Airline disruption management with aircraft swapping and reinforcement learning
Hondet, G., Delgado, L. and Gurtner, G.

A paper presented at the 8th SESAR Innovation Days, Salzburg 03 - 06 Dec, 2018, SESAR.

It is available from the conference organiser at:
https://www.sesarju.eu/sites/default/files/documents/sid/2018/papers...
Airline disruption management with aircraft swapping and reinforcement learning

G. Hondet, L. Delgado, G. Gurtner

École nationale de l’aviation civile

December 5, 2018
Introduction

• Lower costs due to airline disruptions
• Usually, Disruption solution man made by rule of thumb
• Aircraft or flight swapping
• Reinforcement learning
Current work

- V. Mnih et al., “Playing Atari with deep reinforcement learning”, 2013

Work done here
Machine learning technique to discover interesting swap combinations
Outline

1. Simulator
   Specification
   Mechanisms
   Cost & calibration

2. Q learning as solver
   Principle and method
   Q learning algorithm and implementation
   Practical training with the simulator

3. Experiments and results
   Experimental setup
   Results
Outline

1. Simulator
   Specification
   Mechanisms
   Cost & calibration

2. Q learning as solver
   Principle and method
   Q learning algorithm and implementation
   Practical training with the simulator

3. Experiments and results
   Experimental setup
   Results
Specification of simulator

**Purpose**

- Evaluate the delay on a fleet, on a day of operation
- estimate generated costs
- perform actions on the fleet

**Does**

- model reactionary delay
- include other delays as probability distributions
- simulate aircraft swapping and its consequences

**Does not**

- model crew management, nor passengers flow
- manage stand-by aircraft,
- modify or cancel legs
Mechanisms

Timestep
∀ i ∈ [1, m], t_i is the time of the i^{th} landing of the day
(t_1, t_2, \ldots, t_m)

Actions
Allow to alter the simulation,
“swap with aircraft a”

Cost
Immediate cost of a swap
“swapping with a costs c”
Cost & calibration

Cost of what
Delay at departure of the flight after swap

Characteristics

• non linear
• increasing derivative

\[ c(d_1 + d_2) > c(d_1) + c(d_2) \]

• depends on the aircraft type

Calibration
Calibrated against Eurocontrol “Coda Digest 2017”
Outline

1. Simulator
   Specification
   Mechanisms
   Cost & calibration

2. Q learning as solver
   Principle and method
   Q learning algorithm and implementation
   Practical training with the simulator

3. Experiments and results
   Experimental setup
   Results
Reinforcement learning

- Interaction between an agent and its environment
- Find a policy \(\pi: \text{state} \rightarrow \text{action}\)

**Figure:** Reinforcement learning principle
Theoretical basis

State $s \in S$, action $a \in A$.

Maximised value

$$
\mathbb{E} \left( \sum_{t=0}^{T_f} r_t \right) \leftrightarrow Q(s, a) \quad (1)
$$

Bellman equation

$$
Q^* (s, a) = r(s, a) + \sum_{s' \in S} p(s' | s, a) \max_{a'} Q^* (s', a') \quad (2)
$$

- Dynamic programming
- Monte Carlo simulations
**Q learning algorithm**

**procedure** Q-LEARNING($Q$)

\[
\begin{align*}
    s & \leftarrow \text{initial state} \\
    \textbf{while} & \text{episode not finished} \textbf{ do} \\
        a & \leftarrow \text{choose an action from a set} \\
        & \text{play } a, \text{ observe reward } r \text{ and new state } s' \\
        Q & \leftarrow \text{update } Q \text{ with } (s, a, r, s') \\
        s & \leftarrow s' \\
    \textbf{end while}
\end{align*}
\]

end procedure
Lookup table implementation

\[
Q(s,a) = \begin{bmatrix}
Q(s_0,a_0) & Q(s_0,a_1) & \cdots \\
Q(s_1,a_0) & Q(s_1,a_1) & \cdots \\
\vdots & \vdots & \ddots \\
\end{bmatrix}
\]

Update formula

State \(s\), action \(a\), reward \(r\) and next state \(s'\).

\[
Q(s, a) \leftarrow Q(s, a) + \alpha \left( r + \max_{a'} Q(s', a') - Q(s, a) \right)
\]
Bandit methods
Maximise reward, minimise regret

Upper confidence bound

$$Q_t(s, a) + \underbrace{c \sqrt{\ln t / N_t(s, a)}}_{\text{exploration}}$$
Choosing an action

Bandit methods
Maximise reward, minimise regret

Upper confidence bound

\[ Q_t(s, a) + c \sqrt{\frac{\ln t}{N_t(s, a)}} \]  

(4)
Final algorithm

\begin{verbatim}
procedure Q-LEARNING(Q, c, α, A)
    s ← initial state
    while episode not finished do
        a ← CHOOSEACTION(A, c)
        (r, s') ← SIMULATIONSTEP(s, a)
        Q(s, a) ← Q(s, a) + α [r + max_a'∈A Q(s', a') - Q(s, a)]
        s ← s'
    end while
end procedure
\end{verbatim}
Implementing the training

Hyperparameters

- Exploitation exploration trade off
- Initial $Q$ value

Learning rate

$$\sum_{n \geq 0} \alpha_n = \infty; \quad \sum_{n \geq 0} \alpha_n^2 \in \mathbb{R} \quad (5)$$

$$\alpha_n = \frac{1}{N_t(s, a)} \quad (6)$$

Chaining training sessions

$$Q^1 \xrightarrow{\text{training}} Q^2 \xrightarrow{\text{training}} \ldots \xrightarrow{\text{training}} Q^* \quad (7)$$
State space

Observation
Partial information of the environment, \( \mathcal{O} \) the set of observations,

\[(S, A) \xrightarrow{\phi} (\mathcal{O}, A) \xrightarrow{Q} \mathbb{R} \]

Choice of \( \phi \)
- Carries enough information
- But not too specific
- Time independent
Disruption management with reinforcement learning

Hondet, Delgado, Gurtner

Introduction

Simulator

Q learning as solver

Experiments and results

Conclusion

Outline

1. Simulator
   Specification
   Mechanisms
   Cost & calibration

2. Q learning as solver
   Principle and method
   Q learning algorithm and implementation
   Practical training with the simulator

3. Experiments and results
   Experimental setup
   Results
Experimental setup

- Schedule: Vueling, October 12, 2014
- 6 aircraft, 14 stations, 35 flights

Observation
Two different observations tested.

Disruption
Artificial delay added.

Hyperparameters

\[(p_d, c, q_i) = (0.06, 10, -90000)\]
Output format

Spreadsheet like parquet files

Columns
- delays
  - atfm delay
  - departure delay
  - miscellaneous delays
  - reactionary delay
  - artificial delay added
  - taxi time
- action and reward
  - action number
  - swap or not
  - cost
  - cumulative reward
- simulation information
  - departure destination
  - departure origin
  - leg duration
  - departure sobt
  - tail number
  - tail number of swapped aircraft
  - time in the simulation
- Q learning data
  - state-action couple visit count
  - Q value
Learning process

Figure: Average maximum Q values over 5000 episodes.
Comparing with idle behaviour

Figure: Comparing the idle behaviour with the agent.
Conclusion

Results
Cost reduced in some conditions, not reliable enough. Potential lines of research.

Perspectives
- refine observations
- more sophisticated reinforcement learning techniques
- develop further the simulator