A Variable Neighborhood Search approach for the aircraft conflict resolution problem

F. Javier Martín Campo
javier.martin.campo@mat.ucm.es

Joint work with:
A. Alonso-Ayuso, L.F. Escudero, N. Mladenović
Outline

Introduction

Variable Neighborhood Search (VNS)

The Velocity, Turn and Altitude Changes model (VTAC)

Multi-objective methods

Computational experience

Conclusions and future research
Introduction

Variable Neighborhood Search (VNS)

The Velocity, Turn and Altitude Changes model (VTAC)

Multi-objective methods

Computational experience

Conclusions and future research
Motivation

Figure 1. Total scheduled traffic
(revenue passenger-kilometres performed, 2005-2014)
Hypothesis

- An air sector is considered.
- Enroute phase.
- Short term (up to 5 minutes).
- Static.

A. Lau, J. Berling, F. Linke, V. Gollnick, K. Nachtigall. Large-Scale Network Slot Allocation with Dynamic Time Horizons
Given a set of flight configurations (waypoints, velocities, angles of motion, altitude level, etc.)
Problem Objective

Given a set of flight 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
Safety distances

1000 Feet

2.5 Nautical Miles

1000 Feet
Safety distances

1000 Feet

2.5 Nautical Miles

1000 Feet
In order to avoid conflict situations, maneuvers that an aircraft can perform:
How to avoid conflict situations?

In order to avoid conflict situations, maneuvers that an aircraft can perform:

Types of maneuvers

- Horizontal:
  - **Velocity** changes.
  - **Heading angle** changes.
In order to avoid conflict situations, maneuvers that an aircraft can perform:

### Types of maneuvers

- **Horizontal:**
  - Velocity changes.
  - Heading angle changes.

- **Vertical:**
  - Altitude changes.
Literature based on Mathematical Optimization

Authors who have worked on the topic with mathematical optimization (among others):

- N. Durand, G. Granger and S. Cafieri.
- L. Pallottino, E. Feron and A. Bicchi.
- M.A. Christodoulou and C. Costoulakis.
- J. Omer, J. Farges and T. Lehouillier.
- M. Soler, M. Kamgarpour, J. Lloret and J. Lygeros.
- C. Peyronne, A.R. Conn, M. Mongeau and D. Delahaye.
- A. Vela and S. Solak.
Outline

Introduction

Variable Neighborhood Search (VNS)

The Velocity, Turn and Altitude Changes model (VTAC)

Multi-objective methods

Computational experience

Conclusions and future research
Principles of VNS

- It has been successfully applied to many combinatorial optimization problems among others.
Principles of VNS

- It has been successfully applied to many combinatorial optimization problems among others.

Facts

- A local minimum with respect to one neighborhood structure is not necessary so for another.
- A global minimum is a local minimum with respect to all possible neighborhood structures.
- For many problems local minima with respect to one or several neighborhoods are relatively close to each other.
Basic idea: Escape from local optima trap by changing the neighborhood structure.

Basic VNS

- Select a set of neighborhoods
  \[ N = \{ N_1, \ldots, N_{k_{\text{max}}} \} \]
- Do while stop
  \[ k = 1 \]
  Do while \( k < k_{\text{max}} \)
  - Shaking
  - Local search
  - Move or not

Only global opt. is local for any \( N_k \)
Outline

Introduction

Variable Neighborhood Search (VNS)

The Velocity, Turn and Altitude Changes model (VTAC)

Multi-objective methods

Computational experience

Conclusions and future research
The Velocity Changes (VC) model


Two different MILO models are studied: Velocity (VC) and Heading Angle Changes (HAC).
The Velocity Changes (VC) model


Two different MILO models are studied: Velocity (VC) and Heading Angle Changes (HAC).

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 that are not solved.
- It is based on geometric constructions.
Geometric Construction I
Geometric Construction I

\[ \vec{v}_{j} \times m_{j} \]

\[ \vec{v}_{i} \times m_{i} \]
Geometric Construction I

\[ \vec{v}_i^* \]

\[ \vec{v}_j^* \]
Geometric Construction I
Geometric Construction I

$\vec{v}_i - \vec{v}_j$

Aircraft Conflict Resolution, Philadelphia 2016 VTAC model
Geometric Construction I

\[ \vec{v}_i^* - \vec{v}_j^* \]

Aircraft Conflict Resolution, Philadelphia 2016
VTAC model
Geometric Construction I

\[ \vec{v}_i^* - \vec{v}_j^* \]

Aircraft Conflict Resolution, Philadelphia 2016 VTAC model
Conflict Situation

Aircraft Conflict Resolution, Philadelphia 2016

VTAC model
Geometric Construction II
Geometric Construction II

\[ \omega_{ij} \]

\[ d_{ij} \]
Geometric Construction II

Aircraft Conflict Resolution, Philadelphia 2016 VTAC model
No conflict Constraints

No conflict constraint (aircraft at same altitude level)

\[
\frac{(v_i^* \sin(m_i^*) - v_j^* \sin(m_j^*))}{(v_i^* \cos(m_i^*) - v_j^* \cos(m_j^*))} \geq \tan(l_{ij})
\]

or

\[
\frac{(v_i^* \sin(m_i^*) - v_j^* \sin(m_j^*))}{(v_i^* \cos(m_i^*) - v_j^* \cos(m_j^*))} \leq \tan(g_{ij})
\]
No conflict Constraints

No conflict constraint (aircraft at same altitude level)

\[
\frac{(v_i^* + \nu_i) \sin(m_i^*) - (v_j^* + \nu_j) \sin(m_j^*)}{(v_i^* + \nu_i) \cos(m_i^*) - (v_j^* + \nu_j) \cos(m_j^*)} \geq \tan(l_{ij})
\]

or

\[
\frac{(v_i^* + \nu_i) \sin(m_i^*) - (v_j^* + \nu_j) \sin(m_j^*)}{(v_i^* + \nu_i) \cos(m_i^*) - (v_j^* + \nu_j) \cos(m_j^*)} \leq \tan(g_{ij})
\]
No conflict Constraints

No conflict constraint (aircraft at same altitude level)

\[
\frac{(v_i^*) \sin(m_i^* + \mu_i) - (v_j^*) \sin(m_j^* + \mu_j)}{(v_i^*) \cos(m_i^* + \mu_i) - (v_j^*) \cos(m_j^* + \mu_j)} \geq \tan(l_{ij})
\]

or

\[
\frac{(v_i^*) \sin(m_i^* + \mu_i) - (v_j^*) \sin(m_j^* + \mu_j)}{(v_i^*) \cos(m_i^* + \mu_i) - (v_j^*) \cos(m_j^* + \mu_j)} \leq \tan(g_{ij})
\]
**No conflict Constraints**

<table>
<thead>
<tr>
<th>No conflict constraint (aircraft at same altitude level)</th>
</tr>
</thead>
</table>
| \[
\frac{(v_i^* + v_i) \sin(m_i^* + \mu_i) - (v_j^* + v_j) \sin(m_j^* + \mu_j)}{(v_i^* + v_i) \cos(m_i^* + \mu_i) - (v_j^* + v_j) \cos(m_j^* + \mu_j)} \geq \tan(l_{ij})
\]

or

| \[
\frac{(v_i^* + v_i) \sin(m_i^* + \mu_i) - (v_j^* + v_j) \sin(m_j^* + \mu_j)}{(v_i^* + v_i) \cos(m_i^* + \mu_i) - (v_j^* + v_j) \cos(m_j^* + \mu_j)} \leq \tan(g_{ij})
\] |
The infeasibility condition for a pair of aircraft flying at the same altitude level when there is no null denominator is:

\[
\tan(g_{ij}) \leq \frac{(v_{i} + \nu_{i}) \sin(m_{i} + \mu_{i}) - (v_{j} + \nu_{j}) \sin(m_{j} + \mu_{j})}{(v_{i} + \nu_{i}) \cos(m_{i} + \mu_{i}) - (v_{j} + \nu_{j}) \cos(m_{j} + \mu_{j})} \leq \tan(l_{ij})
\]
The infeasibility condition for a pair of aircraft flying at the same altitude level when there is no null denominator is:

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

whereas when there is a null denominator is:

\[
-\cot(g_{ij}) \leq \frac{(v_i + \nu_i) \sin(m_i + \mu_i + \pi/2) - (v_j + \nu_j) \sin(m_j + \mu_j + \pi/2)}{(v_i + \nu_i) \cos(m_i + \mu_i + \pi/2) - (v_j + \nu_j) \cos(m_j + \mu_j + \pi/2)} \leq -\cot(l_{ij})
\]
Unconstrained problem: Penalty cost function I

The infeasibility condition for a pair of aircraft flying at the same altitude level when there is no null denominator is:

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

whereas when there is a null denominator is:

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

And, the objective functions are:

\[ \min_{f \in F} \sum_{f \in F} |\nu_f| \quad \min_{f \in F} \sum_{f \in F} |\mu_f| \quad \min_{f \in F} \sum_{f \in F} |\gamma_f| \]
So, the penalty cost function is composed of the objective function and the following one (infeasibility condition):

\[
g(\nu, \mu, \gamma) = \begin{cases} 
  \sum_{i < j \in F} \max \{0, \min \{\tan(l_{ij}) - t_{ij}, t_{ij} - \tan(g_{ij})\}\} & \text{if } cp_{ij} = 0 \text{ and } \\
  \sum_{i < j \in F} \max \{0, \min \{-\cot(l_{ij}) - t'_{ij}, t'_{ij} + \cot(g_{ij})\}\} & \text{if } cp_{ij} = 1 \text{ and } \\
  z_i + \gamma_i = z_j + \gamma_j & 
\end{cases}
\]

\[
\text{if } cp_{ij} = 0 \text{ and } \\
\text{if } cp_{ij} = 1 \text{ and }
\]

\[
z_i + \gamma_i = z_j + \gamma_j
\]
Unconstrained problem: Penalty cost function II

So, the penalty cost function is composed of the objective function and the following one (infeasibility condition):

\[
g(\nu, \mu, \gamma) = \begin{cases} 
\sum_{i<j\in F} \max \{0, \min\{\tan(l_{ij}) - t_{ij}, t_{ij} - \tan(g_{ij})\}\} & \text{if } cp_{ij} = 0 \text{ and } z_i + \gamma_i = z_j + \gamma_j \\
\sum_{i<j\in F} \max \{0, \min\{-\cot(l_{ij}) - t'_{ij}, t'_{ij} + \cot(g_{ij})\}\} & \text{if } cp_{ij} = 1 \text{ and } z_i + \gamma_i = z_j + \gamma_j 
\end{cases}
\]

where:

\[
t_{ij} = \frac{(v_i + \nu_i) \sin(m_i + \mu_i) - (v_j + \nu_j) \sin(m_j + \mu_j)}{(v_i + \nu_i) \cos(m_i + \mu_i) - (v_j + \nu_j) \cos(m_j + \mu_j)}
\]

\[
t'_{ij} = \frac{(v_i + \nu_i) \sin(m_i + \mu_i + \pi/2) - (v_j + \nu_j) \sin(m_j + \mu_j + \pi/2)}{(v_i + \nu_i) \cos(m_i + \mu_i + \pi/2) - (v_j + \nu_j) \cos(m_j + \mu_j + \pi/2)}
\]
Unconstrained problem: Penalty cost function II

So, the penalty cost function is composed of the objective function and the following one (infeasibility condition):

\[
g(\nu, \mu, \gamma) = \begin{cases} 
\sum_{i<j \in F} \max \{0, \min\{\tan(l_{ij}) - t_{ij}, t_{ij} - \tan(g_{ij})\}\} & \text{if } cp_{ij} = 0 \text{ and } z_i + \gamma_i = z_j + \gamma_j \\
\sum_{i<j \in F} \max \{0, \min\{-\cot(l_{ij}) - t'_{ij}, t'_{ij} + \cot(g_{ij})\}\} & \text{if } cp_{ij} = 1 \text{ and } z_i + \gamma_i = z_j + \gamma_j 
\end{cases}
\]

where:

\[
t_{ij} = \frac{(v_i + \nu_i) \sin(m_i + \mu_i) - (v_j + \nu_j) \sin(m_j + \mu_j)}{(v_i + \nu_i) \cos(m_i + \mu_i) - (v_j + \nu_j) \cos(m_j + \mu_j)}
\]

\[
t'_{ij} = \frac{(v_i + \nu_i) \sin(m_i + \mu_i + \pi/2) - (v_j + \nu_j) \sin(m_j + \mu_j + \pi/2)}{(v_i + \nu_i) \cos(m_i + \mu_i + \pi/2) - (v_j + \nu_j) \cos(m_j + \mu_j + \pi/2)}
\]

Penalty cost function

\[
f(\nu, \mu, \gamma) = \min\{Mg(\nu, \mu, \gamma) + \text{objective function}\}
\]
Penalty cost function
The following data structure is used in our VNS approach:

1. Aircraft configuration matrix (for each aircraft: velocity, heading angle, altitude level, abscissa, ordinate, maximum and minimum maneuvers allowed).

2. Solution matrix (for each aircraft: velocity, heading angle and altitude level).

3. Auxiliar matrices:
   - Parameter $c_{pij}$.
   - $TL$, $TG$, $CTL$ and $CTG$.
   - $A$ (penalty function).

Aircraft Conflict Resolution, Philadelphia 2016 VTAC model
Algorithm 1: Updating matrix A

Function Updating(j, A, ν, μ, γ, V, T, Z, CP, TL, TG, CTL, CTG);

i = 0;

repeat
    if CP(i, j) = 0 and Z_i + γ_i = Z_j + γ_j then
        Use T, TL and TG to calculate A_ij
        by using g;
    else
        if CP(i, j) = 1 and Z_i + γ_i = Z_j + γ_j then
            Use T + π/2, CTL and CTG to
calculate A_ij by using g;
        else
            A_ij = 0;
        end
    end
    i = i + 1;
until i = n;
The main features of the local search (intensification phase) 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 $\text{ang}$ and $-\text{ang}$ radians; its velocity $\text{vel}$ and $-\text{vel}$ nm/h; its altitude level $\text{alt}$ and $-\text{alt}$ until no solution improvement.

- When a solution is improved, the parameters of the problem must be updated.
Algorithm 2: First improvement local search for the CDR problem

Function FirstImprovement($\nu, \mu, \gamma, vel, ang, alt, A, V, T, Z, CP, TL, TG, CTL, CTG$);

$k = 1$;
repeat
  $j = 1$;
  repeat
    Move aircraft $j$ by $vel$ nm/h if $k = 1$;
    Move aircraft $j$ by $ang$ rads. if $k = 2$;
    Move aircraft $j$ by $alt$ levels if $k = 3$;
    if $f(x) < f(x')$ then
      $j = 0$; $x' = x$;
    else
      Move aircraft $j$ by $-vel$ nm/h if $k = 1$;
      Move aircraft $j$ by $-ang$ rads. if $k = 2$;
      Move aircraft $j$ by $-alt$ levels if $k = 3$;
      if $f(x) < f(x')$ then
        $j = 0$; $x' = x$;
      else
        $j = j + 1$;
    end
  until $j > n$;
until $k > 3$;
Updating($j, A, \nu, \mu, \gamma, V, T, Z, CP, TL, TG, CTL, CTG$)
The shaking procedure (**diversification** phase) consists of

- Parameter $k$ determines both, number of aircraft to consider and corresponding maneuver to modify.
- $n/4$ aircraft are candidates to change any maneuver.
- Randomly, the sign of the corresponding maneuver is chosen.
Algorithm 3: Shaking for the CDR problem

Function Shaking($\nu, \mu, \gamma, \text{vel}, \text{ang}, \text{alt}, A, V, M, CP, TL, TG, CTL, CTG$);

$nn \leftarrow k \mod n/4$ ;
$u_1 = \text{Rand}(0,1); u_2 = \text{Rand}(0,1)$ ;
$u_3 = \lceil 3 \cdot \text{Rand}(0,1) \rceil$ ;
$\text{vel} \leftarrow u_1 k, \text{ang} \leftarrow u_1 k, \text{alt} \leftarrow \lceil u_1 k \rceil$ ;
$j = 0$;

repeat
    if $u_2 < 0.5$ then
        Move aircraft $j$ by $\text{vel}$ nm/h if $u_3 = 1$;
        Move aircraft $j$ by $\text{ang}$ rads. if $u_3 = 2$;
        Move aircraft $j$ by $\text{alt}$ levels if $u_3 = 3$;
    else
        Move aircraft $j$ by $-\text{vel}$ nm/h if $u_3 = 1$;
        Move aircraft $j$ by $-\text{ang}$ rads. if $u_3 = 2$;
        Move aircraft $j$ by $-\text{alt}$ levels if $u_3 = 3$;
    end

    Updating($j, A, \nu, \mu, \gamma, V, T, Z, CP, TL, TG, CTL, CTG$);

    $j \leftarrow j + 1$;

until $j = nn$;
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\((\nu, \mu, \gamma, vel, ang, alt, A, V, T, Z, CP, TL, TG, CTL, CTG)\);
repeat
    \(k \leftarrow 1\);
    repeat
        \(x' \leftarrow \text{Shake}(x, k)\) /* Shaking */;
        \(x'' \leftarrow \text{FirstImprovement}(\nu', \mu', \gamma', vel, ang,
            alt, A, V, T, Z, CP, TL, TG, CTL, CTG)\);
        if \(f(x'') < f(x)\) then
            \(x \leftarrow x'\); \(k \leftarrow 1\) /* Make a move */;
            \(t_{li} \leftarrow \text{CpuTime}()\);
        else
            \(k \leftarrow k + 1\) /* Next neighborhood */;
        end
    until \(k = k_{max}\);
    if \(t - t_{li} > t_r\) then
        break;
    end
until \(t > t_{max}\);
Outline

Introduction

Variable Neighborhood Search (VNS)

The Velocity, Turn and Altitude Changes model (VTAC)

Multi-objective methods

Computational experience

Conclusions and future research
Multi-objective approaches used

- Lexicographic Goal Programming.
- Compromise Programming.
- Compromise Programming combining $l_1$ and $l_\infty$ distances.
Multi-objective approaches used

- Lexicographic Goal Programming.
- Compromise Programming.
- Compromise Programming combining $l_1$ and $l_\infty$ distances.

All of them need the *pay-off matrix*. 
An **ideal** value for a single objective function is the best possible value when that objective is optimized subject to the corresponding set of constraints.
Pay-off matrix

<table>
<thead>
<tr>
<th>Ideal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>An <strong>ideal</strong> value for a single objective function is the best possible value when that objective is optimized subject to the corresponding set of constraints.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-ideal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A <strong>nonideal</strong> value is the worst value for a single objective function when optimizing another objective.</td>
</tr>
</tbody>
</table>

- It is a $n \times n$ where $n$ is the number of objective functions.
- In the diagonal the **ideal** values are presented.
- In the rest of positions the **non-ideal** values are presented.
Introduced by Chames et al. (1955).
GP sequentially solves optimization problems for each objective function.
There is a priority order.
Our priority order is to minimize altitude level, heading angle and velocity changes to meet comfort \(^1\).

1. Optimizing AC.

\(^1\)Following the guidelines in Cetek (2009), Realistic speed change maneuvers for air traffic conflict avoidance and their impact on aircraft economics.
Lexicographic Goal Programming for VTAC

Our priority order is to minimize altitude level, heading angle and velocity changes to meet comfort \(^1\).

1. Optimizing AC.
2. Optimizing TC together with the following additional constraint,

\[
\sum_{f \in \mathcal{F}} c_f^\gamma \gamma_f \leq z_a^* + \lceil \varepsilon (z_a^{**} - z_a^*) \rceil
\]

\(^1\)Following the guidelines in Cetek (2009), Realistic speed change maneuvers for air traffic conflict avoidance and their impact on aircraft economics.
Our priority order is to minimize altitude level, heading angle and velocity changes to meet comfort ¹.

1. Optimizing AC.

2. Optimizing TC together with the following additional constraint,

$$\sum_{f \in F} c_f^\gamma \gamma_f \leq z_a^* + \lceil \varepsilon (z_a^{**} - z_a^*) \rceil$$

3. Optimizing VC together with the previous additional constraint and the following one,

$$\sum_{f \in F} c_f^\mu |\mu_f| \leq z_t^* + \varepsilon (z_t^{**} - z_t^*)$$

¹Following the guidelines in Cetek (2009), Realistic speed change maneuvers for air traffic conflict avoidance and their impact on aircraft economics.
Compromise Programming

- Introduced by Cochrane and Zeleny (1973).
- The decision maker prefers a solution as much closer as possible to the ideal value.
- A distance is minimized in the objective function.
1. Solves the VTAC model with the following objective function (minimizing the $l_1$ distance):

\[
\min \rho_v \frac{\sum_{f \in F} c^\nu_f |\nu_f| - z_v^*}{z_v^{**} - z_v^*} + \rho_t \frac{\sum_{f \in F} c^\mu_f |\mu_f| - z_t^*}{z_t^{**} - z_t^*} + \rho_a \frac{\sum_{f \in F} c^\gamma_f \gamma - z_a^*}{z_a^{**} - z_a^*}
\]
Double Compromise Programming

- It tries to bound the maximum deviation with respect to the ideal value.
- It consists on two steps.
  1. Minimizing $l_\infty$ distance.
  2. Minimizing $l_1$ distance avoiding higher deviations than the obtained in the previous step.
1. Solves the VTAC model with the following objective function (minimizing the $l_\infty$ distance):

$$\min \max \left\{ \frac{\sum_{f \in \mathcal{F}} c_f^\nu |\nu_f| - z_v^*}{z_v^{**} - z_v^*}, \frac{\sum_{f \in \mathcal{F}} c_f^\mu |\mu_f| - z_t^*}{z_t^{**} - z_t^*}, \frac{\sum_{f \in \mathcal{F}} c_f^\gamma |\gamma_f| - z_a^*}{z_a^{**} - z_a^*} \right\}$$
1. Solves the VTAC model with the following objective function (minimizing the $l_\infty$ distance):

\[
\min \max \left\{ \sum_{f \in F} c_f^\nu |\nu_f| - z_v^*, \sum_{f \in F} c_f^\mu |\mu_f| - z_t^*, \sum_{f \in F} c_f^\gamma |\gamma_f| - z_a^* \right\}
\]

2. Solves the VTAC model with the following objective function (minimizing the $l_1$ distance) avoiding deviations higher than the one obtained in the previous step:

\[
\min \rho_v \frac{\sum_{f \in F} c_f^\nu |\nu_f| - z_v^*}{z_v^{**} - z_v^*} + \rho_t \frac{\sum_{f \in F} c_f^\mu |\mu_f| - z_t^*}{z_t^{**} - z_t^*} + \rho_a \frac{\sum_{f \in F} c_f^\gamma |\gamma_f| - z_a^*}{z_a^{**} - z_a^*}
\]
Outline

Introduction

Variable Neighborhood Search (VNS)

The Velocity, Turn and Altitude Changes model (VTAC)

Multi-objective methods

Computational experience

Conclusions and future research
Initial situation, *roundabout case*
Preliminary computational results

<table>
<thead>
<tr>
<th>Instance</th>
<th>Minotaur</th>
<th>Gap VNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case nc</td>
<td>$z_v^*$</td>
<td>$z_t^*$</td>
</tr>
<tr>
<td>C2-4 1</td>
<td>0.0000</td>
<td>0.0048</td>
</tr>
<tr>
<td>C3-4 3</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>C4-4 6</td>
<td>0.0000</td>
<td>0.0096</td>
</tr>
</tbody>
</table>

Table: Ideal and non-ideal values

<table>
<thead>
<tr>
<th>Instance</th>
<th>Minotaur</th>
<th>VNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>$t_v^*$</td>
<td>$t_t^*$</td>
</tr>
<tr>
<td>C2-4</td>
<td>0.56</td>
<td>0.08</td>
</tr>
<tr>
<td>C3-4</td>
<td>0.17</td>
<td>0.13</td>
</tr>
<tr>
<td>C4-4</td>
<td>71.27</td>
<td>1.09</td>
</tr>
</tbody>
</table>

Table: Computing times (in seconds)

4xIntel Core i5-2430M, 2.40 GHz, 8Gb RAM, Xubuntu 14.04 OS

Gaps obtained as: $\frac{z_{VNS} - z_M}{z_M} \cdot 100\%$
Outline

Introduction

Variable Neighborhood Search (VNS)

The Velocity, Turn and Altitude Changes model (VTAC)

Multi-objective methods

Computational experience

Conclusions and future research
Conclusions and future research

Conclusions:

- A Variable Neighborhood Search algorithm has been presented.
- Conflict situations are solved by performing the three maneuvers.
- The local search and shaking phases are based on angle and velocity discretization (altitude level changes are discrete).
- A multi-objective framework is presented to provide different methods that could be applied to choose solutions when economic or comfort terms are preferred.
Conclusions and future research

Conclusions:

- A Variable Neighborhood Search algorithm has been presented.
- Conflict situations are solved by performing the three maneuvers.
- The local search and shaking phases are based on angle and velocity discretization (altitude level changes are discrete).
- A multi-objective framework is presented to provide different methods that could be applied to choose solutions when economic or comfort terms are preferred.

Future research:

- Integrate VNS and Multi-objective.
- Validation of the VNS approach in an extensive computational experiment.
- Refine the local search as well as the shaking phase of the algorithm.
Thanks a lot for your attention!
A Variable Neighborhood Search approach for the aircraft conflict resolution problem

F. Javier Martín Campo
javier.martin.campo@mat.ucm.es

Joint work with:
A. Alonso-Ayuso, L.F. Escudero, N. Mladenović