Operational Concept of Traffic Pattern Classifier for Optimal Ground Holding

Adriana Andreeva-Mori, Naoki Matayoshi
Next Generation Aeronautical Innovation Hub Center, Aeronautical Technology Directorate
Japan Aerospace Exploration Agency (JAXA)
Tokyo, Japan

Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)
17-21 June, 2019, Vienna, Austria
I. Introduction: ground holding overview

- Ground holding (GH): meant to reduce airborne holding, thus leading to fuel savings, lower air traffic controllers’ workload and higher safety.

- Uncertainties (departure time, flight time, capacity, etc.) make it difficult to set optimal control parameters.
  - GH is too short $\rightarrow$ unnecessary airborne holding (burning fuel and occupying airspace).
  - GH is too long $\rightarrow$ no airborne holding, but lost throughput.
  - Finding the balance between those two is the key.

- Most domestic flights in Japan are short range.
  - Ground holding matters a lot!
I. Introduction: GH problem approaches

- Constant buffer approach: easy implementation
- Real-time optimization using complex dynamic models: good performance
- Data-driven approach
  - “Similar days” concept (see research by Gorripaty, A. Pozdnukhov, M. Hansen and Y. Liu; A. Estes, M. O. Ball and D. J. Lovell; K. D. Kuhn)
  - In a nutshell: identify similar days in the past data available, to provide insights into the potential performance of the ground delay program applied on the target day
I. Introduction: GH problem approaches

Data-driven approach

- Estes, Ball and Lovell proposed a black-box-like tool to help decision-makers with ground delay planning.
- The key strength of their tool: it does not require any explicit modeling of the airspace system, but relies on finding similar days in the data.
- Similar approach can be used to “propose GDPs and receive estimates of the predicted performance”.
- Our research aims in the same direction as Estes, Ball and Lovell’s work.
- We look for efficient ways to help air decision-makers to plan traffic management initiatives, in particular ground holding assignments.
I. Introduction: current research outline

Lack of data prevents us from applying approaches adopted by other researchers.

Instead, build a simulated database based on numerical simulation of GH program practiced in Japan, i.e. the constant buffer method.

Cost function: airborne delay costs, ground delay costs, and lost throughput (capacity utilization introduced by Liu and Hansen) costs.

Develop a traffic pattern classifier which predicts the optimal ground holding control parameters based on traffic features.

Can simulate both past and future traffic initiatives:
- immediate as well as long-term, tactical level planning
- performance analysis.
I. Introduction: presentation flow

I. Introduction

II. Operational concept and basic principles

III. Data generation through numerical simulations

IV. Design and feasibility tests of the proposed traffic pattern classifier

V. Concluding remarks and discussions
II. Operational concept

- Long-term goal: develop a ground holding algorithm based on real-time air traffic pattern classification and off-line buffer optimization

Traffic pattern classifier

Traffic class selection

Optimal ground holding database

Ground holding time for each flight

<table>
<thead>
<tr>
<th>Flight</th>
<th>Fix time</th>
<th>Control time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLT001</td>
<td>9:10</td>
<td>-</td>
</tr>
<tr>
<td>FLT002</td>
<td>9:15</td>
<td>9:20</td>
</tr>
<tr>
<td>FLT003</td>
<td>9:17</td>
<td>9:25</td>
</tr>
</tbody>
</table>

FEATURES

- Initial ETA
- Pattern Classification with Machine Learning
- Traffic class A
- Traffic class B
- Traffic class C
- Optimal control parameters
II. Basic principles

1) The traffic pattern classifier acts as a decision support tool to aid the selection of the most adequate ground holding parameters.

2) The ground holding database can contain both simulated and real data, and thus model past as well as new operations.

3) Introduction of the traffic pattern classifier decreases optimality, but eases real-life implementation of the algorithm.
II. Basic principles

1) The traffic pattern classifier acts as a decision support tool to aid the selection of the most adequate ground holding parameters.
   - Decision support tools are no novelty in ATM.
   - The proposed traffic pattern classifier’s output will not be automatically reflected into the ground holding program chosen for the particular day and time.
   - Controller’s experience and expertise play a considerable role in ground holding assignments.
   - The traffic pattern classifier can, to a certain extent, automate this process and deliver predictable performance independent on individual experiences.

2) The ground holding database can contain both simulated and real data, and thus model past as well as new operations.

3) Introduction of the traffic pattern classifier decreases optimality, but eases real-life implementation of the algorithm.
II. Basic principles

1) The traffic pattern classifier acts as a decision support tool to aid the selection of the most adequate ground holding parameters.

2) The ground holding database can contain both simulated and real data, and thus model past as well as new operations.
   - The introduction of new traffic management initiatives is often preceded by extensive computationally expensive simulations which estimate the benefits and compare the results with past operations.
   - A data base of simulated traffic and ground holding control results makes this computational time irrelevant.
   - Numerical simulations can also fill the gap of lack of experience with new traffic management initiatives.

3) Introduction of the traffic pattern classifier decreases optimality, but eases real-life implementation of the algorithm.
II. Basic principles

1) The traffic pattern classifier acts as a decision support tool to aid the selection of the most adequate ground holding parameters.

2) The ground holding database can contain both simulated and real data, and thus model past as well as new operations.

3) Introduction of the traffic pattern classifier decreases optimality, but eases real-life implementation of the algorithm.
   - The machine learning does not generate any new knowledge on ground holding flight assignments itself.
   - The ground holding parameters are chosen real time, but among the predefined set available in the database.
   - However, it solves the computational time issue caused by high-fidelity traffic model simulations and is an important step to real-world implementations.
   - Such an approach might not be common in the ATM field, but has been widely discussed among weather, water, and climate researchers (for example, see the number of presentations related to model learning at the 99th AMS Annual Meeting).
III. Ground holding database generation

A. Ground holding simulation for a single ETA queue: assumptions

- Each traffic scenario described by 30 aircraft landing on a single runway.
- ETA queues randomly generated
- Apply “constant buffer method”: flights are assigned ground holding when they are expected to be held in the air longer than a certain \textit{Buffer} time
- Separation set to 2 min
III. Ground holding database generation

A. Ground holding simulation for a single ETA queue: assumptions

- Departure time error \((\text{DeptError})\)
- Flight time errors \((\text{EnrouteError})\)
- ATA (time of arrival)

\[
\text{ATA} = \text{ETA} + \text{GroundDelay} + \text{DeptError} + \text{EnrouteError} + \text{AirborneATCDelay}
\]
III. Ground holding database generation

A. Ground holding simulation for a single ETA queue: assumptions

Cost function

- Ground delay cost
- Airborne delay cost
- Lost capacity (throughput) cost

Minimize

\[
\text{TotalDelayCost} = \sum_{i=1}^{N=30} (\text{GroundDelay}_{i}^{\text{GHP}} - \text{GroundDelay}_{i}^{\text{Nominal}}) \times c_g \\
+ \sum_{i=1}^{N=30} (\text{AirborneATCDelay}_{i}^{\text{GHP}} - \text{AirborneATCDelay}_{i}^{\text{Nominal}}) \times c_a \\
+ \left( \max(\text{ATA}_{i}^{\text{GHP}}) - \max(\text{ATA}_{i}^{\text{Nominal}}) \right) \times c_c
\]

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
<th>Value [EUR]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c_g)</td>
<td>Airborne delay cost per flight, per minute</td>
<td>24</td>
</tr>
<tr>
<td>(c_a)</td>
<td>Ground delay cost, per flight, per minute</td>
<td>66</td>
</tr>
<tr>
<td>(c_c)</td>
<td>Lost capacity cost, per minute</td>
<td>700</td>
</tr>
</tbody>
</table>
III. Ground holding database generation

B. Ground holding simulation for a single ETA queue: sample results

CostOpt = -6571 EUR, BufferOpt = 9 min
Low CostOpt value → efficient GH

CostOpt = -798 EUR, BufferOpt = 12 min
CostOpt near zero → inefficient GH
III. Ground holding database generation

B. Ground holding simulation for a single ETA queue: sample results

- Savings due to ground holding are dependent on the ETA queue. Choosing the optimum Buffer value might not be sufficient to produce sufficient savings, i.e. the ground holding effect for some ETA queues is limited.
- The optimum Buffer value which minimizes the cost function depends greatly on the individual ETA queue.
- If the air traffic manager can correctly classify the ETA queue pattern, i.e. the traffic pattern, they will be able to set the optimum ground holding program parameters (here, choose the Buffer value) and decide whether to actively pursue the ground holding program application to this particular traffic in view of the potential savings.
III. Ground holding database generation

C. Ground holding simulation for a single ETA queue: database generation

- Following the methodology described above, a database for 1000 different ETA queues is generated.
- To account for departure time and flight time uncertainties in each ETA queue, 1000 run Monte Carlo simulations are done.
- The generated database has the following information for each ground holding control:
  - ETA queue (ETA for all 30 flights in the queue)
  - Median value of the cost function for each *Buffer* between 1 and 15 min
  - The optimum *Buffer* which minimizes the median value of the cost function
  - Traffic parameters such as separation required at the control fix, uncertainties distribution parameters of the departure and flight times
IV. Traffic pattern classifier: design and results

A. Problem statement

- Features: relative traffic density $\text{rtd}$ to describe the ETA queue.

$$\text{rtd}_i = \frac{\text{number of ETA flights in } [i - w_{\text{minus}}, i + w_{\text{plus}}]}{w_{\text{minus}} + w_{\text{plus}}} / \text{ReqSep}$$

- $w_{\text{minus}} = 5 \text{ min}$
- $w_{\text{plus}} = 5 \text{ min}$
- ReqSep = 2 min
- $\text{rtd}_{20} = \frac{7}{5} = 1.4$

- Here, all 58 $\text{rtd}$ values for $w_{\text{minus}} = w_{\text{plus}} = 2 \text{ min}$
IV. Traffic pattern classifier: design and results

A. Problem statement

We try to answer three questions by applying machine learning.

1) What are the potential cost savings for the particular ETA queue?

2) What Buffer should be set to achieve those savings?

3) How robust are the potential cost savings in respect to the Buffer value, i.e. if Buffer is selected with a certain error, how much will the achieved savings differ from the potential optimal ones?
IV. Traffic pattern classifier: design and results

A. Problem statement

- Formulate as a regression problem
- Use support vector machine with a quadratic kernel
- Developed in MATLAB® 2018b and Statistics and Machine Learning Toolbox
- Validation is done by cross-validation.
- The data consists of 1000 ETA queues (rtd values) and the median value of the cost function for Buffer_i (i=1,2,…15) as determined for each ETA queue (Cost_i).
- Cost_i are the target values for the regression problem.
- For each ETA queue and each Buffer_i, a predicted value of Cost_i is found.
- Next, for each ETA queue, CostOptPredicted and BufferOptPredicted are determined as the minimum Cost_i for i ∈ [1, 15].
IV. Traffic pattern classifier: design and results

B. Simulation results

- **CostTrue vs. CostPredicted**
- **RMSE= 632.4 EUR (over all Buffer values)**
- → Can predict savings from GH for any Buffer
IV. Traffic pattern classifier: design and results

B. Simulation results

- Better accuracy for high Buffer values

RMSE of CostPredicted over all Buffer values

Prediction errors of Cost for each Buffer
IV. Traffic pattern classifier: design and results

B. Simulation results: Q1

CostOptPredicted = \min(CostPredicted) for each ETA queue

CostOptTrue vs. CostOptPredicted

RMSE = 451.5

High prediction accuracy → a decision on whether ground delay should be introduced for a certain ETA queue can be made based on the cost savings predicted by the traffic pattern classifier.

1) What are the potential cost savings for the particular ETA queue?

2) What Buffer should be set to achieve those savings?

3) How robust are the potential cost savings in respect to the Buffer value?
IV. Traffic pattern classifier: design and results

B. Simulation results: Q2

- $BufferOptPredicted \rightarrow \text{CostOptPredicted} = \min (\text{CostPredicted})$ for each ETA queue
- $BufferOptTrue \text{ vs. } BufferOptPredicted$
- RMSE = 1.28 min

1) What are the potential cost savings for the particular ETA queue?
2) What $Buffer$ should be set to achieve those savings?
3) How robust are the potential cost savings in respect to the $Buffer$ value?
IV. Traffic pattern classifier: design and results

B. Simulation results: Q3

1) What are the potential cost savings for the particular ETA queue?

2) What Buffer should be set to achieve those savings?

3) How robust are the potential cost savings in respect to the Buffer value?

Assume we choose the optimal Buffer value according to the prediction BufferOptPredicted.

In such a case, the potential savings will be Cost_{BufferOptPredicted}, which, by definition are worse than Cost_{BufferOptTrue} = CostOptTrue.

Unrealised savings: Cost_{Unrealised}

- Mean (Cost_{Unrealised}) = 297.5 EUR
- RMSE = 472.4 EUR
- mean absolute percentage error = 18%
V. Concluding remarks

- We proposed the concept of a traffic pattern classifier applied to optimal ground holding.
- We tested the concept which consists of real-time air traffic pattern classification and off-line buffer optimization on simulated data and proved its feasibility.
- The input of the real-time component consists of traffic features, which are fed into a pre-trained machine learning algorithm to determine whether ground holding should be applied to the current traffic and if so, what parameters should be selected for the ground holding program so that the potential savings are the greatest.
- Based on the ground holding optimal control parameters, departure times can be assigned to each flight part of the ground holding program.
- The classifier successfully predicted the potential savings from ground holding program and predicted an optimal buffer setting which would result in sub-optimal cost savings (the RMSE was 472.4 EUR).
V. Future work

- **Database improvement**
  - more detailed models of the traffic and, whenever possible, real past data
  - currently developing departure time error models to describe the departures from major domestic airports in Japan
  - inclusion of predictability as traffic feature, as discussed by Liu and Hansen, to better describe the uncertainties of the environment

- **Machine learning algorithms improvements**

- **Shift to a “grey box” approach**

- **Discussions with decision makers for prototype testing in real environment over the next 3 years are also ongoing.**
ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI “Synergistic ground holding algorithm based on real-time air traffic pattern classification and off-line buffer optimization”, Grant Number 18H05925 and 19K21092.