Transporting people with reduced mobility through the airport

Marjan van den Akker, René van Twist, Han Hoogeveen

12 November, 2015
My background

- M.Sc. Mathematics, Eindhoven University of Technology
- Ph.D. Scheduling, Eindhoven University of Technology
- Engineer in Informatics Division, National Aerospace Center NLR: Air Traffic Management, simulation, optimization
- Lecturer/Researcher, Department of Information and Computing Sciences, Utrecht University: advanced planning algorithms, simulation, software- and gameprojects.
The problem

▶ People with reduced mobility (PRM) require assistance when travelling through the airport
▶ Airport with different terminals
▶ Within terminal:
  • Employees transporting up to two PRMs, depending on their disability.
  • Outside lounges PRMs have to be assisted/supervised by employee (not left alone)
  • Supervised lounges
▶ Special inter-terminal busses
▶ Special platform busses to access aircraft at a remote stand
▶ Busses transport at most 12 PRM’s
We want to find a schedule for the airport employees and busses. 
We received instance data from an airport transporting 300 to 500 PRM’s daily, which we used for our experiments.
Journeys of PRMs

- Arrival journey.
- Departure journey.
- Transfer journey.
Journeys of PRMs

- Arrival journey.
- Departure journey.
- Transfer journey.
Journeys of PRMs

- Arrival journey.
- Departure journey.
- **Transfer journey.**
Waiting

Waiting in the lounge:

A PRM obtains uncomfortable waiting by:

- Waiting at handover
- Detour
Objective (2)

Find a schedule of employees and busses such that:

- We minimize the number of declined PRMs where we prioritize PRMs who booked ahead.
- Minimize the uncomfortable waiting time

The schedule must be found within 2 minutes of computation time.

$\mathcal{NP}$-hard optimization problem.
Approach Reinhardt et al

- List insertion heuristic
- Simulated annealing to find good order of list
- Found good schedules some 0 PRMs declined.
- Schedules contain uncomfortable waiting time
Additions

Add robustness to the optimization objectives.

Synchronization

▪ Segment groups
  • Fixed start times for segment groups involving boarding/disembarking
  • Use the lounges to split journeys.
▪ Improve schedule by focus on one or more PRMs sharing a trip.

Strictly avoid uncomfortable waiting time: fixed at 0

In a deterministic setting
Robustness

A connection is robust when there is slack time between subsequent segments on an employee or bus.

The slack \( s_{r_1, r_2} \) of 2 subsequent segments \( r_1 \) and \( r_2 \) on employee \( e \) is equal to the start time of \( r_2 \) minus the end time of \( r_1 \) minus the travel time:

\[
rp(r_1, r_2) = \begin{cases} 
0 & \text{if } r_1 \text{ and } r_2 \text{ concern the same PRM} \\
(20 - \min(s_{r_1, r_2}, 20))^2 & \text{otherwise}
\end{cases}
\]

We use lexicographical ordering to include robustness in our objective function.
Synchronization: segment groups

Transfer:

PB L B IB L G B PB

PB L G B PB
Synchronization: segment groups

Segment Groups:

Transfer:

- Fixed start times for segment groups involving boarding/disembarking of aircraft.

Strictly avoid uncomfortable waiting time: fixed at 0
Synchronization: Sharing segments

Segments between 2 PRMs can be shared with one employee/bus:
- Arrive with the same flight.
- Depart with the same flight.
- Travel between the same terminals.

Transfer journeys of 2 PRMs:
Synchronization: Sharing segments

Segments between 2 PRMs can be shared with one employee/bus:

- Arrive with the same flight.
- Depart with the same flight.
- Travel between the same terminals.

Transfer journeys of 2 PRMs:
Decomposition Model

In our approach to this problem we want to make use of a decomposition model:

**Master-level:** Determine feasible start times for the segments.

- Local Search, i.e. iterative search heuristic.

**Sub-level,** i.e. in each step of Local Search: assign the tasks to the employees

- Matching
Local search

- Simulated annealing.
- Assign feasible start times for segments.
- Accepts or declines PRMs.
- Starts with empty solution.
- Mutations:
  - Plan PRM
  - Decline PRM
  - Merge segment group
  - Move segment group
  - Split merged segment group.
- Objective function: lexicographical ordering.
Local matching

- Uses old solution prior mutation.
- Represents the schedule as graph $G = \{V, E\}$.
- Matches mutated segments along with competing segments.

Before ($R_{source}$) | Overlapping ($R_{overlap}$) | After ($R_{sink}$)

Employee 1: $r_1$ | $r_2$ | $r_3$
Employee 2: $r_4$ | $r_5$ | $r_6$
Employee 3: $r_7$ | $r_8$ | $r_9$

- Part of the schedule change.
- Exchange sequences of tasks.
Local matching

- Uses old solution prior mutation.
- Represents the schedule as graph $G = \{V, E\}$.
- Matches mutated segments along with competing segments.

Before ($R_{source}$)  Overlapping ($R_{overlap}$)  After ($R_{sink}$)

Employee 1: $r_1 \rightarrow r_4 \rightarrow r_7$
Employee 2: $r_4 \leftrightarrow r_5 \leftrightarrow r_8$
Employee 3: $r_7 \leftrightarrow r_8 \rightarrow r_9$

- Part of the schedule change.
- Exchange sequences of tasks.
Local matching

- Uses old solution prior mutation.
- Represents the schedule as graph $G = \{ V, E \}$.
- Matches mutated segments along with competing segments.

Before ($R_{source}$)  Overlapping ($R_{overlap}$)  After ($R_{sink}$)

Employee 1: $r_1$  $r_5$  $r_9$
Employee 2: $r_4$  $\text{new segment } r$  $r_6$
Employee 3: $r_7$  $r_8$  $r_2$  $r_3$

- Part of the schedule change.
- Exchange sequences of tasks.
Results Local matching

<table>
<thead>
<tr>
<th>instance</th>
<th>time(s)</th>
<th>iterations</th>
<th>Best</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>declined</td>
<td>robust</td>
</tr>
<tr>
<td>1</td>
<td>1,8</td>
<td>11143</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>70,7</td>
<td>500000</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>43,4</td>
<td>289371</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>17,7</td>
<td>115761</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>51,9</td>
<td>243457</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>47,0</td>
<td>160797</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>10,4</td>
<td>69227</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>14,5</td>
<td>82530</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>9,5</td>
<td>58167</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>52,4</td>
<td>257654</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>51,2</td>
<td>174627</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Optimum solutions except instance 2. (broken instance)
- Robust
- Fast
Simulation

- Robust vs no robust.
- Disturbances:
  - Additional PRMs. (100)
  - Flight delays.
  - Walking time differs.
- Start with initial schedule.
- Same instances, seed, flight delays and PRMs.
Simulation results

<table>
<thead>
<tr>
<th>statistic</th>
<th>robust mean</th>
<th>robust var.</th>
<th>no robust mean</th>
<th>no robust var.</th>
<th>p score</th>
</tr>
</thead>
<tbody>
<tr>
<td>declined booked</td>
<td>1, 15</td>
<td>1, 49</td>
<td>1, 08</td>
<td>1, 49</td>
<td>0, 52</td>
</tr>
<tr>
<td>declined immediate</td>
<td>2, 4</td>
<td>3, 89</td>
<td>2, 37</td>
<td>3, 88</td>
<td>0, 56</td>
</tr>
<tr>
<td>wait time</td>
<td>87, 35</td>
<td>2024, 87</td>
<td>170, 6</td>
<td>8212, 02</td>
<td>$1, 10^{-15}$</td>
</tr>
<tr>
<td>full reschedules</td>
<td>11, 4</td>
<td>20, 50</td>
<td>11, 15</td>
<td>21, 12</td>
<td>0, 37</td>
</tr>
</tbody>
</table>

- Less wait time.
- Same amount of rescheduling.
Found schedules 0 PRMs declined
Realized 0 uncomfortable waiting time.
Robustness cost function helps to improve schedule
Identical shifts makes problem easier.
If 0 uncomfortable waiting time turns out to be infeasible, algorithm can be extended with splitting of segment groups.