Introducing flexibility and demand-based fairness in slot scheduling decisions

Odysseus, June 2018,

Jamie Fairbrother\textsuperscript{1}  Konstantinos G. Zografos\textsuperscript{1}

\textsuperscript{1}Centre for Transport and Logistics, Lancaster University
Introduction
Airport congestion

- In many airports around the world, demand to use the airport infrastructure exceeds available capacity
- This leading to congestion-related delays or infeasible slot scheduling
- The expansion of infrastructure is not possible in the short to medium term and so congestion must be mitigated through the management of demand.
• Outside of the US, demand is managed through the IATA WSG
• Airports are designated as *coordinated* and airlines must obtain slots to use an airport
• A slot is a time interval during which an aircraft can use an airport infrastructure for the purposes of landing or take-off
• A *coordinator* proposes an initial allocation for slots to airlines based on their requests
Slot Allocation Problem: Key Features

- Slots at a coordinated airport are allocated for a six month season
- Requests are typically made in series:
  - A series of slots consists of 5 or more requests for the same time and (usually) the same day
  - Request series are mostly allocated slots the same time
- Allocation should satisfy airport capacity constraints
- Allocation should satisfy turnaround constraints
- Slots are allocated according to the following main priority groups
  - Historics
  - New entrants
  - Others
- The allocation should in some way minimize the schedule displacement
The complexity of allocating slots under IATA scheme has lead to the development of integer programming models.

Main features of IATA scheme first modeled as an MIP in [Zografos et al., 2012] which minimized total displacement.

Incorporation of fairness:

- Displacement proportional to requests [Zografos and Jiang, 2017]
- Displacement proportional to peak-time requests [Fairbrother and Zografos, 2018]

[Zografos et al., 2017] introduced objectives to improve acceptability:

- Minimize maximum displacement
- Maximize slots allocated to acceptable time windows

[Ribeiro et al., 2018] proposed new multiobjective model with:

- Multiple objectives: number of rejections, maximum displacement, total displacement, number of displacement
- Differentiation between “historical” and “change to historical” requests
Inflexibility and Flexibility

- A request series which coincides on at least one day may block another for all it’s requested days.

- Toy example:
  - One historical request series (2 days)
  - One new entrant request series (4 days)
  - Both prefer slot at time period 2
  - Capacity of 1 slot per time period

- Historical request series is given priority and blocks new entrant
Inflexibility and Flexibility

- A request series which coincides on at least one day may block another for all its requested days
- Toy example:
  - One historical request series (2 days)
  - One new entrant request series (4 days)
  - Both prefer slot at time period 2
  - Capacity of 1 slot per time period

- Historical request series is given priority and blocks new entrant
- Allowing flexibility, new entrant series can be allocate slots for days not used by historic request series
Inflexibility and Flexibility

- A request series which coincides on at least one day may block another for all its requested days
- Toy example:
  - One historical request series (2 days)
  - One new entrant request series (4 days)
  - Both prefer slot at time period 2
  - Capacity of 1 slot per time period

- Historical request series is given priority and blocks new entrant
- Allowing flexibility, new entrant series can be allocate slots for days not used by historic request series

- By allowing some flexibility in the slots allocated to a series, we can reduce significantly the overall displacement
Aims

1. Develop a model which allows some flexibility in the slots allocated to a series
2. Incorporate demand-based fairness into new model
3. Study trade-off between flexibility, fairness and efficiency using real slot request data
Flexibility Model and Demand-based Fairness
Slot assignment variables

• Model is generalisation of that of [Zografos et al., 2012]
• Let:

\[ M = \text{set of request (series)} \]
\[ D = \text{set of days in scheduling season} \]
\[ D_m \subseteq D = \text{set of days which make up request } m \]
\[ T = \{1, \ldots, T\} = \text{set of time coordination intervals} \]

• Then, the main decision variables for our model are:

\[ y_{dt}^m = \begin{cases} 
1 & \text{if request } m \text{ is allocated slot at time } t \text{ on day } d, \\
0 & \text{otherwise} 
\end{cases} \]
Range Constraints

• It is necessary that the slots allocated to a request series are close to each other

• Introduce auxiliary variables:

\[ \tau_m = \text{earliest slot allocated to request series } m \]
\[ \bar{\tau}_m = \text{latest slot allocated to request series } m \]

• We require that each request series is assigned slots at times within a given range of each other:

\[ \bar{\tau}_m - \tau_m \leq r_m, \ m \in \mathcal{M} \]

where \( r_m \) is a tolerance specified by the airline making the request \( m \)
Base Model

\[
\text{minimize } \sum_{m \in M} |t - t_m|y_{td}^m
\]

subject to

\[
\sum_{t \in T} y_{td}^m = 1, \ m \in M, \ d \in D_m
\]

\[
\sum_{m \in M} \sum_{t \in T_c} a_{td}^d b_{mc} y_{td}^m \leq u_{ds}^d, \ \forall c \in C, \ d \in D, \ s \in T_c
\]

\[
l_{m_1 m_2} \leq \sum_{t \in T_2} t y_{m_2}^{td} - \sum_{t \in T_1} t y_{m_1}^{td} \leq \bar{l}_{m_1 m_2}, \ (m_1, m_2) \in P, \ d \in D_m
\]

\[
y_{td}^m \leq \bar{y}_m, \ m \in M, \ d \in D_m
\]

\[
\bar{y}_m - y_{td}^m \leq r_m, \ m \in M
\]

\[
y_{td}^m \in \{0, 1\}
\]

(minimize total displacement)

(assign slot to each request)

(rolling capacity constraints)

(turnaround time constraints)

(definition of \(\bar{y}_m, \bar{y}_m\))

(range constraints)
## Notation

### Sets

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{A}$</td>
<td>set of airlines</td>
</tr>
<tr>
<td>$\mathcal{D}$ ($\mathcal{D}_m$)</td>
<td>set of days (for which movement $m$ requested)</td>
</tr>
<tr>
<td>$\mathcal{M}$ ($\mathcal{M}_a$)</td>
<td>set of movement requests (by airline $a$)</td>
</tr>
<tr>
<td>$\mathcal{P} \subseteq \mathcal{M} \times \mathcal{M}$</td>
<td>set of arrival-departure pairs ($m_a$, $m_d$)</td>
</tr>
<tr>
<td>$C$</td>
<td>set of airport capacity constraints</td>
</tr>
<tr>
<td>$T = {1, \ldots, T}$</td>
<td>set of coordination time intervals</td>
</tr>
</tbody>
</table>

### Parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_m$</td>
<td>requested time for movement $m$</td>
</tr>
<tr>
<td>$\delta_c$</td>
<td>duration of constraint $c$</td>
</tr>
<tr>
<td>$l_p$</td>
<td>turnaround time for movement pair $p \in \mathcal{P}$</td>
</tr>
<tr>
<td>$a^d_m$</td>
<td>indicates whether constraint $c \in C$ is active on day $d \in \mathcal{D}$</td>
</tr>
<tr>
<td>$b_{mc}$</td>
<td>contribution of movement $m$ to constraint $c$</td>
</tr>
<tr>
<td>$u_c^{ds}$</td>
<td>capacity for constraint $c \in C$ on day $d \in \mathcal{D}$ at time period $s$</td>
</tr>
<tr>
<td>$r_m$</td>
<td>maximum range of slot times for request $m$</td>
</tr>
<tr>
<td>$\nu_m$</td>
<td>marginal cost of request $m$</td>
</tr>
</tbody>
</table>

### Decision variables

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y^{td}_m$</td>
<td>indicates whether movement $m$ is assigned slot $t$ on day $d$</td>
</tr>
<tr>
<td>$\tau_m$ ($\bar{\tau}_m$)</td>
<td>earliest (latest) slot time assigned to $m$</td>
</tr>
</tbody>
</table>
Fairness - Peak Request Proportionality Principle

• For a schedule to be acceptable it is important that it distributes schedule displacement in a fair way to the airlines

• We define fairness based on the following principle [Fairbrother and Zografos, 2018]

\[
\text{proportion of schedule displacement} \approx \text{proportion of peak requests}
\]

• Solving base model may yield schedule which distributes schedule displacement disproportionately
Calculation of peak periods

- The arrival (departure) slack is the number of extra arrivals (departures) that can be satisfied without breaking capacity constraints.
- A period is an arrival (departure) peak if there is zero slack in the schedule which optimizes base model.

\[
A^d_t = \begin{cases} 
1 & \text{if } (t, d) \text{ is peak arrival period} \\
0 & \text{otherwise}
\end{cases}
\]

\[
D^d_t = \begin{cases} 
1 & \text{if } (t, d) \text{ is peak departure period} \\
0 & \text{otherwise}
\end{cases}
\]
Fairness indices

- We require the following additional notation to take into account airlines:

\[
\begin{align*}
s_a &= \sum_{m \in M_a} \sum_{d \in D_m} \sum_{t \in T} f_t^m y^d_m \\
S &= \sum_{a \in A} s_a \\
c_a &= \sum_{m \in M_a^A} \sum_{d \in D_m^A} A_t^d + \sum_{m \in M_a^D} \sum_{d \in D_m^D} D_t^d \\
C &= \sum_{a \in A} c_a
\end{align*}
\]

(displacement for airline \( a \))

(total schedule displacement)

(number of peak requests by airline \( a \))

(total number of peak requests)

For each airline, we define the following fairness index:

\[
\mu_a = \frac{s_a}{S} = \frac{c_a}{C} = \frac{\text{proportion of schedule displacement}}{\text{proportion of peak requests}}
\]
Fairness constraints

Based on the peak request proportionality principal, this can be interpreted as follows:

\[ \mu_a = 1.0 \quad \text{airline } a \text{ is fairly treated} \]
\[ \mu_a < 1.0 \quad \text{airline } a \text{ is favoured} \]
\[ \mu_a > 1.0 \quad \text{airline } a \text{ is disfavoured} \]

Fairness is then ensured through the addition to the base model of a constraint on the maximum deviation from absolute (MDA) fairness:

\[
\max_{a \in A} |\mu_a - 1| \leq \epsilon
\]
Numerical Tricks
Problem Size

- The flexibility slot allocation model has $O(|T| \sum_{m \in \mathcal{M}} |D_m|)$ variables as opposed to $O(|T||\mathcal{M}|)$ variables in previous models [Zografos et al., 2012, Ribeiro et al., 2018]
- It also has many more turnaround constraints as these are now required for every day and every movement ($\sum_{m \in \mathcal{M}} |D_m|$) rather than just every movement
- We have to employ simplifications and special techniques to solve problem efficiently
Time Windows

- The base model assumes that a request can be allocated a slot any time in the day
- In reality, slots will only be acceptable if they are close to the requested time
- We therefore enforce that each request series is allocated slots in the time window \([t_m - R, t_m + R]\)
- This reduces number of variables to \(O(2R \sum_{m \in M} |D_m|)\)
Date Aggregation

- Some dates in the scheduling season have an identical set of requests.
- By considering groups of identical dates as a single date we can significantly reduce the number of constraints and variables in the problem.
- This reduces the amount of flexibility we have to distribute displacement (important for fairness) but reduces the variability of the resulting schedule.
• For most time periods in the scheduling season, there is sufficient capacity to meet demand, and so most capacity constraints are inactive
• We can speed up the solution of our model by adding capacity constraints as *lazy constraints*
• This is achieved through a callback in the branch-and-bound algorithm
Problem set-up

- Slot allocation problem solved using real request data for a medium-sized airport

<table>
<thead>
<tr>
<th>Request Series</th>
<th>Total requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical</td>
<td>406</td>
</tr>
<tr>
<td>New Entrants</td>
<td>48</td>
</tr>
<tr>
<td>Others</td>
<td>444</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>898</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>15 minutes</th>
<th>1 hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrivals</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>Departures</td>
<td>-</td>
<td>6</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

- 143 (out of 212) scheduling days after date aggregation
- Problems solved using Gurobi 7.5 on Intel(R) Core(TM) i5-4690 CPU @ 3.50GHz
Hierarchical and non-hierarchical methods

Hierarchical solve

- To take into account the different priority classes we solve the model *hierarchically*
- That is, problem is solved first for highest priority group, the airport capacities are updated, the problem is solved for the next highest priority group and so on.

Non-hierarchical solve

- We also solve the problem *non-hierarchically*, that is we solve the model simultaneously for all requests, disregarding the priority classes
- In this case, problem with fairness constraints is computational intractible

Range

- Problem is solved for following ranges on all requests: 0, 15 minutes, 30 minutes, 45 minutes, 1 hour
Non-hierarchical results - Total displacement

- For time windows we use $R = 2.5$ hours
Non-hierarchical results - Variation in Allocated Slots

Most requests series has almost no variation in allocated time.
Hierarchical results - historics

- For time windows we use $R = 4$ hours
- Efficient frontier calculated using $\epsilon$-constraint method

(a) Historics
- Flexible schedules have much better trade-off between displacement and fairness

(b) Historics - Zoomed In
(a) New entrants
- For new entrants, historical schedule has a stronger effect on outcome than flexibility
- Flexibility significantly improves schedule displacement for other requests

(b) Others
Conclusions

• We introduced a new model which incorporates efficiency, fairness and flexibility

• Flexibility in slots allocated to request series subject to range constraints

• Model tested using request data for medium-sized airport
  • Flexibility significantly reduces displacement, but effect attenuates for greater ranges
  • Most requests have little or no variation slots allocated to request series
  • In hierarchical case, there is a significant improvement in displacement and the trade-off with MDA fairness
  • For smaller priority groups, the allocation to higher groups may have a stronger impact on outcomes than range
Future Work

- Investigate different ways of constraining variation in allocated to series:
  - E.g. Number of deviations from a baseline
  - Develop more sophisticated algorithms to solve problems (decomposition techniques)


Backup slides
Rolling Capacity Constraints

- Limited apron capacity, and landing and take-off separations on a runway mean constrain the number of arrival and departures that can be scheduled in a given period of time.

- *Rolling capacity constraints* limit aircraft movements over a given duration (e.g. 15 minutes, 1 hour):

\[
\sum_{m \in M} \sum_{t=s}^{s+\delta_c-1} a_{mc}^d b_{mc}^t d_{mc}^t \leq u_{cs}^d, d \in D, s \in \{1, \ldots, T - \delta_c + 1\}
\]

where

- \(\delta_c\) = duration of constraint \(c\)
- \(a_{mc}^d\) = indicates whether constraint \(c \in C\) is active on day \(d \in D\)
- \(b_{mc}^t\) = contribution of movement \(m\) to constraint \(c\)
- \(u_{cs}^d\) = capacity for constraint \(c \in C\) on day \(d \in D\) at time period \(s\)
Turnaround Constraints

- After an aircraft lands at an airport it requires time to prepare for its next flight (unload/reload baggage, refuel etc.)
- This is called *turnaround time*
- Turnaround constraints are imposed on pairs of requests corresponding to the arrival and departure of an aircraft

\[
\begin{align*}
\bar{l}_{m_1m_2} &\leq \sum_{t \in T} \text{ty}_{m_2}^{td} - \sum_{t \in T} \text{ty}_{m_1}^{td} \leq \bar{l}_{m_1m_2}, \quad (m_1, m_2) \in \mathcal{P}, \quad d \in \mathcal{D}_m
\end{align*}
\]

where

- \(\mathcal{P} \subset \mathcal{M} \times \mathcal{M}\) set of arrival-departure pairs \((m_a, m_d)\)
- \(\bar{l}_{m_1m_2}/\bar{l}_{m_1m_2}\) turnaround time for movement pair \((m_1, m_2) \in \mathcal{P}\)
Selection of schedule from efficient frontier

- Let
  \[ S^* = \text{solution to base model} \]
  \[ S^*(\epsilon) = \text{solution to } \epsilon\text{-fairness problem} \]

- The *price of fairness* for schedule from \( \epsilon \)-fairness problem is defined to be:
  \[ \frac{S^*(\epsilon) - S^*}{S^*} \]

- For some tolerance, \( \alpha \geq 0 \), we select most fair schedule with price of fairness less than \( \alpha \)