Merging Flows in Terminal Maneuvering Area using Time Decomposition Approach

Ji MA

Daniel DELAHAYE, Mohammed SBIHI, Marcel MONGEAU

MAIAA – Laboratory in Applied Mathematics, Computer Science and Automatics for Air Transport.
ENAC – École Nationale de l’Aviation Civile

ICRAT conference, 23 June 2016
Outline

1. Background and problem description
2. Problem modeling
3. Solution approaches
4. Simulation results
5. Conclusions and perspectives
According to Airbus global market forecast 2015-2034, air traffic will **double in the next 15 years**.

- 39 out of the 47 aviation mega cities are **largely congested** today.
  - airport infrastructure is adequate
  - airports with potential for congestion
  - airports where conditions make it impossible to meet demand
Terminal Maneuvering Area (TMA) (1/2)

ICAO DOC 4444: TMA is a control area normally established at the confluence of ATS routes in the vicinity of one or more major aerodromes.

TMA of Paris region

Source: BEA
TMA is one of the most complex types of airspace.

- Runway capacity
- Separation
- Weather
- Noise
- Prohibited area
- Restricted area
- Dangerous area
- Sustainable development
Merging and organizing arrival aircraft from **different entry points** into an orderly stream in a **short time horizon**.

**Diagram:**
- TOD
- IAF
- FAF
- Merging of arrival flows along various IAF towards same FAF
- Spread of traffic flows depending on final destination
- Merging of arrival flows along various STAR towards same IAF
- STAR
- Arrival
Outline

1. Background and problem description
2. Problem modeling
3. Solution approaches
4. Simulation results
5. Conclusions and perspectives
Given data (1/2)

A set of flights \( \mathcal{F} = \{1, \ldots, N_f\} \)

For each flight \( f \in \mathcal{F} \),

- \( e_f \) : initial entering point number at TMA;
- \( t^f_s \) : initial entering time at TMA;
- \( v^f_s \) : initial entering speed at TMA;
- \( c_f \) : wake turbulence category (Heavy, Medium, Light).
Given data (2/2)

A set of routes $\mathcal{R} = \{r_k | k \in \mathbb{N}, 1 \leq k \leq R\}$

where $R$ is the number of routes and $r_k$ is one route with entering point $k$

- One route is composed of several links, the first one starts from the entering point and the last link ends at the runway;
- Each link is defined by two nodes (waypoint) and constitutes a part of the route.
**Figure**: Real aircraft speed profile with respect to time

**Figure**: Speed change model
Two kinds of decision variables associated with the problem:

- $t_f \in T_f$ entering time at TMA of aircraft $f$ (in second), where
  $$T_f = \{t_f^s + j \times \delta t \mid j \in \mathbb{Z}, \Delta t_{min}/\delta t \leq j \leq \Delta t_{max}/\delta t\}$$

- $v_f \in V_f$ speed of aircraft $f$ at the entering point of TMA, where
  $$V_f = \{v_f^s + j \delta v_f \mid j \in \mathbb{Z}, |j| \leq (v_f^{max} - v_f^{min})/\delta v_f\}$$
Two kinds of decision variables associated with the problem:

- $t_f \in T_f$ entering time at TMA of aircraft $f$ (in second), where
  \[ T_f = \{t_s^f + j \times \delta t \mid j \in \mathbb{Z}, \Delta t_{min}/\delta t \leq j \leq \Delta t_{max}/\delta t \} \]

- $v_f \in V_f$ speed of aircraft $f$ at the entering point of TMA, where
  \[ V_f = \{v_{min}^f + j \delta_v^f \mid j \in \mathbb{Z}, |j| \leq (v_{max}^f - v_{min}^f)/\delta_v^f \} \]

Decision vector: $x = (t, v)$
Separation requirements

- Minimum horizontal separation of 3 NM in TMA
- Wake turbulence separation

<table>
<thead>
<tr>
<th>Category</th>
<th>Leading Aircraft</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Heavy</td>
<td>Medium</td>
</tr>
<tr>
<td>Trailing Aircraft</td>
<td>Heavy</td>
<td>4</td>
</tr>
<tr>
<td>Medium</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Light</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

**Table:** Separation minima for two successive aircraft, in NM

- Single-runway separation requirements
Three kinds of conflicts (1/3)

**Link conflicts**

Conflict detected:
\[ d_{f,g}^{u}(x) < s_{fg} \text{ or } d_{f,g}^{v}(x) < s_{fg} \text{ or the order of sequencing changes} \]
Three kinds of conflicts (2/3)

- **Node conflicts**

![Diagram showing node conflicts with detection zones and conflict detection criteria: \( T^{n}_{in}(x) > T^{n}_{out}(x) \)]
Runway conflicts

**Table**: Single-runway separation requirements, in seconds. ¹

<table>
<thead>
<tr>
<th>Category</th>
<th>Leading Aircraft, ( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Heavy</td>
</tr>
<tr>
<td>Trailing Aircraft, ( g )</td>
<td></td>
</tr>
<tr>
<td>Heavy</td>
<td>96</td>
</tr>
<tr>
<td>Medium</td>
<td>157</td>
</tr>
<tr>
<td>Light</td>
<td>207</td>
</tr>
</tbody>
</table>

Objective function

We minimize

- the total number of node conflicts

\[ S(x) = \lambda \left( \sum_{f,g \in \mathcal{F}} \sum_{n \in r_f \cap r_g} N^n_{fg}(x) + \sum_{f,g \in \mathcal{F}} \sum_{l \in r_f \cap r_g} L^l_{fg}(x) + \sum_{f,g \in \mathcal{F}} P_{fg}(x) \right) + \gamma D(x) \]
Objective function

We minimize

- the total number of node conflicts

\[ S(x) = \lambda \left( \sum_{f,g \in \mathcal{F}} \sum_{n \in r_f \cap n_g} N^n_{fg}(x) + \sum_{f,g \in \mathcal{F}} \sum_{l \in r_f \cap n_g} L^l_{fg}(x) + \sum_{f,g \in \mathcal{F}} P_{fg}(x) \right) + \gamma D(x) \]

- the total number of link conflicts
Objective function

We minimize

- the total number of node conflicts
- the total number of link conflicts
- the total number of runway conflicts

\[ S(x) = \lambda \left( \sum_{f,g \in F} \sum_{n \in r_f \cap r_g} N_{fg}^n(x) + \sum_{f,g \in F} \sum_{l \in r_f \cap r_g} L_{fg}(x) + \sum_{f,g \in F} P_{fg}(x) \right) + \gamma D(x) \]
Objective function

We minimize

- the total number of node conflicts
- the total number of link conflicts
- the total number of runway conflicts
- decision deviation: $D(x) = |\{f \in F | t_f(x) \neq t_s^f \text{ or } v_f(x) \neq v_s^f \}|$

Objective function

\[ S(x) = \lambda \left( \sum_{f, g \in F} \sum_{n \in r_f \cap r_g} N_{fg}^n(x) + \sum_{f, g \in F} \sum_{l \in r_f \cap r_g} L_{fg}^l(x) + \sum_{f, g \in F} P_{fg}(x) \right) + \gamma D(x) \]
Outline

1. Background and problem description
2. Problem modeling
3. Solution approaches
4. Simulation results
5. Conclusions and perspectives
Solution approaches

Two resolution approaches

- Resolve the complete problem with an optimization algorithm
- Using time decomposition approach combined with the optimization algorithm

→ Sliding window approach
Sliding window approach (1/2)

- $W$: the length of the sliding window
- $T_s(k)$: the beginning time of the $k^{th}$ sliding window
- $T_e(k)$: the ending time of the $k^{th}$ sliding window
**Sliding window approach (1/3)**

- **$W$** : the length of the sliding window ;
- **$T_s(k)$** : the beginning time of the $k^{th}$ sliding window ;
- **$T_e(k)$** : the ending time of the $k^{th}$ sliding window ;
- **$S$** : time shift of the sliding window.
For each aircraft $f \in \mathcal{F}$,

- $t_s^f$: the earliest entering (start) time at TMA;
- $\overline{t_s^f}$: the latest entering (start) time at TMA;
- $t_e^f$: the earliest landing (end) time;
- $\overline{t_e^f}$: the latest landing (end) time.

\[ t_s^f \leq \overline{t_s^f} \leq t_e^f \leq \overline{t_e^f} \]
Sliding window approach (3/3)
Sliding window approach (3/3)

\[ t_s^f \quad t_e^f \]

Completed

\[ T_s(k) \quad T_e(k) \]

On-going

\[ t_s^f \quad t_e^f \]
Sliding window approach (3/3)

\[ t_s^f \quad t_e^f \]

completed

\[ T_s(k) \quad T_e(k) \]

on-going

\[ t_s^f \quad t_e^f \]

active

\[ T_s(k) \quad T_e(k) \]
Sliding window approach (3/3)

- **Completed**: $t_s^f, t_e^f$
- **On-going**: $t_s^f, t_e^f$
- **Active**: $t_s^f, t_e^f$
- **Planned**: $t_s^f, t_e^f$

$T_s(k), T_e(k)$
Simulated annealing (1/3)

- Temperature
- Stopping criterion
- Objective function
- Neighborhood

Objective function

\[ \text{jumps accepted with probability } e^{-\frac{\Delta E}{T}} \]

Search space

\[ \Delta E \]
### Stopping criterion

- Maximal number of transitions;
- Maximal running time of algorithm;
- No more improvement after a certain number of transitions (or time);
- Final temperature \( T_f = T_{\text{init}} \times \epsilon \).

### Temperature

- Linear Law: \( T_i = T_0 - \beta \times i, \quad \beta > 0 \);
- Logarithmic law: \( T_i = T_0 / \log(i) \);
- Decrease by tier;
- Geometric law: \( T_{i+1} = T_i \times \alpha \quad 0 < \alpha < 1 \).
Neighborhood

- **Roulette wheel selection**
  
  Example:

<table>
<thead>
<tr>
<th>Flight</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of conflicts</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>14</td>
<td>6</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

- $S_F = 34$, random value $\sigma = 0.5 \Rightarrow S_F \times \sigma = 17$
- Recalculate the sum until $S_f \geq 17$, then stop and get $f = 4$
- change $v_f$ or $t_f$ of flight $f$

- **Random generation**
  
  - Generate a random flight $f$
  - change $v_f$ or $t_f$ of flight $f$
Outline

1. Background and problem description
2. Problem modeling
3. Solution approaches
4. Simulation results
5. Conclusions and perspectives
**Case Study**

**Traffic flow proportion**

<table>
<thead>
<tr>
<th>Entry node</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>OKIPA</td>
<td>31.8%</td>
<td>35.5%</td>
<td>36.1%</td>
</tr>
<tr>
<td>BANOX</td>
<td>19.7%</td>
<td>20%</td>
<td>20.1%</td>
</tr>
<tr>
<td>LORNI</td>
<td>33%</td>
<td>27%</td>
<td>25.9%</td>
</tr>
<tr>
<td>MOPAR</td>
<td>15.5%</td>
<td>17.5%</td>
<td>17.9%</td>
</tr>
<tr>
<td>Total Arrivals</td>
<td>239</td>
<td>355</td>
<td>374</td>
</tr>
</tbody>
</table>

**Table:** Daily Traffic flow Characteristics of Paris CDG runway 26L
Sliding window approach + Simulated annealing

**Figure:** Computational time of the two methods

**Table:** Conflicts comparison of the two methods

<table>
<thead>
<tr>
<th>Scenario</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial conflicts</td>
<td>626</td>
<td>1642</td>
<td>1510</td>
</tr>
<tr>
<td>SA residual conflicts</td>
<td>0</td>
<td>0</td>
<td>48</td>
</tr>
<tr>
<td>SA+sliding-window residual conflicts</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
</tbody>
</table>
Sliding window approach + Simulated annealing

**Figure**: Computational time of the two methods

**Table**: Comparison of the two methods for scenario 2

<table>
<thead>
<tr>
<th>Method</th>
<th>SA algorithm</th>
<th>SA+sliding-window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average delay of entrance time at TMA</td>
<td>86 s</td>
<td>81 s</td>
</tr>
<tr>
<td>Entrance delay standard deviation</td>
<td>177 s</td>
<td>160 s</td>
</tr>
<tr>
<td>Average speed change in %</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Speed change standard deviation in %</td>
<td>5.7</td>
<td>4.6</td>
</tr>
</tbody>
</table>

**Table**: Conflicts comparison of the two methods

<table>
<thead>
<tr>
<th>Scenario</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial conflicts</td>
<td>626</td>
<td>1642</td>
<td>1510</td>
</tr>
<tr>
<td>SA residual conflicts</td>
<td>0</td>
<td>0</td>
<td>48</td>
</tr>
<tr>
<td>SA+sliding-window residual conflicts</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
</tbody>
</table>

**Figure**: Number of flights without decision changes

---

Ji MA (ENAC)  
Merging flows using Time Decomposition Approach  
23 June 2016  
29 / 34
FIGURE: Delay at TMA entrance comparison of different objectives
1. Background and problem description

2. Problem modeling

3. Solution approaches

4. Simulation results

5. Conclusions and perspectives
Conclusions

- A mathematical formulation of the aircraft merging problem in TMA
- Novel approach by time decomposition
- Generating a less CPU time and less aircraft deviations solution compared to the simulated annealing applied to the full problem
Perspectives

- Balance the runway capacity
- Integration of TMA and airport
Thank you for your attention!