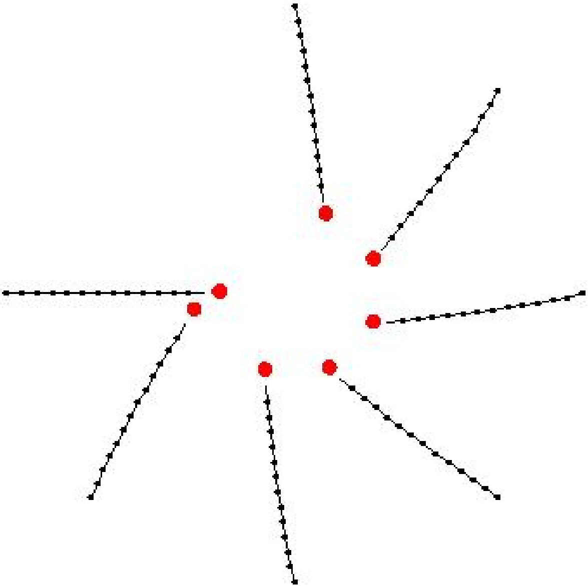
7 Conflicting Aircraft



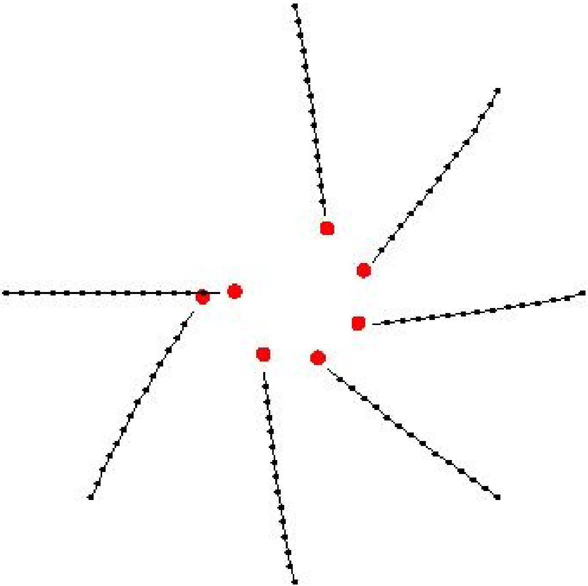
7 Conflicting Aircraft



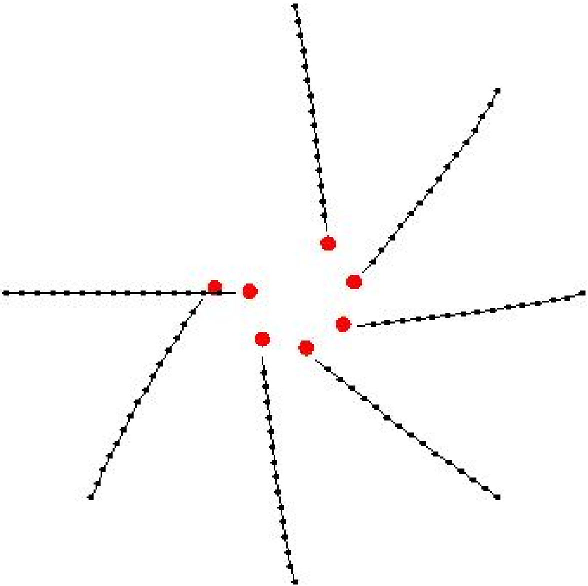
7 Conflicting Aircraft



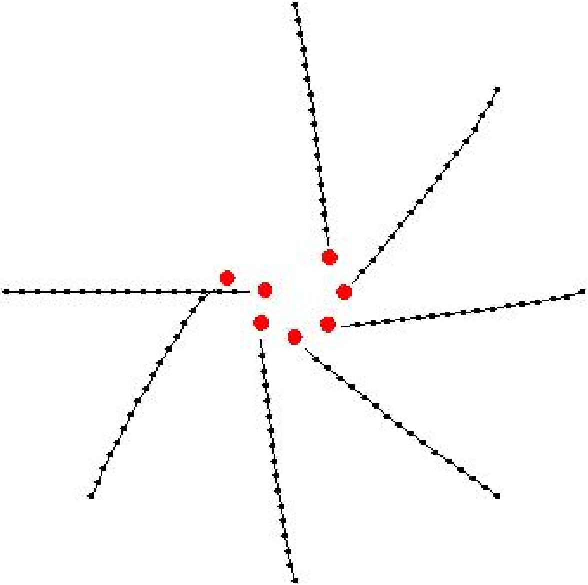
7 Conflicting Aircraft



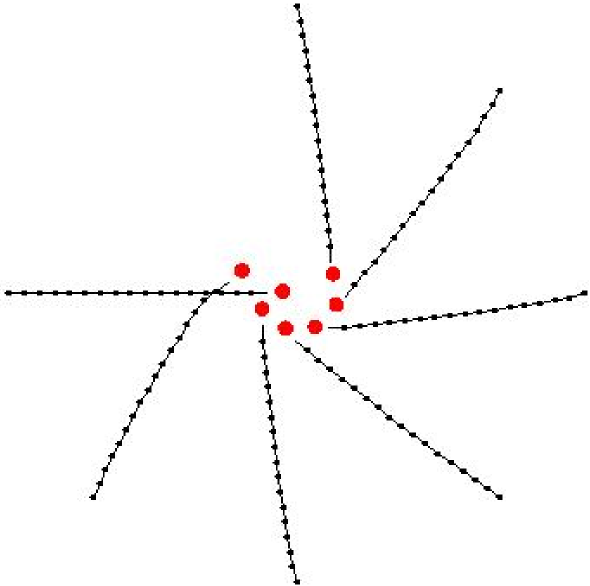
7 Conflicting Aircraft



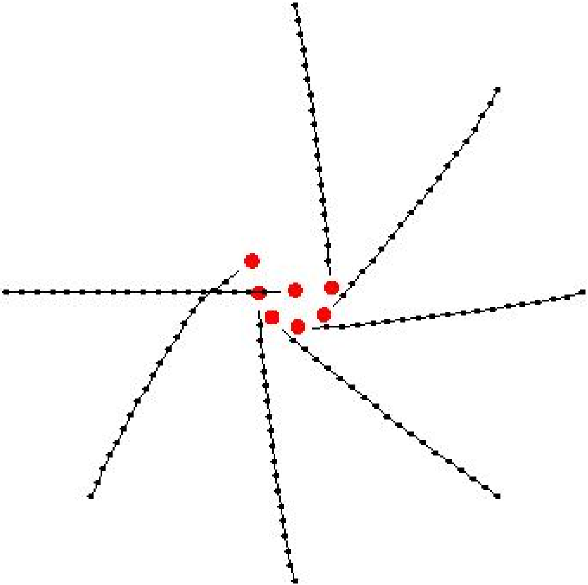
7 Conflicting Aircraft



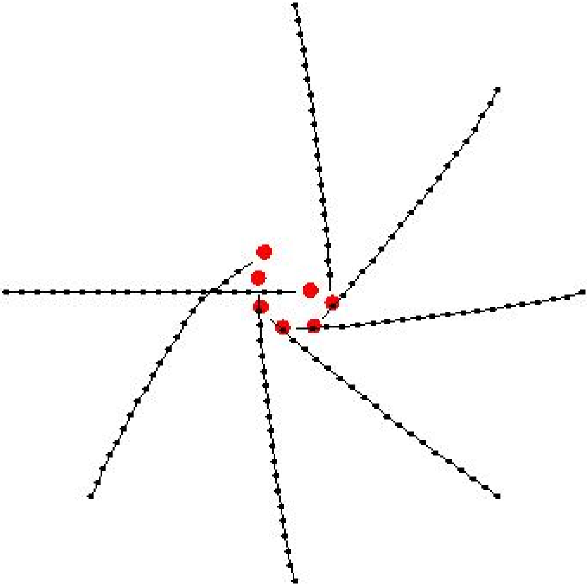
7 Conflicting Aircraft



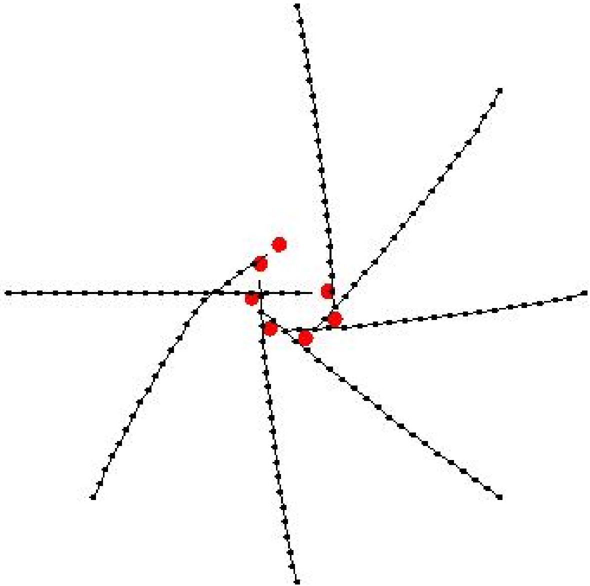
7 Conflicting Aircraft



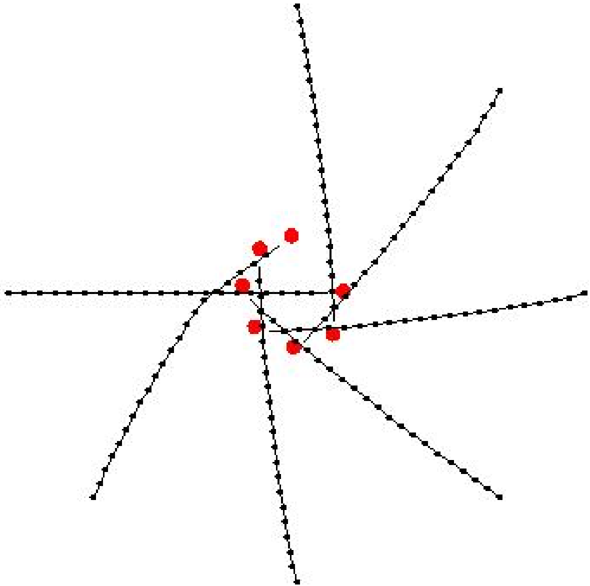
7 Conflicting Aircraft



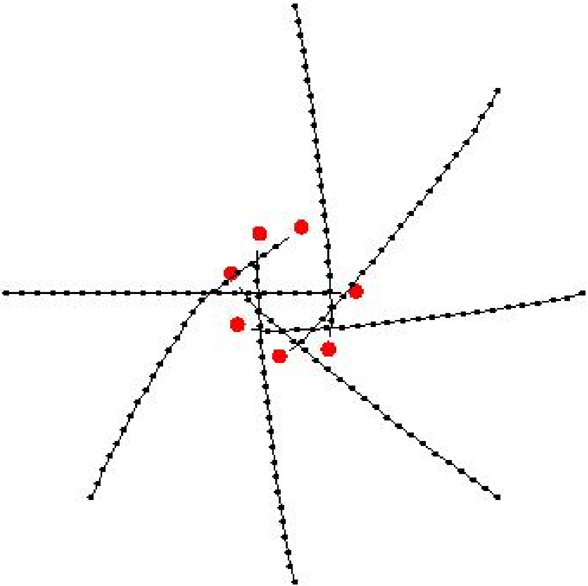
7 Conflicting Aircraft



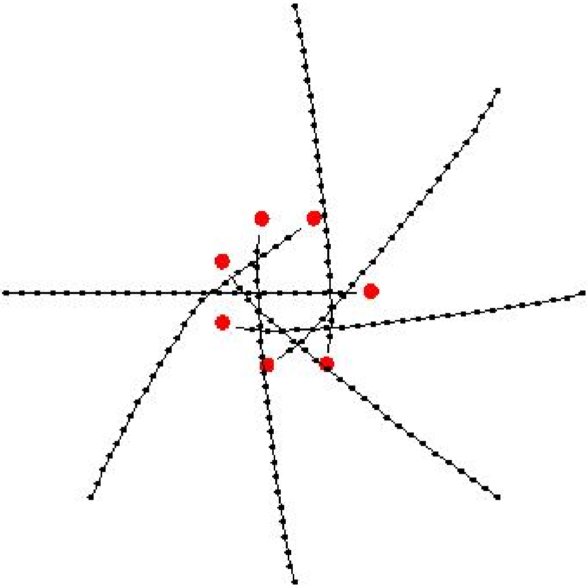
7 Conflicting Aircraft



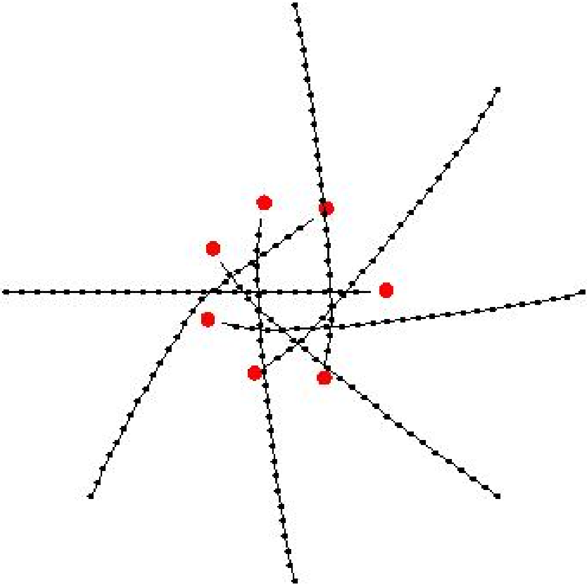
7 Conflicting Aircraft



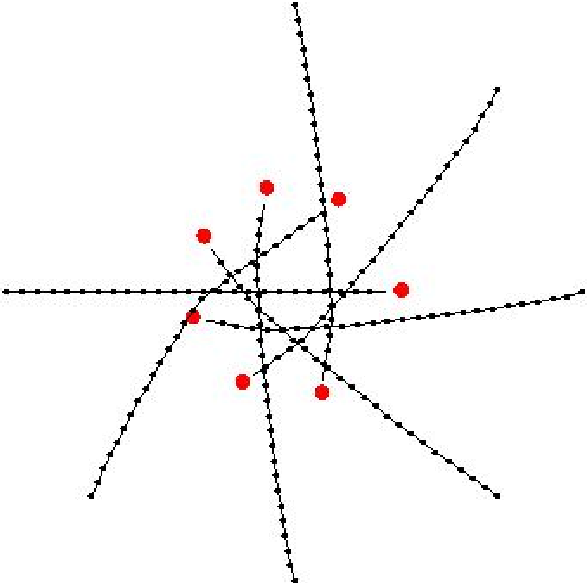
7 Conflicting Aircraft



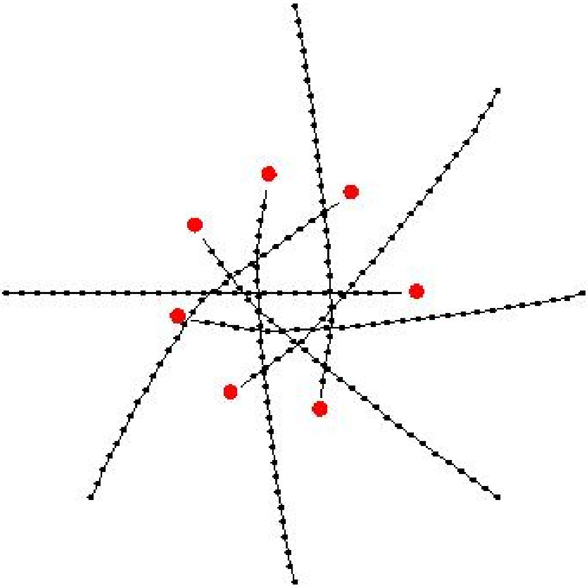
7 Conflicting Aircraft



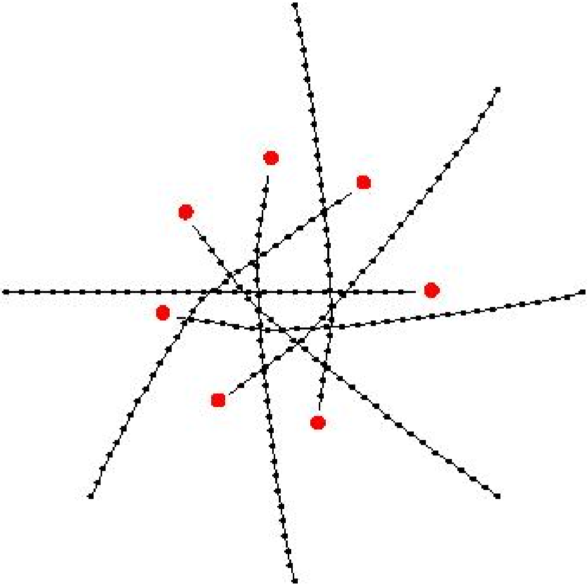
7 Conflicting Aircraft



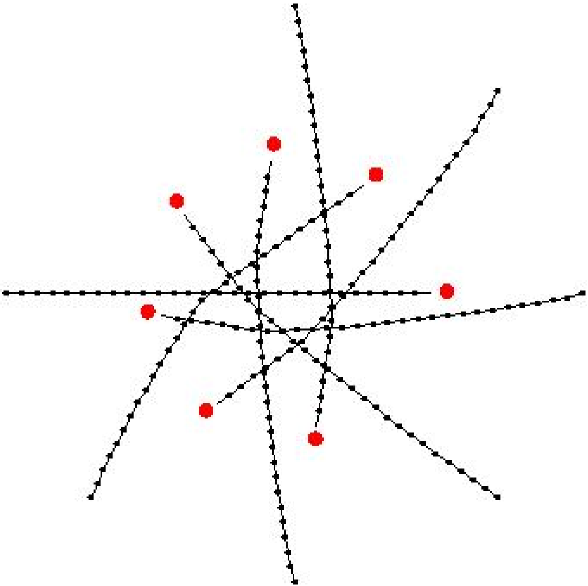
7 Conflicting Aircraft



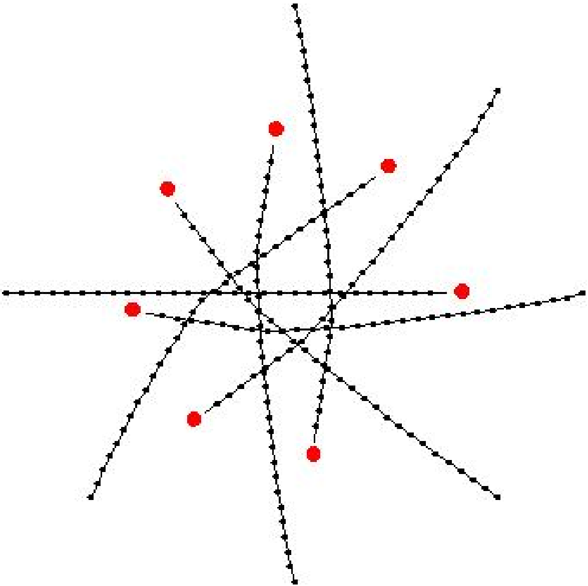
7 Conflicting Aircraft



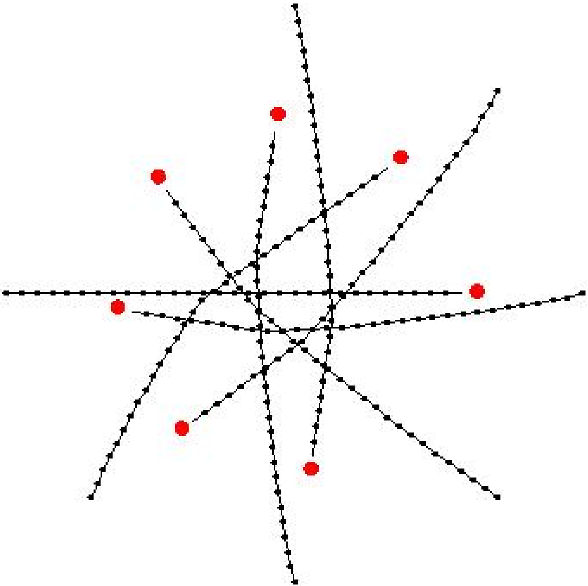
7 Conflicting Aircraft



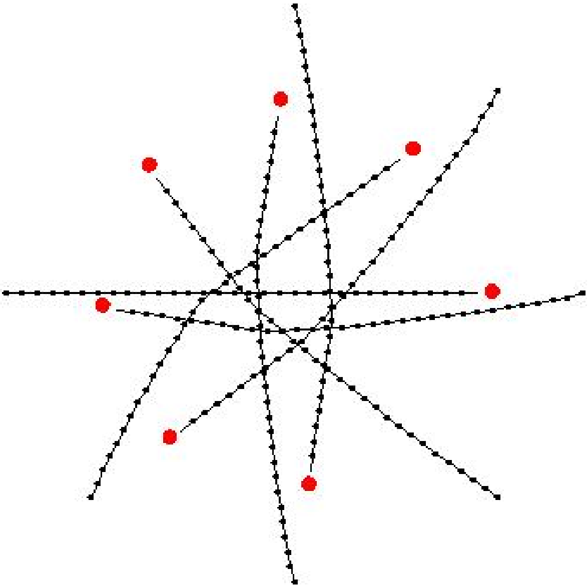
7 Conflicting Aircraft



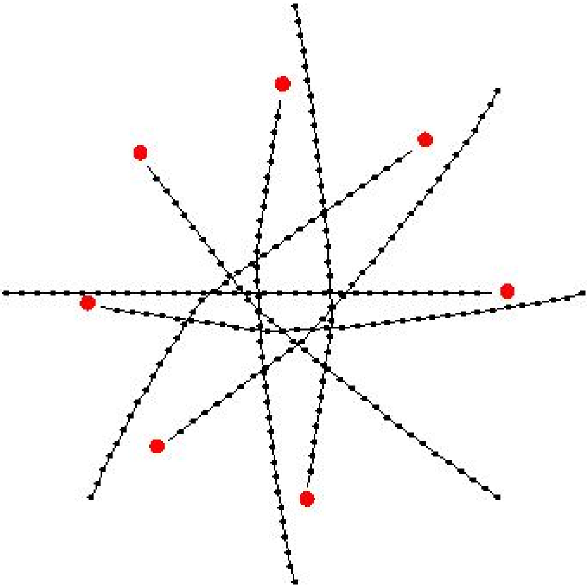
7 Conflicting Aircraft



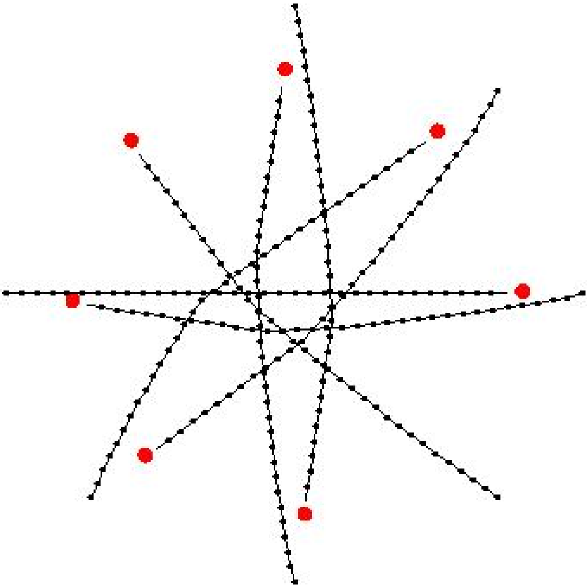
7 Conflicting Aircraft



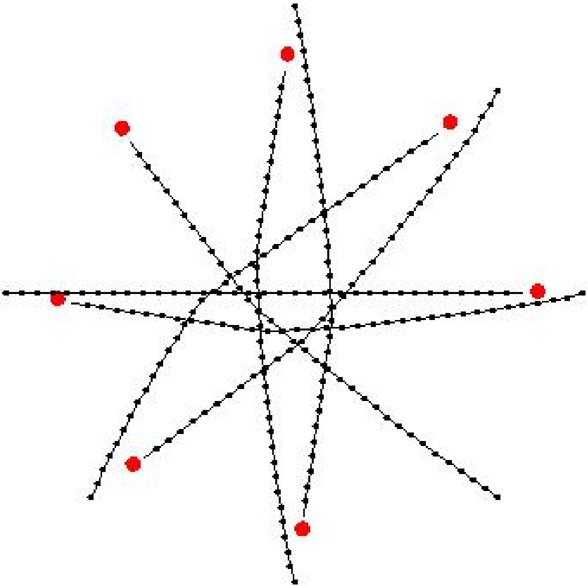
7 Conflicting Aircraft



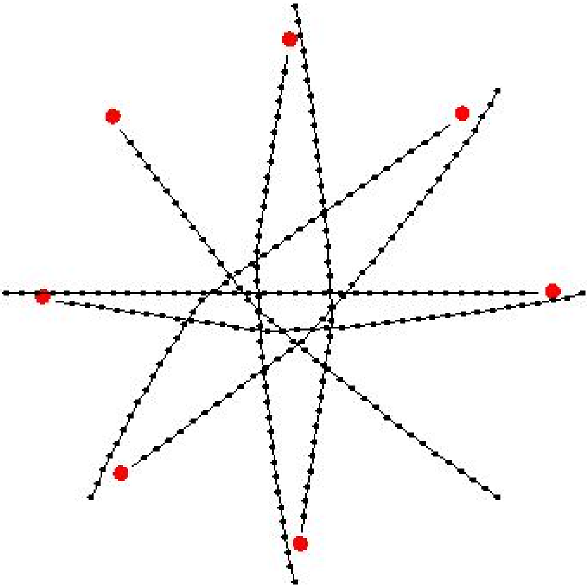
7 Conflicting Aircraft



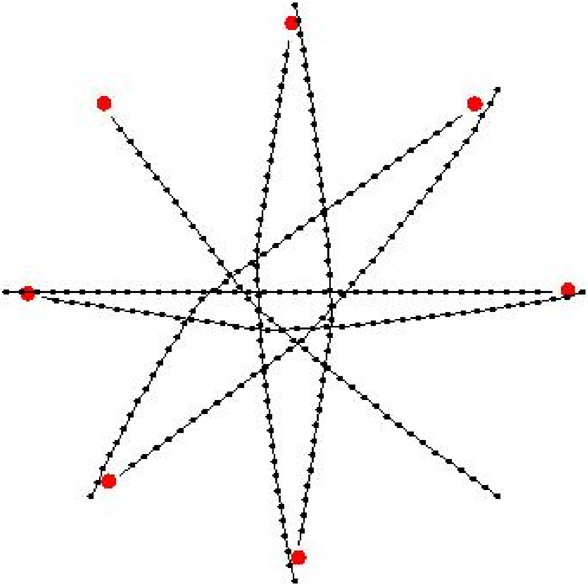
7 Conflicting Aircraft



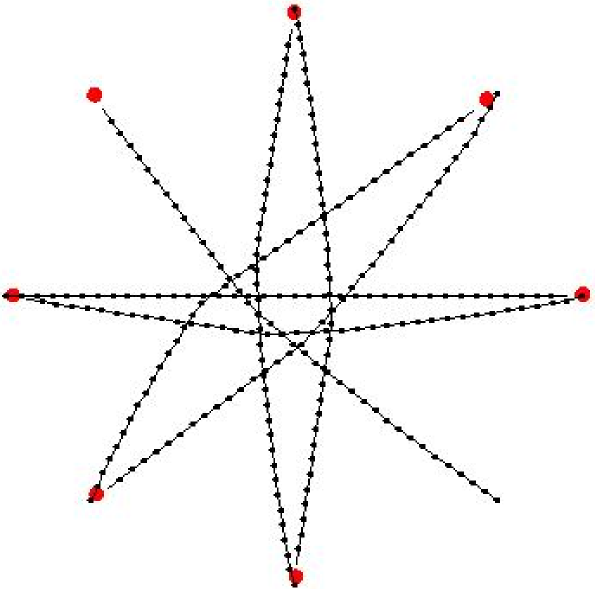
7 Conflicting Aircraft



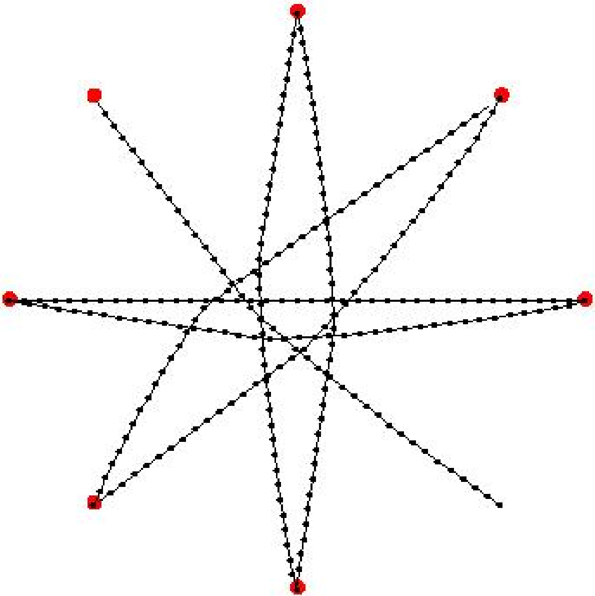
7 Conflicting Aircraft



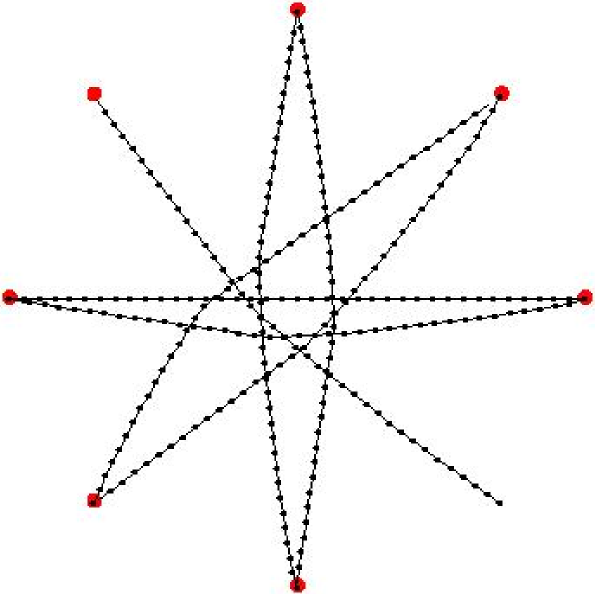
7 Conflicting Aircraft



7 Conflicting Aircraft



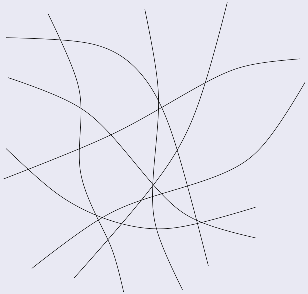
7 Conflicting Aircraft



Conflict Resolution for a traffic day

How does it work ?

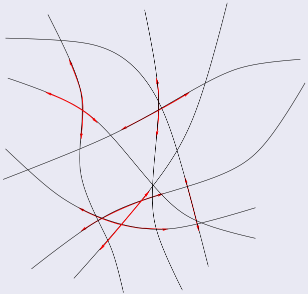
We compute aircraft trajectories for a day of traffic over France.



Conflict Resolution for a traffic day

How does it work ?

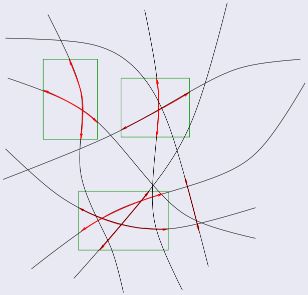
We extract trajectories segments between t et $t + 21$ min.



Conflict Resolution for a traffic day

How does it work ?

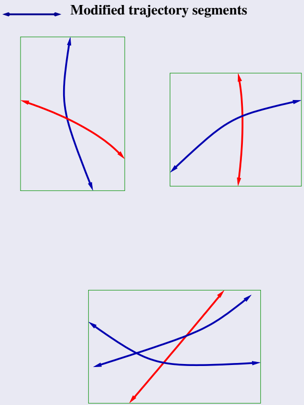
We identify clusters of conflict.



Conflict Resolution for a traffic day

How does it work ?

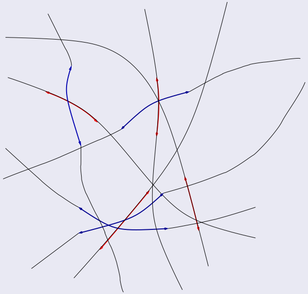
We solve conflicts within each cluster using the light propagation algorithm.



Conflict Resolution for a traffic day

How does it work ?

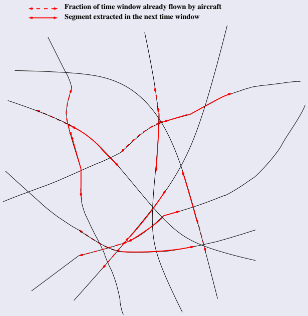
We reintroduce the new segments in the database and we recompute the remaining parts of trajectories.



Conflict Resolution for a traffic day

How does it work ?

The time window is slid by 7 min. $t \leftarrow t + 7$.



Conflict Resolution for a full day of traffic

Numerical Results

The 8/12/2008 traffic day was tested with 8212 aircraft.

- 3344 clusters.
- 99% of clusters were resolved (the last % is due to aircraft already in conflict when algorithm starts ; could be solve initial time shifting
- Number of modified trajectories is 1501.
- Average extension distance= -4.41 Nm.

Stochastic Extension

Open loop FMS error has been used for our simulation (+-15 Nm after 1 Hour)

- This algorithm has been extended with such uncertainties and is able to manage 98% of the conflicts.
- The remaining 2% have been solve by RTA setting (closed FMS mode).

Tactical Trajectory Design

- Time extension of light Propagation Algorithm
- Approach based on B-SplinesCap Gemini
- Approach based on biharmonic navigation functions

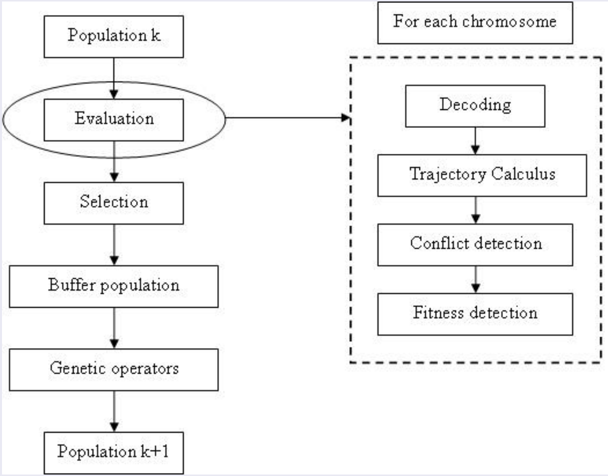
Problem presentation

Our methodology

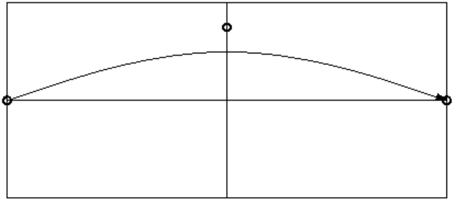
- A combination of an optimization method and a smooth trajectory model : B-splines.
- **B-splines are controlled by the optimization method via their control points**

Genetic Algorithm

Structure



Trajectory model



Semi-infinite programming formulation

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & g(x;t) > \alpha \qquad \forall t \in [t_1, t_2] \end{aligned} \tag{2}$$

- where t is continuous, it is the semi-infinite parameter.

Semi-infinite programming formulation

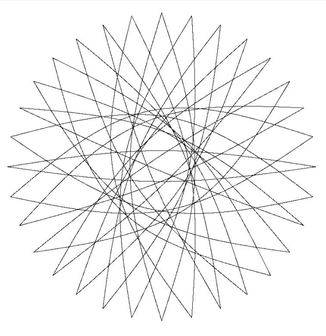
- Our objective function : relative distance increase.
- Insure standard separation between each pair of aircraft at all time

$$c^{ij}(u;t) = \|\gamma^{\beta^i(u)}(s(t)) - \gamma^{\beta^j(u)}(s(t))\|_2 > \tau \qquad \forall t \in [0, t^{ij}_{max}]$$

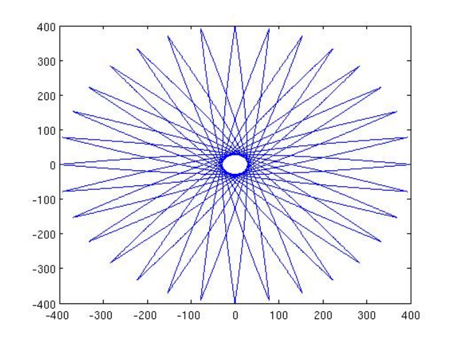
SIP is a local optimization method

Results and comparison

32 aircrafts situation



Genetic Algorithm



Semi-infinite programming.

Next : use GA to initialize control points for SIP

Tactical Trajectory Design

- Time extension of light Propagation Algorithm
- Approach based on B-Splines
- Approach based on biharmonic navigation functionsCap Gemini

Collision-free trajectory planning using biharmonic navigation functions

Objective

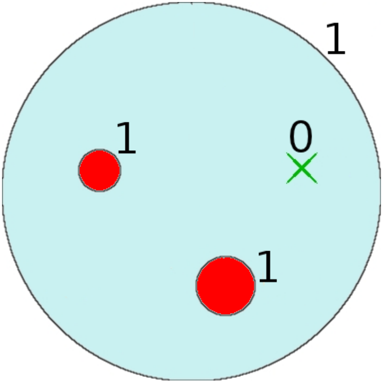
- Create trajectories guaranteeing obstacle avoidance and enforcing ATM constraints for several aircraft.

Constraints

1. Speed has to stay in a given range
2. Trajectories have be smooth

Navigation Function

Potential Field Analogy in order to compute the navigation function ϕ.



Navigation function and navigation field

The navigation field is given by : $-\nabla\phi$

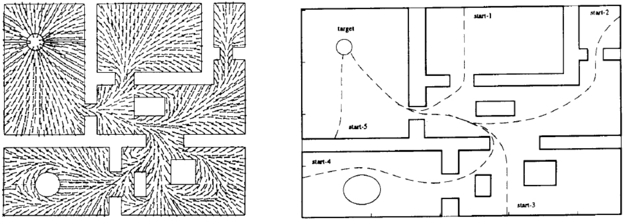


FIGURE: Example of navigation field

Navigation function and navigation field

The navigation field is given by : $-\nabla\phi$

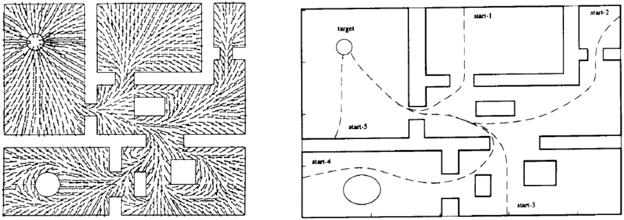


FIGURE: Example of navigation field

With these navigation fields, we can be sure that :

- any trajectory stays in the free space
- any trajectory reaching the minimum stays at this minimum

There is no guarantee on the speed and trajectories may not be smooth $\Rightarrow$ Bi-Harmonic Functions.

Mechanical stress field

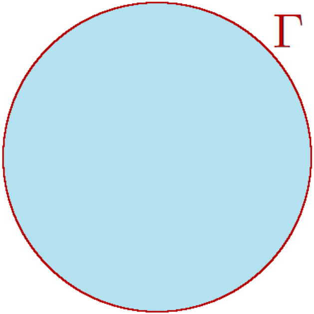


FIGURE: The mechanical stress field

Mechanical stress field

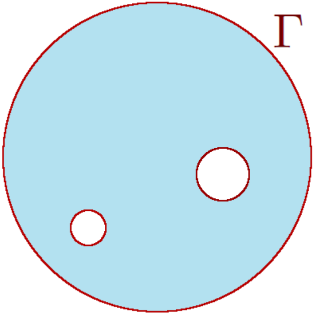


FIGURE: The mechanical stress field

Mechanical stress field

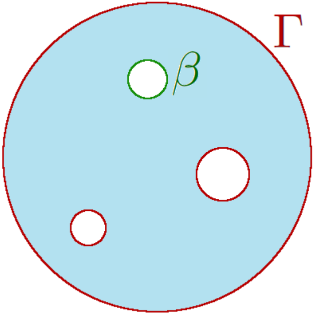


FIGURE: The mechanical stress field

Mechanical stress field

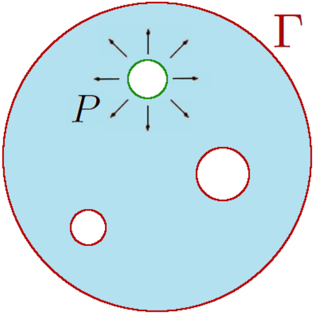


FIGURE: The mechanical stress field

Mechanical stress field

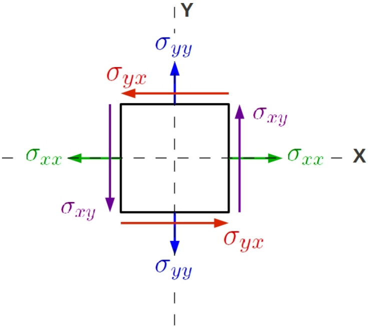


FIGURE: Stresses representation

Biharmonic functions : guideline

- Solve $\triangle^2 F = 0$ + boundary conditions
- Compute the stresses by :

$$\sigma_{xx} = \partial^2_{yy} F(x,y) \qquad \sigma_{yy} = \partial^2_{xx} F(x,y) \qquad \sigma_{xy} = -\partial^2_{xy} F(x,y)$$

$\Rightarrow$ Tensor field

- Compute the principal stresses($=$ eigenvalues)

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \Rightarrow \begin{bmatrix} \sigma_{min} & 0 \\ 0 & \sigma_{max} \end{bmatrix}$$

- Compute the eigenvectors corresponding to σ_{min}

$\Rightarrow$ Navigation field

Fields with obstacle

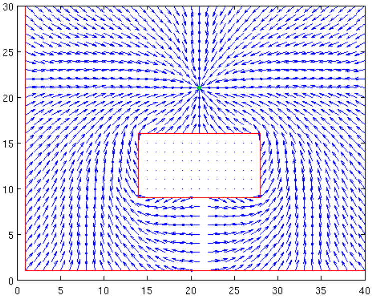


FIGURE: With one obstacle

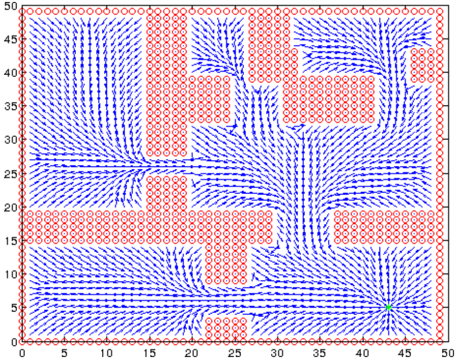


FIGURE: For a more complex geometry

Conclusions

Biharmonic Navigation Functions

- Ensure conflict free trajectory design
- With mathematical proof
- With speed range constraint
- With curvature constraint
- May be used in tactical phase

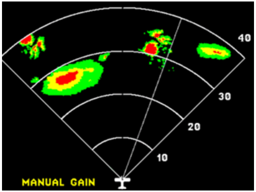
Have to be extended to the stochastic framework $\Rightarrow$ Stochastic Biharmonic Functions

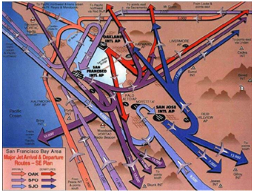
Agenda

- Some Trajectory Models
- Strategic Trajectory Design
- Pre-Tactical Trajectory Design
- Tactical Trajectory Design
- Emergency Trajectory Design

On-Board A/C Optimal Trajectory Generation

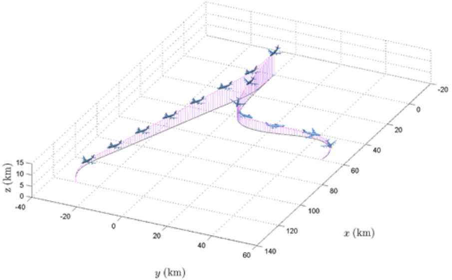
- Over 70% of fatal aviation accidents are in **take-off/landing phases**.
- Cockpit **emergency handling** from crew can result in completely different outcomes : Swissair Flight 111, US Airways Flight 1549
- Landing in mountainous terrain (e.g., LinZhi airport in China), avoiding inclement weather, or other aircraft in the area requires reliable **obstacle avoidance**.





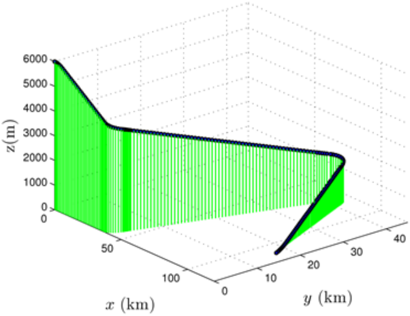
Aircraft Emergency Landing

- **Time** is the most critical factor
  - Swissair flight 111 : 14min
  - US Airways flight 1549 : 3min
- **Fuel** may be a limiting factor too
- **Challenges**
  - **Real-Time** requirement
  - **Convergence** guarantees



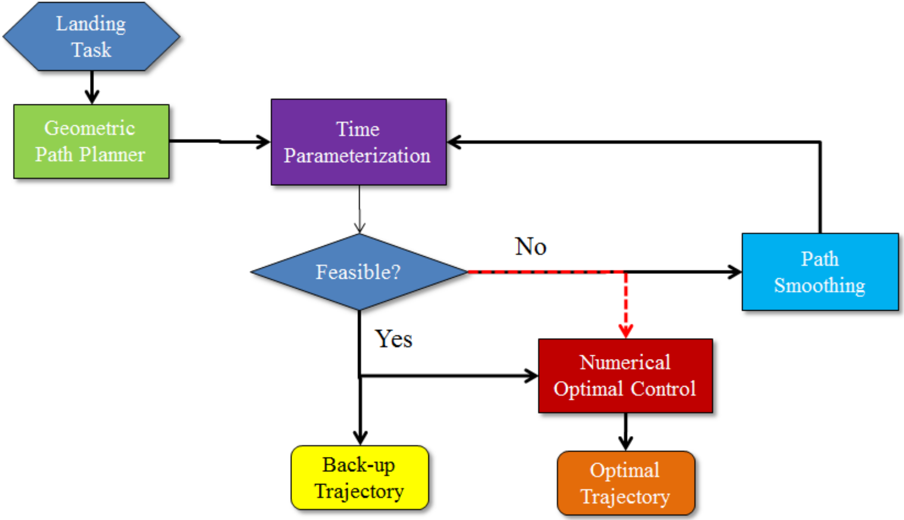
An Alternative

- Use a **hierarchical approach**
- Geometric planner
  - State constraints, obstacles
  - Path generator



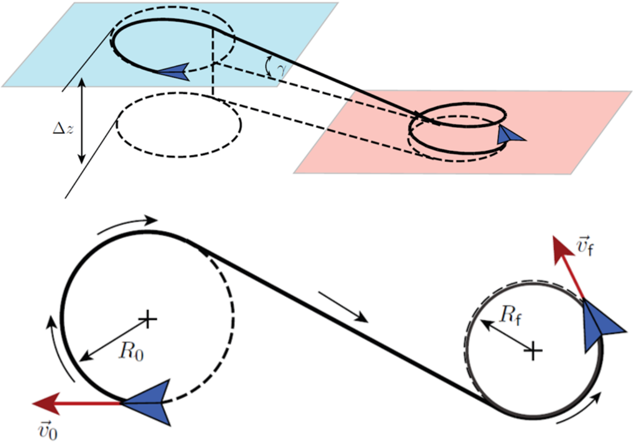
- Motion planner
  - Time parameterization
  - Trajectory generator
- **Key Idea :** First find flyable **path** to avoid obstacles ; then find a feasible **trajectory** to follow along this path.
- Requires the solution of **optimal time parameterization** (or **velocity generation**) problem.
- The latter is a **one-dimensional** optimal control problem that can be solved very **efficiently** !

On-Line Optimal Trajectory Generation Schematic



Initial Path Guess

Use Dubins paths with continuous descent



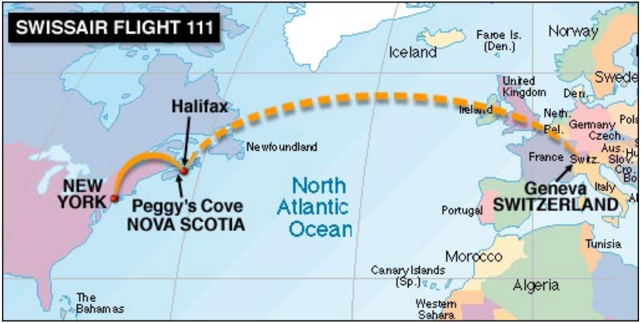
Application to Real Test Cases

Swissair 111

US Air 1549

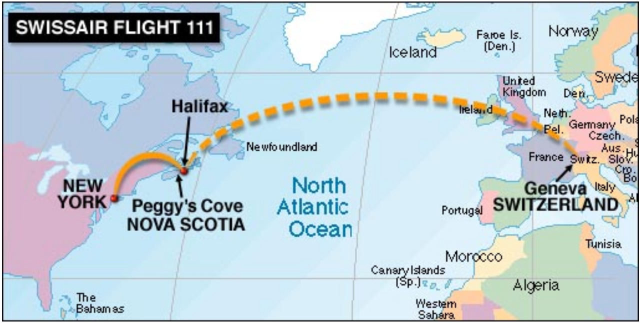
Test Case 1 : Swissair 111

- Swissair 111 (McDonnell Douglas MD-11) from JFK (NY) to Geneva (Switzerland).



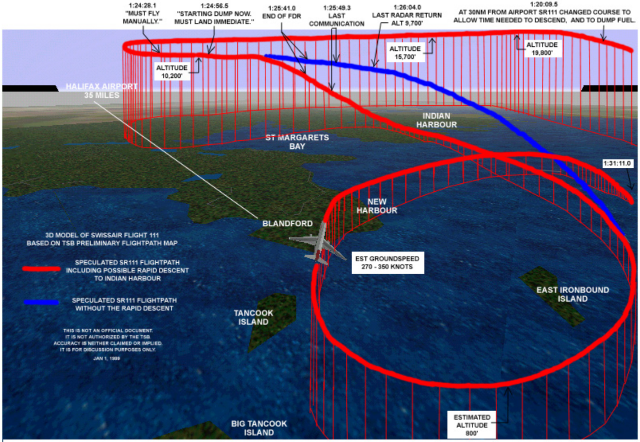
Test Case 1 : Swissair 111

- Swissair 111 (McDonnell Douglas MD-11) from JFK (NY) to Geneva (Switzerland).
- On Wednesday, 2 September 1998, the aircraft crashed into the Atlantic Ocean southwest of Halifax International Airport (due to fire on Board).

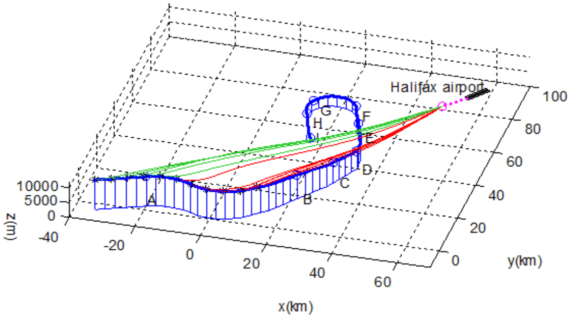


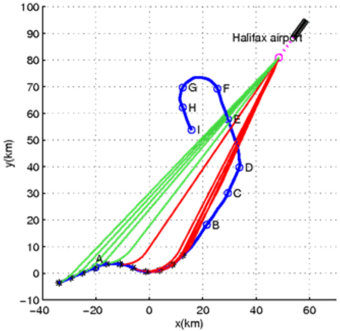
Test Case 1 : Swissair 111

Test Case 1 : Swissair 111



Test Case 1 : Swissair 111

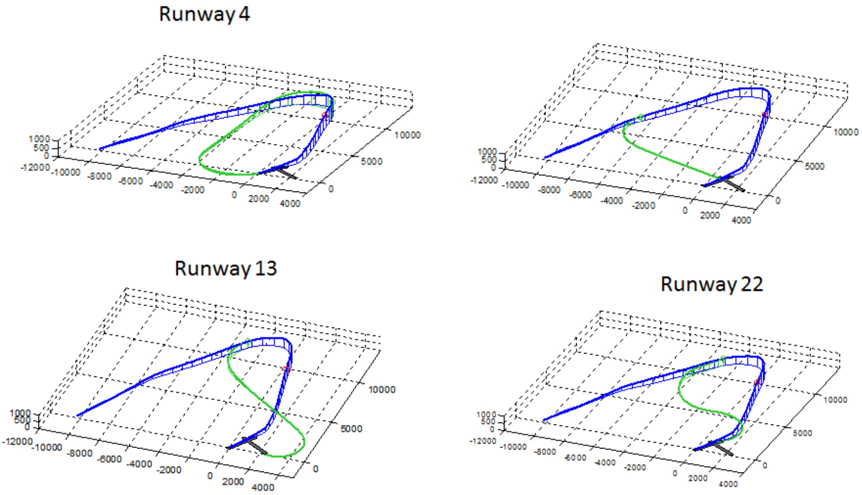




Test Case 2 : US Air 1549

VIDEO !

Test Case 2 : US Air 1549



QUESTIONS ?