A Variable Neighborhood Search approach for the 3-maneuver aircraft conflict resolution problem

A. Alonso-Ayuso
antonio.alonso@urjc.es

Rey Juan Carlos University

A. Alonso-Ayuso, L.F. Escudero, N. Mladenović

Outline

Outline

Problem Objective

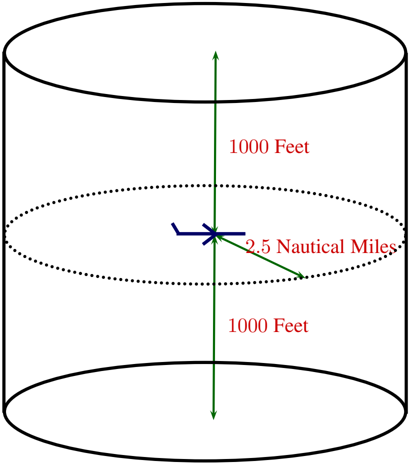
Given a set of flights configurations (waypoints, velocities, angles of motion, altitude level, etc.)

Problem objective

What control strategy should be followed by the pilots and the air traffic service provider to prevent the aircraft from coming too close to each other?

Conflict **definition:** It is an event in which two or more aircraft experience a loss of minimum separation.

Safety distances



How to avoid conflict situations?

In order to avoid conflict situations, an aircrat can perfom the following maneuvers

Types of maneuvers

- Velocity changes.
- Heading angle changes.
- Altitude changes.

Features of the “Velocity Changes” (VC) model I

Pallottino, Feron and Bicci (2002), “*Conflict resolution problems for air traffic management systems solved with mixed integer programming*”, **IEEE, Transactions on Intelligent Transportation Systems 3(1), 3–11**

These authors consider **two different MILO models**:

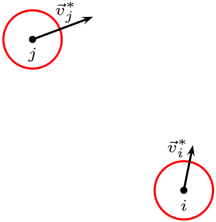
- Velocity (VC), and,
- Heading Angle Changes (HAC).

Features of the “Velocity Changes” (VC) model II

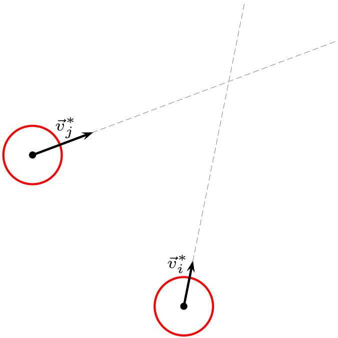
Features of the VC model

- It does not allow neither **altitude** nor **heading angle** changes causing several infeasible situations.
- It is incomplete due to specific **cases are not solved**.
- It is based on **geometric constructions**.

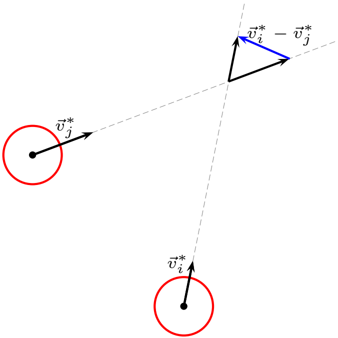
Geometric Construction I



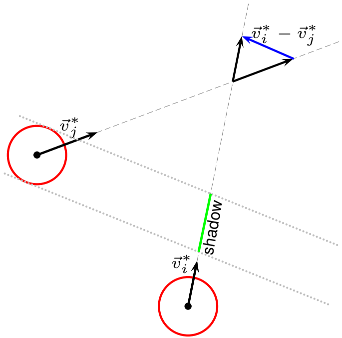
Geometric Construction I



Geometric Construction I

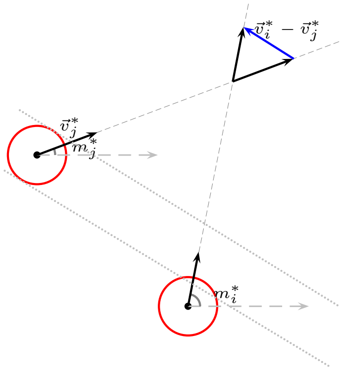


Geometric Construction I



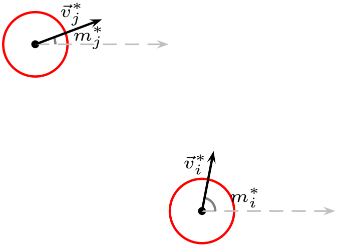
No conflict exists

Conflict Situation

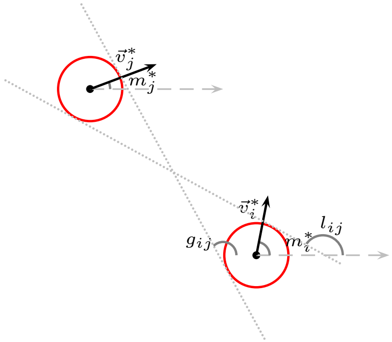


A conflict between aircraft i and j is expected

Geometric Construction II



Geometric Construction II



No conflict Constraints:

$$\frac{(v_i^* \quad)\sin(m_i^* \quad) - (v_j^* \quad)\sin(m_j^* \quad)}{(v_i^* \quad)\cos(m_i^* \quad) - (v_j^* \quad)\cos(m_j^* \quad)} \geqslant \tan(l_{ij})$$

or

$$\frac{(v_i^* \quad)\sin(m_i^* \quad) - (v_j^* \quad)\sin(m_j^* \quad)}{(v_i^* \quad)\cos(m_i^* \quad) - (v_j^* \quad)\cos(m_j^* \quad)} \leqslant \tan(g_{ij})$$

No conflict Constraints: Velocity changes

$$\frac{(v_i^* + \nu_i)\sin(m_i^* \qquad) - (v_j^* + \nu_j)\sin(m_j^* \qquad)}{(v_i^* + \nu_i)\cos(m_i^* \qquad) - (v_j^* + \nu_j)\cos(m_j^* \qquad)} \geqslant \tan(l_{ij})$$

or

$$\frac{(v_i^* + \nu_i)\sin(m_i^* \qquad) - (v_j^* + \nu_j)\sin(m_j^* \qquad)}{(v_i^* + \nu_i)\cos(m_i^* \qquad) - (v_j^* + \nu_j)\cos(m_j^* \qquad)} \leqslant \tan(g_{ij})$$

No conflict Constraints: Turn changes

$$\frac{(v_i^* \qquad) \sin(m_i^* + \mu_i) - (v_j^* \qquad) \sin(m_j^* + \mu_j)}{(v_i^* \qquad) \cos(m_i^* + \mu_i) - (v_j^* \qquad) \cos(m_j^* + \mu_j)} \geqslant \tan(l_{ij})$$

or

$$\frac{(v_i^* \qquad) \sin(m_i^* + \mu_i) - (v_j^* \qquad) \sin(m_j^* + \mu_j)}{(v_i^* \qquad) \cos(m_i^* + \mu_i) - (v_j^* \qquad) \cos(m_j^* + \mu_j)} \leqslant \tan(g_{ij})$$

No conflict Constraints: Velocity and Turn changes

$$\frac{(v_i^*+\nu_i)\sin(m_i^*+\mu_i) - (v_j^*+\nu_j)\sin(m_j^*+\mu_j)}{(v_i^*+\nu_i)\cos(m_i^*+\mu_i) - (v_j^*+\nu_j)\cos(m_j^*+\mu_j)} \geqslant \tan(l_{ij})$$

or

$$\frac{(v_i^*+\nu_i)\sin(m_i^*+\mu_i) - (v_j^*+\nu_j)\sin(m_j^*+\mu_j)}{(v_i^*+\nu_i)\cos(m_i^*+\mu_i) - (v_j^*+\nu_j)\cos(m_j^*+\mu_j)} \leqslant \tan(g_{ij})$$

No conflict Constraints: adding altitude changes

$$\frac{(v_i^* + \nu_i)\sin(m_i^* + \mu_i) - (v_j^* + \nu_j)\sin(m_j^* + \mu_j)}{(v_i^* + \nu_i)\cos(m_i^* + \mu_i) - (v_j^* + \nu_j)\cos(m_j^* + \mu_j)} \geqslant \tan(l_{ij})$$

or

$$\frac{(v_i^* + \nu_i)\sin(m_i^* + \mu_i) - (v_j^* + \nu_j)\sin(m_j^* + \mu_j)}{(v_i^* + \nu_i)\cos(m_i^* + \mu_i) - (v_j^* + \nu_j)\cos(m_j^* + \mu_j)} \leqslant \tan(g_{ij})$$

Altitude changes: a new 0-1 variable per flight is added for considering altitude changes.
These new variable *activates/deactivates* the no-conflict constraints.

Outline

1. Introduction
2. The Velocity, Turn and Altitude Changes model (VTAC)
   - Our previous approaches to solve the VTAC model
   - Variable Neighborhood Search for the TC and ideas for VTAC
   - Multi-objective criteria
3. Computational experience
   - Illustrative instance
   - Small-size instances
   - Real-size instances
4. Conclusions and future research

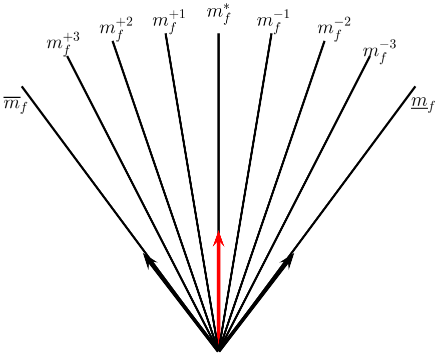
MINO: Exact scheme (Minotaur)

Features

- The conflicts are **detected** as well as **solved**.
- A **geometric construction** is used.
- A **mixed 0–1 nonconvex nonlinear model** is proposed.
- The more the aircraft to consider, the larger the time to solve the model.

SMILO: Sequential Mixed Integer Linear Optimization

For a specific turn change, the problem is linear. Different angles can be considered by using binary variables, β



$$(v_f + \nu_f^+ - \nu_f^-)\sin(m_f + \mu_f^+ - \mu_f^-)$$

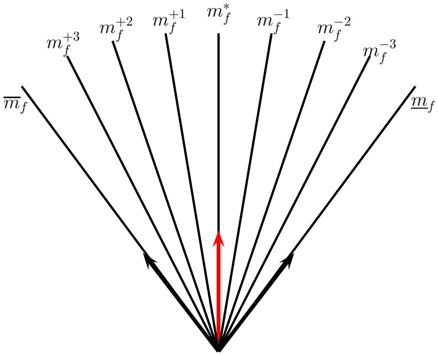
is replaced by

$$(v_f + \nu_f^+ - \nu_f^-)\left(\sum_{k\in\mathcal{K}} \sin(m_f^k)\beta_f^k\right),$$

where only one β-variable will be equal to 1.

SMILO: Sequential Mixed Integer Linear Optimization

For a specific turn change, the problem is linear. Different angles can be considered by using binary variables, β



The quadratic term $\nu_f \beta_f^k$ is linearized using the Fortet constraints:

$$
\begin{aligned}
x &\leqslant \nu \\
x &\leqslant M\beta \\
\nu - x &\leqslant M(1-\beta)
\end{aligned}
$$

where $x \equiv \nu \cdot \beta$

Algorithm for the SMILO approach

SMILO algorithm

1. Start the algorithm taken into account the initial angle bounds for each aircraft f.
2. Solve the MILO model.
3. Take the solution obtained and discretize the angle again with smaller rank amplitude.
4. Go to step 1 except if the improvement in the new solution is less than 1% with respect to the one obtained before.

Unconstrained problem: Penalty cost function I

The infeasibility condition when there is no null denominator is:

$$\tan(g_{ij}) \leqslant \frac{v_i \sin(m_i + \mu_i) - v_j \sin(m_j + \mu_j)}{v_i \cos(m_i + \mu_i) - v_j \cos(m_j + \mu_j)} \leqslant \tan(l_{ij})$$

Unconstrained problem: Penalty cost function I

The infeasibility condition when there is no null denominator is:

$$\tan(g_{ij}) \leqslant \frac{v_i \sin(m_i + \mu_i) - v_j \sin(m_j + \mu_j)}{v_i \cos(m_i + \mu_i) - v_j \cos(m_j + \mu_j)} \leqslant \tan(l_{ij})$$

whereas when there is a null denominator is:

$$-\cot(g_{ij}) \leqslant \frac{v_i \sin(m_i + \mu_i + \pi/2) - v_j \sin(m_j + \mu_j + \pi/2)}{v_i \cos(m_i + \mu_i + \pi/2) - v_j \cos(m_j + \mu_j + \pi/2)} \leqslant -\cot(l_{ij})$$

Unconstrained problem: Penalty cost function I

The infeasibility condition when there is no null denominator is:

$$\tan(g_{ij}) \leqslant \frac{v_i \sin(m_i + \mu_i) - v_j \sin(m_j + \mu_j)}{v_i \cos(m_i + \mu_i) - v_j \cos(m_j + \mu_j)} \leqslant \tan(l_{ij})$$

whereas when there is a null denominator is:

$$-\cot(g_{ij}) \leqslant \frac{v_i \sin(m_i + \mu_i + \pi/2) - v_j \sin(m_j + \mu_j + \pi/2)}{v_i \cos(m_i + \mu_i + \pi/2) - v_j \cos(m_j + \mu_j + \pi/2)} \leqslant -\cot(l_{ij})$$

And, the objective function is:

$$\min \sum_{f \in \mathcal{F}} |\mu_f|$$

Unconstrained problem: Penalty cost function I

The infeasibility condition when there is no null denominator is:

$$\tan(g_{ij}) \leqslant \frac{(v_i+\nu_i)\sin(m_i+\mu_i)-(v_j+\nu_j)\sin(m_j+\mu_j)}{(v_i+\nu_i)\cos(m_i+\mu_i)-(v_j+\nu_j)\cos(m_j+\mu_j)} \leqslant \tan(l_{ij})$$

whereas when there is a null denominator is:

$$-\cot(g_{ij}) \leqslant \frac{(v_i+\nu_i)\sin(m_i+\mu_i+\pi/2)-(v_j+\nu_j)\sin(m_j+\mu_j+\pi/2)}{(v_i+\nu_i)\cos(m_i+\mu_i+\pi/2)-(v_j+\nu_j)\cos(m_j+\mu_j+\pi/2)} \leqslant -\cot(l_{ij})$$

And, the objective functions are:

$$\min \sum_{f\in\mathcal{F}} |\nu_f| \quad \min \sum_{f\in\mathcal{F}} |\mu_f| \quad \min \sum_{f\in\mathcal{F}} |\gamma_f|$$

Unconstrained problem: Penalty cost function II

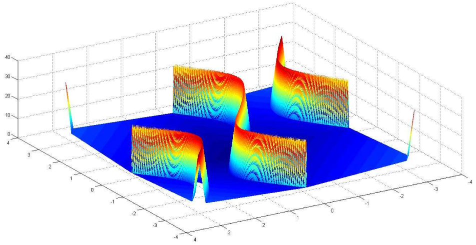
So, the penalty cost function is composed of the objective function and the following one:

$$g(\mu) = \begin{cases} \sum_{i<j\in\mathcal{F}} \max\left\{0, \min\{\tan(l_{ij}) - t_{ij}, t_{ij} - \tan(g_{ij})\}\right\} & \text{if } cp_{ij} = 0 \\ \sum_{i<j\in\mathcal{F}} \max\left\{0, \min\{-\cot(l_{ij}) - t'_{ij}, t'_{ij} + \cot(g_{ij})\}\right\} & \text{if } cp_{ij} = 1 \end{cases}$$

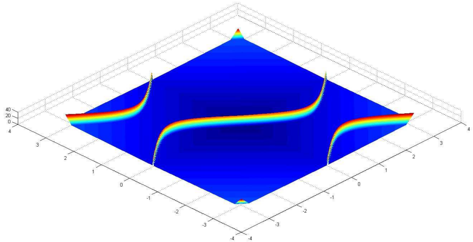
Penalty cost function

$$f(\mu) = \min\left\{Mg(\mu) + \sum_{f\in\mathcal{F}} |\mu_f|\right\}$$

Penalty cost function



Penalty cost function



Local Search I

The main features of the local search are:

- It is based on first improvement instead of best improvement to obtain a feasible solution as soon as possible. We have tried with best improvement but the solution quality was not different.
- When a solution is improved, the parameters of the problem must be updated.

Local Search I

The main features of the local search are:

- It is based on first improvement instead of best improvement to obtain a feasible solution as soon as possible. We have tried with best improvement but the solution quality was not different.
- Each aircraft changes its angle of motion ang and $-ang$ radians until no solution improvement
- When a solution is improved, the parameters of the problem must be updated.

Local Search I

The main features of the local search are:

- It is based on first improvement instead of best improvement to obtain a feasible solution as soon as possible. We have tried with best improvement but the solution quality was not different.
- Each aircraft changes its angle of motion ang and $-ang$ radians until no solution improvement; its velocity vel and $-vel$ units; and its altitude level alt and $-alt$ levels.
- When a solution is improved, the parameters of the problem must be updated.

Local Search II

Algorithm 2: First improvement local search for the CDR problem

Function `FirstImprovement`$(\mu, ang, A, V, M, CP, TL, TG, CTL, CTG)$;
1 $j = 0$;
repeat
2   Move aircraft j by ang degrees to get new μ;
    if $f(\mu) < f(\mu')$ **then**
        $j = 0$; $\mu' = \mu$;
    else
        Move aircraft j by $-ang$ degrees to get new μ;
        if $f(\mu) < f(\mu')$ **then**
            $j = 0$; $\mu' = \mu$;
        else
            $j = j + 1$;
        end
    end
until $j > n$;
3 `Updating`$(j, A, V, M, CP, TL, TG, CTL, CTG)$

Shaking I

The shaking procedure consists of

- Parameter k determines both, number of aircraft to consider and amount of radians to turn one aircraft
- $n/4$ aircraft are candidates to change their angle of motion at most
- Randomly, the sign of angle movement is selected

Shaking I

The shaking procedure consists of

- Parameter k determines both, number of aircraft to consider and amount of radians to turn one aircraft; velocity and altitude must be added
- $n/4$ aircraft are candidates to change their angle of motion at most; and the rest of maneuvers
- Randomly, the sign of angle movement is selected; and the sign of velocity and altitude variations

Shaking II

Algorithm 3: Shaking for the CDR problem

```
Function Shaking(μ, ang, A, V, M, CP, TL, TG, CTL, CTG);
nn ← k  mod n/4 ;
u1 = Rand(0, 1); u2 = Rand(0, 1); ang ← u1 k;
j = 0;
repeat
    if u2 < 0.5 then
        Move aircraft j by ang degrees;
    else
        Move aircraft j by −ang degrees;
    end
    Updating(j, A, V, M, CP, TL, TG, CTL, CTG);
    j ← j + 1;
until j = nn;
```

Basic VNS algorithm

Algorithm 4: Steps of the VNS for the CDR problem

Function `VNS` $(x, k_{max}, t_r, t_{max})$;
Calculate CP, TL, TG, CTL, CTG, A ;
`FirstImprovement`$(\mu', ang, A, V, M, CP, TL, TG, CTL, CTG)$;
repeat
    $k \leftarrow 1$;
    repeat
        $x' \leftarrow$ `Shake`(x, k) `/* Shaking */`;
        $x'' \leftarrow$ `FirstImprovement`$(\mu', ang, A, V, M, CP, TL, TG, CTL, CTG)$;
        if $f(x'') < f(x)$ **then**
            $x \leftarrow x'$; $k \leftarrow 1$; $t_{li} \leftarrow$ `CpuTime()` `/* Make a move */`;
        else
            $k \leftarrow k + 1$ `/* Next neighborhood */` ;
        end
        $t \leftarrow$ `CpuTime()`;
    until $k = k_{max}$;
    if $t - t_{li} > t_r$ **then**
        break;
    end
until $t > t_{max}$;

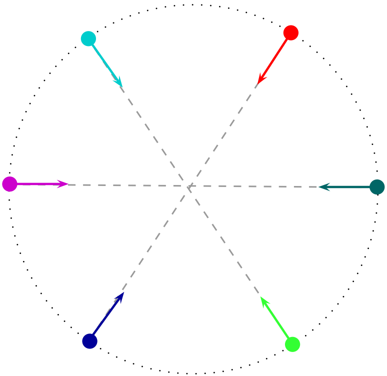
Multi-objective approaches used

- Lexicographical
- Compromise
- Mixture of minimizing largest deviation and compromise criterion

All of them need the *pay-off matrix*

Outline

Initial situation



Computational experience for illustrative instance

The solver that has been used for the MINLO TC model is: Minotaur

	Minotaur		SILO		VNS		GAP (Minotaur)	
Inst	z	t	z	t	z	t	SILO	VNS
I2	0.2507	0.02	0.2659	0.05	0.2507	0.21	6.06	0.00
I3	0.4345	0.03	0.4416	0.11	0.4345	1.86	1.63	0.00
I4	0.7108	0.11	0.7140	0.08	0.7108	0.10	0.45	0.00
I5	1.0715	1.23	1.0944	0.28	1.0715	1.83	2.14	0.00
I6	1.5161	9.32	1.5772	0.29	1.5161	0.98	4.03	0.00
I7	2.0457	143.66	2.1341	1.25	2.0462	1.44	4.32	0.02
I8	2.6620	567.92	2.7123	4.45	2.6620	1.45	1.89	0.00
I9	3.3673	4332.67	3.4338	15.50	3.3697	4.77	1.97	0.07

4xIntel Core i5-2430M, 2.40 GHz, 4 GB RAM and Linux Xubuntu 11.10 OS

Computational experience for small-size instances

	Best known	Minotaur		SILO		VNS		GAP (Best known)		
Case	z	z	t	z	t	z	t	Minot	SILO	VNS
C05	0.1670	0.1690	0.16	0.1743	0.09	0.1671	1.56	0.16	6.52	0.00
C07	0.2254	0.2255	1.67	0.2317	0.19	0.2255	2.68	0.00	5.74	0.05
C10	0.3822	0.3832	342.27	0.4029	0.51	0.3824	4.78	0.15	6.80	0.08

Computational experience for real-size instances

	SILO		VNS		GAP (Best known)	
Case	z	t	z	t	SILO	VNS
C12	0.6990	0.71	0.6877	6.76	3.06	1.31
C15	1.2033	2.18	1.1543	8.23	5.62	0.67
C20	2.1118	4.11	2.0480	12.46	3.34	2.04
C25	3.1776	256.94	3.0309	13.27	6.90	2.53

Outline

Conclusions and current work

- A Variable Neighborhood Search algorithm has been presented.
- Conflict situations are solved by performing only turn changes.
- The local search and shaking phases are based on angle discretization.
- The computational experience shows that VNS improves the solutions obtained for the TC model.

Future research

- Extending the model by including velocity and altitude maneuvers.
- If other maneuvers are considered, performing a ranked multiobjective optimization.
- Improving the local search as well as the shaking phase of the algorithm.

Thanks a lot for your attention!

¡Muchas gracias por su atención!

A Variable Neighborhood Search approach for the 3-maneuver aircraft conflict resolution problem

A. Alonso-Ayuso
antonio.alonso@urjc.es

Rey Juan Carlos University

A. Alonso-Ayuso, L.F. Escudero, N. Mladenović