Comparison of First-Come First-Served and Optimization Based Scheduling Algorithms for Integrated Departure and Arrival Management

Bae-Seon Park*, Hyeonwoong Lee, and Hak-Tae Lee / Yeonju Eun and Daekeun Jeon
Inha University / Korea Aerospace Research Institute

Zhifan Zhu / Hanbong Lee and Yoon C. Jung
Stinger Ghaffarian Technologies / NASA Ames Research Center

AIAA AVIATION 2018
Outline

- Background
- Two Scheduling Approaches
- Scheduling Result Comparison
- Conclusions
BACKGROUND
Background

- ICN, GMP, and CJU
  - Heavy traffic
- KARI is developing an integrated departure and arrival management system.
  - Schedulers (Dep., Arr.)
  - Taxi time estimation
  - Data management
  - Controller display
Motivation

Scheduling algorithms are one of the key components.
- The Extended First-Come First-Served (EFCFS) scheduler has been developed in Inha University.
- The Mixed Integer Linear Programming (MILP) based scheduler has been developed in KARI in collaboration with NASA.

Compare two different scheduling algorithms systematically
- Cross verification
- Examine the performance differences between EFCFS and MILP
TWO SCHEDULING APPROACHES

Extended First-Come First-Served Approach
Optimization Based Approach
Compatibility of the Two Algorithms
EFCFS Enhancements
Extended First-Come First-Served Approach*

- Sequential scheduling based on priority
- Schedule of the higher priority aircraft is frozen first.
- Departure sequence can be switched.
- Minimum delay solution for each flight

Optimization Based Approach*

- Based on 3-step approach
  - Scheduling problems of the Step 2 and 3 were formulated as MILP optimization

Optimization Based Approach

Runway scheduling

\[
\text{minimize } \sum_{i \in D} (t_i - \text{Earliest}T_i) \quad \text{Earliest possible Takeoff Time}
\]

Decision variable: takeoff time of the departure aircraft \( i \)

Taxiway scheduling

\[
\text{minimize } \alpha_p \left( \sum_{i \in D, r \in R} \max(t_{i,r} - \text{DesiredOff}T_{i,r}, 0) \right) + \alpha_d \left( \sum_{i \in D, r \in R} t_{i,r} - \sum_{i \in D, g \in G} t_{i,g} \right) + \alpha_d \left( \sum_{i \in A, g \in G} t_{i,g} - \sum_{i \in A, r \in R} t_{i,r} \right)
\]

Decision variable: passage times at all intersections along the taxi routes

Required separation between aircraft moving on the surface and other considerations about aircraft movements were all formulated as linear equality/inequality constraints.
Compatibility of the Two Algorithms

- Use the same predetermined routes
- For arrival flights, taxi scheduling only
  - Estimated landing times are given.

Common constraints
- Earliest possible pushback times of departures
- No deadlock in bi-directional taxiway links
- Aircraft separation along the taxiways
- Runway separation based on aircraft wake turbulence category (WTC)
- Miles-In-Trails at selected fixes (MIT)
EFCFS Enhancements

- Runway separation minima based on aircraft WTC*

EFCFS Enhancements

- Applying MIT constraints
  - Extending the node-link from the runway to the metering fix

Extra node-link for departure fix
SCHEDULING RESULT COMPARISON

Problem Set
Scheduling Results
Computation Times
Incheon International Airport (ICN)
Problem Set

- 40 departures and 20 arrivals around 1 hour at ICN
- Fleet mixes of all scenarios are equal
  - Departure: 14 Medium and 26 Heavy class aircraft
  - Arrival: 7 Medium and 13 Heavy class aircraft
- Arrival landing times were not adjusted
  - No landing delays
  - Taxi delays can be added while taxiing from runway exits to gates
- Randomly generated 100 scenarios
  - Gate departure times, estimated landing times, and gate numbers (Taxi routes) are randomly assigned.
Problem Set

- **Runways and departure fixes**

<table>
<thead>
<tr>
<th>Departure fixes</th>
<th>Runways</th>
<th># of flights</th>
<th>MIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>East</td>
<td>15R/33L</td>
<td>5</td>
<td>X</td>
</tr>
<tr>
<td>South East</td>
<td>15R/33L</td>
<td>5</td>
<td>X</td>
</tr>
<tr>
<td>South</td>
<td>15R/33L</td>
<td>6</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>16/34</td>
<td>9</td>
<td>O</td>
</tr>
<tr>
<td>West</td>
<td>16/34</td>
<td>15</td>
<td>O</td>
</tr>
</tbody>
</table>

- **MIT constraints**
  - 15 nautical miles
  - Applied to the West and South fixes
  - The East and South East fixes were unconstrained
Scheduling Results

- Accumulated results for 100 scenarios
- Case 1
  - Without MIT constraints (2 mins / 3 mins)
- Case 2
  - With MIT constraints (2 mins / 3 mins)
- Case 3
  - Artificially increased runway separation minima without MIT constraints for takeoffs (2 → 5 mins / 3 → 10 mins)
Scheduling Results – Case 1

Delay distributions

- MILP shows smaller average runway takeoff delay

* Delay definitions

1. Gate delay
   = Calculated push-back time – Original push-back time
2. Takeoff delay
   = Calculated takeoff time – Original takeoff time
3. Original takeoff time
   = Original push-back time + Unimpeded taxi time

Gate departure delay

Runway takeoff delay

Average delay of EFCFS: 2.8 minutes
Average delay of MILP: 2.6 minutes

Average delay of EFCFS: 3.2 minutes
Average delay of MILP: 2.6 minutes
Scheduling Results – Case 1

- Maximum delay distributions
  - MILP has better performances than EFCFS
  - EFCFS is slightly shifted to the right side

![Diagram showing gate departure delay and runway takeoff delay comparisons between EFCFS and MILP.]
Scheduling Results – Case 1

- Distribution of makespan differences
  - MILP shows slightly better performance
Scheduling Results – Case 2

- Delay distributions
  - EFCFS has more flights with 5 – 10 minutes runway takeoff delays
  - MILP has more flights with the runway takeoff delays in 1 minute
Scheduling Results – Case 2

- Maximum delay distributions
  - The difference between MILP and EFCFS became smaller than Case 1

![Diagram showing gate departure delay and runway takeoff delay](image-url)
Scheduling Results – Case 2

- Distribution of makespan differences
  - MILP shows slightly better performance
Scheduling Results – Case 3

- Delay distributions
  - EFCFS shows larger average delays for both gate departure and runway takeoff
  - EFCFS has more flights with the delays larger than 70 minutes

Average delay of EFCFS: 34.7 minutes
Average delay of MILP: 30 minutes

Average delay of EFCFS: 35.2 minutes
Average delay of MILP: 30 minutes
Scheduling Results – Case 3

- **Maximum delay distributions**
  - EFCFS produced larger maximum delays
  - Distributions of EFCFS are shifted to the right side

![Gate departure delay](image)

- Average maximum delay of EFCFS: 87.3 minutes
- Average maximum delay of MILP: 73.2 minutes

![Runway takeoff delay](image)

- Average maximum delay of EFCFS: 87.8 minutes
- Average maximum delay of MILP: 73.2 minutes
Scheduling Results – Case 3

- Distribution of makespan differences
  - The Makespan differences are biased in the positive direction
  - MILP shows much better performance with large runway separations

![Histogram showing distribution of makespan differences. The average makespan of EFCFS is 169.5 minutes, whereas the average makespan of MILP is 154.1 minutes.](image-url)
Computation Times

- **Case 1 (No MIT)**
  - EFCFS: 0.82 seconds
  - MILP: 6.39 seconds

- **Case 2 (with MIT)**
  - EFCFS: 0.99 seconds
  - MILP: 9.22 seconds

* Desktop specification
  Intel i7-6820HQ, 2.79 GHz / 32GB RAM
Scheduling results – Summary

- **MILP**
  - Slightly smaller average and maximum takeoff delays
  - Slightly smaller average makespans
- **EFCFS** is about 10 times faster for the given problem size.
- **MILP’s** advantage is more noticeable in high delay situations.
- **Applying MIT constraints**
  - The differences in results between EFCFS and MILP became smaller.
  - The computation times of MILP were increased.
CONCLUSIONS
Conclusions

- Two different scheduling approaches were compared
  - Common constraints were considered
  - 100 scenarios were randomly generated
- MILP generally showed better performance in terms of minimizing delays, but the differences were small.
- EFCFS is much faster in computational performance
  - Real time situations
  - Scheduling large number of aircraft
Future Research Plans

- Testing more scenarios considering higher delay such as operations with severe weather condition or future traffic demand

- Handling uncertainty
  - Add buffer times
  - Update periodically with fast-time simulation
  - Use probabilistic model
Questions?