Stakeholder Cooperation for Improved Predictability and Lower Cost Remote Services

Joen Dahlberg, Tatiana Polishchuk, Valentin Polishchuk, Christiane Schmidt
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.
• In Sweden: two remotely controlled airports in operation, five more studied.
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.
• In Sweden: two remotely controlled airports in operation, five more studied.
• Splits the cost of Air Traffic Services (ATS) provision and staff management between several airports
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.
• In Sweden: two remotely controlled airports in operation, five more studied.
• Splits the cost of Air Traffic Services (ATS) provision and staff management between several airports
  • Labour accounts for up to 85% of ATS cost
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.
• In Sweden: two remotely controlled airports in operation, five more studied.
• Splits the cost of Air Traffic Services (ATS) provision and staff management between several airports
  • Labour accounts for up to 85% of ATS cost
  ➤ Significant cost savings for small airports (30-120 movements a day)
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.
• In Sweden: two remotely controlled airports in operation, five more studied.
• Splits the cost of Air Traffic Services (ATS) provision and staff management between several airports
  • Labour accounts for up to 85% of ATS cost
  ➡ Significant cost savings for small airports (30-120 movements a day)
• To ensure safety: No simultaneous movements at airports controlled from the same module
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.

• In Sweden: two remotely controlled airports in operation, five more studied.

• Splits the cost of Air Traffic Services (ATS) provision and staff management between several airports
  • Labour accounts for up to 85% of ATS cost
    ➪ Significant cost savings for small airports (30-120 movements a day)

• To ensure safety: No simultaneous movements at airports controlled from the same module
  ➪ In extreme case in Sweden: simultaneous movements at all five airports
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.
• In Sweden: two remotely controlled airports in operation, five more studied.
• Splits the cost of Air Traffic Services (ATS) provision and staff management between several airports
  • Labour accounts for up to 85% of ATS cost
    → Significant cost savings for small airports (30-120 movements a day)
• To ensure safety: No simultaneous movements at airports controlled from the same module
  → In extreme case in Sweden: simultaneous movements at all five airports
    → Each airport needs separate RTM
• Remotely operated towers enable control of multiple aerodromes from a single Remote Tower Module (RTM) in a Remote Tower Center.
• In Sweden: two remotely controlled airports in operation, five more studied.
• Splits the cost of Air Traffic Services (ATS) provision and staff management between several airports
  • Labour accounts for up to 85% of ATS cost
    ➪ Significant cost savings for small airports (30-120 movements a day)
• To ensure safety: No simultaneous movements at airports controlled from the same module
  ➪ In extreme case in Sweden: simultaneous movements at all five airports
    ➪ Each airport needs separate RTM
  ➪ Possibilities to perturb flight schedules? (current flight schedules consider only the single airport, ATCO might have to put a/c on hold anyhow…)
Problem Formulation
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL
• Split the time into 5-min intervals, called *slots*, and put every flight into its slot
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL
• Split the time into 5-min intervals, called slots, and put every flight into its slot

Input matrix F:
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL
• Split the time into 5-min intervals, called slots, and put every flight into its slot
  ➡ Input matrix F:
  ❖ Row per airport (a)

<table>
<thead>
<tr>
<th></th>
<th>04:00</th>
<th></th>
<th>05:00</th>
<th></th>
<th>06:00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 5 10 15 20 25 30 35 40 45 50 55</td>
<td>0 5 10 15 20 25 30 35 40 45 50 55</td>
<td>0 5 10 15 20 25 30 35 40 45 50 55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP1</td>
<td>0 1 1 0 0 0 0 0 1 0 0 0</td>
<td>0 1 1 0 0 0 0 0 1 0 0 0</td>
<td>0 0 1 0 0 1 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP2</td>
<td>0 0 1 0 1 0 0 1 0 1 0 0</td>
<td>1 1 0 1 0 1 0 0 0 1 0 0</td>
<td>0 0 1 0 1 0 1 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP3</td>
<td>0 0 0 0 0 1 0 1 0 0 0 0</td>
<td>0 1 1 0 0 0 0 0 1 0 0 0</td>
<td>0 1 0 1 0 1 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP4</td>
<td>0 0 1 0 1 0 0 0 1 0 0 0</td>
<td>0 0 0 0 1 0 0 0 0 0 0 0</td>
<td>1 0 1 1 0 0 0 0 1 1 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP5</td>
<td>0 0 1 1 0 0 0 1 0 1 0 0</td>
<td>1 1 1 0 0 0 0 0 1 0 0 0</td>
<td>0 0 1 0 0 1 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL
  • Split the time into 5-min intervals, called *slots*, and put every flight into its slot
  ➡ Input matrix F:
    ✷ Row per airport (a)
    ✷ Column per each slot (s)
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL
• Split the time into 5-min intervals, called *slots*, and put every flight into its slot

**Input matrix F:**
- Row per airport (a)
- Column per each slot (s)
- $F_{as} = 1$ if a movement happens at airport a at time slot s

<table>
<thead>
<tr>
<th></th>
<th>04:00</th>
<th></th>
<th>05:00</th>
<th></th>
<th>06:00</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td>50</td>
<td>55</td>
</tr>
<tr>
<td>API1</td>
<td>0 0 1 1 0 0 0 0 1 1 0</td>
<td>0 1 1 1 0 0 0 0 1 0 0</td>
<td>0 0 0 1 0 0 1 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP2</td>
<td>0 0 1 0 1 0 0 1 0 1 1 0</td>
<td>1 1 0 1 0 1 0 0 0 1 0 0</td>
<td>0 0 0 1 1 0 1 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP3</td>
<td>0 0 0 0 1 0 0 0 0 1 0 0</td>
<td>0 1 1 0 0 0 0 0 1 0 1 0</td>
<td>0 0 0 1 0 1 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP4</td>
<td>0 0 0 1 0 1 0 0 0 1 0 0</td>
<td>0 0 0 0 1 0 0 0 0 1 1 0 0</td>
<td>0 1 1 0 0 0 0 1 1 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP5</td>
<td>0 0 1 1 0 0 0 1 0 1 1 0</td>
<td>0 1 1 0 0 0 0 1 0 0 0 0</td>
<td>0 0 0 1 0 1 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL

• Split the time into 5-min intervals, called slots, and put every flight into its slot

→ Input matrix F:
  - Row per airport (a)
  - Column per each slot (s)
  - \( F_{as} = 1 \) if a movement happens at airport a at time slot s
  - \( F_{as} = 0 \) otherwise

<table>
<thead>
<tr>
<th></th>
<th>04:00</th>
<th></th>
<th>05:00</th>
<th></th>
<th>06:00</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>AP1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>AP3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>AP5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL
• Split the time into 5-min intervals, called slots, and put every flight into its slot
  ➡ Input matrix $F$:
  ❖ Row per airport (a)
  ❖ Column per each slot (s)
  ❖ $F_{as} = 1$ if a movement happens at airport a at time slot s
  ❖ $F_{as} = 0$ otherwise

<table>
<thead>
<tr>
<th></th>
<th>04:00</th>
<th></th>
<th>05:00</th>
<th></th>
<th>06:00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>55</td>
<td>0</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>45</td>
<td>50</td>
<td>55</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>50</td>
<td>55</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

• Conflict: two movements during the same slot in different airports (in F: two 1s in the same column)
• Input: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL

• Split the time into 5-min intervals, called *slots*, and put every flight into its slot

→ Input matrix F:
  
  ❖ Row per airport (a)
  ❖ Column per each slot (s)
  ❖ \( F_{as} = 1 \) if a movement happens at airport a at time slot s
  ❖ \( F_{as} = 0 \) otherwise

<table>
<thead>
<tr>
<th></th>
<th>04:00</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>05:00</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>06:00</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 5 10 15 20 25 30 35 40 45 50 55</td>
<td>0 5 10 15 20 25 30 35 40 45 50 55</td>
<td>0 5 10 15 20 25 30 35 40 45 50 55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP1</td>
<td>0 0 1 1 0 0 0 0 1 0 0 0</td>
<td>0 1 1 1 0 0 0 0 1 0 0 0</td>
<td>0 0 0 1 0 0 1 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP2</td>
<td>0 0 1 0 1 0 0 1 0 1 1 0</td>
<td>1 1 0 1 0 1 0 0 0 1 0 0</td>
<td>0 0 0 1 1 0 1 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP3</td>
<td>0 0 0 0 0 1 0 1 0 0 0 0</td>
<td>0 1 1 0 0 0 0 1 0 1 0 0</td>
<td>0 0 0 1 0 1 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP4</td>
<td>0 0 0 1 0 1 0 0 0 1 0 0</td>
<td>0 0 0 0 1 0 0 0 0 1 1 0</td>
<td>0 1 1 0 0 0 1 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP5</td>
<td>0 0 1 1 0 0 0 1 0 1 0 0</td>
<td>0 1 1 1 0 0 0 0 1 0 0 0</td>
<td>0 0 0 1 0 1 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• **Conflict:** two movements during the same slot in different airports (in F: two 1s in the same column)

• Conflicting airports should never be assigned to the same RTM
• **Input**: Aircraft movements at each airport from Demand Data Repository (DDR) hosted by EUROCONTROL

• **Split the time into 5-min intervals, called slots**, and put every flight into its slot

→ **Input matrix F**:
  - Row per airport (a)
  - Column per each slot (s)
  - \( F_{as} = 1 \) if a movement happens at airport a at time slot s
  - \( F_{as} = 0 \) otherwise

<table>
<thead>
<tr>
<th></th>
<th>04:00</th>
<th>05:00</th>
<th>06:00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0  5 10 15 20 25 30 35 40 45 50 55</td>
<td>0  5 10 15 20 25 30 35 40 45 50 55</td>
<td>0  5 10 15 20 25 30 35 40 45 50 55</td>
</tr>
<tr>
<td>AP1</td>
<td>0  1 1 0 0 0 0 0 1 0 0</td>
<td>1 1 1 0 0 0 0 0 1 0 0</td>
<td>0  0 0 1 0 0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>AP2</td>
<td>0  1 0 1 0 1 0 1 1 0</td>
<td>1 1 0 1 0 1 0 0 0 1 0</td>
<td>0  0 0 1 1 0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>AP3</td>
<td>0  0 0 0 0 1 0 1 0 0</td>
<td>0 1 1 0 0 0 0 0 1 0 1</td>
<td>0  0 0 1 0 1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>AP4</td>
<td>0  0 1 0 1 0 0 0 1 0</td>
<td>0 0 0 0 1 0 0 0 1 0 1</td>
<td>0  1 0 1 0 0 0 1 1 0 0 0</td>
</tr>
<tr>
<td>AP5</td>
<td>0  0 1 1 0 0 0 1 0 1</td>
<td>0 1 1 0 0 0 0 0 1 0 0</td>
<td>0  0 0 1 0 0 1 0 0 0 0 0</td>
</tr>
</tbody>
</table>

**Conflict**: two movements during the same slot in different airports (in F: two 1s in the same column)

• **Conflicting airports should never be assigned to the same RTM**
• Output: Shifted flights and Airport-to-RTM assignment
• Output: Shifted flights and Airport-to-RTM assignment
• Goal: "small" shifts to the flight schedules → decreased number of required RTMs
• Output: Shifted flights and Airport-to-RTM assignment
• Goal: “small” shifts to the flight schedules → decreased number of required RTMs
• Measure for shift?
• Output: Shifted flights and Airport-to-RTM assignment
• Goal: ”small” shifts to the flight schedules → decreased number of required RTMs
• Measure for shift?
  ❖ Maximum slot shift $\Delta$ (in minutes; multiple of 5, as we shift only by whole slots)
• Output: Shifted flights and Airport-to-RTM assignment
• Goal: ”small” shifts to the flight schedules \( \rightarrow \) decreased number of required RTMs
• Measure for shift?
  ❖ Maximum slot shift \( \Delta \) (in minutes; multiple of 5, as we shift only by whole slots)
  ❖ Number of shifts \( S \)
• Output: Shifted flights and Airport-to-RTM assignment
• Goal: “small” shifts to the flight schedules → decreased number of required RTMs
• Measure for shift?
  ❖ Maximum slot shift $\Delta$ (in minutes; multiple of 5, as we shift only by whole slots)
  ❖ Number of shifts $S$
• MAP = maximum number of airports per module
Formal problem definition:
Formal problem definition:
Flights Rescheduling and Airport-to-Module Assignment (FRAMA)
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

Given:
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

*Given:*
- Flight slots in a set of airports (the matrix F)
Formal problem definition:

Flights Rescheduling and Airport-to-Module Assignment (FRAMA)

Given:
- Flight slots in a set of airports (the matrix F)
- Maximum allowable shift of a flight
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

Given:
- Flight slots in a set of airports (the matrix $F$)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, $S$
Formal problem definition:

Flights Rescheduling and Airport-to-Module Assignment (FRAMA)

Given:
- Flight slots in a set of airports (the matrix F)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, S
- Maximum number of airports per RTM, MAP
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

**Given:**
- Flight slots in a set of airports (the matrix F)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, S
- Maximum number of airports per RTM, MAP
- Total number of modules, M
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

**Given:**
- Flight slots in a set of airports (the matrix F)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, S
- Maximum number of airports per RTM, MAP
- Total number of modules, M

**Find:** New slots for the flights and an assignment of airports to RTMs such that
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

Given:
- Flight slots in a set of airports (the matrix $F$)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, $S$
- Maximum number of airports per RTM, $MAP$
- Total number of modules, $M$

Find: New slots for the flights and an assignment of airports to RTMs such that
- At most $S$ flights are moved
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

Given:
- Flight slots in a set of airports (the matrix F)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, S
- Maximum number of airports per RTM, MAP
- Total number of modules, M

Find: New slots for the flights and an assignment of airports to RTMs such that
- At most S flights are moved
- Each flight is moved by at most Δ
Formal problem definition: 

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)** 

**Given:**  
- Flight slots in a set of airports (the matrix $F$)  
- Maximum allowable shift of a flight  
- Maximum total number of allowable shifts, $S$  
- Maximum number of airports per RTM, MAP  
- Total number of modules, $M$  

**Find:** New slots for the flights and an assignment of airports to RTMs such that  
- At most $S$ flights are moved  
- Each flight is moved by at most $\Delta$  
- No conflicting airports are assigned to the same RTM
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

Given:
- Flight slots in a set of airports (the matrix F)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, $S$
- Maximum number of airports per RTM, MAP
- Total number of modules, $M$

Find: New slots for the flights and an assignment of airports to RTMs such that
- At most $S$ flights are moved
- Each flight is moved by at most $\Delta$
- No conflicting airports are assigned to the same RTM
- At most MAP airports are assigned per module
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

**Given:**
- Flight slots in a set of airports (the matrix $F$)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, $S$
- Maximum number of airports per RTM, $MAP$
- Total number of modules, $M$

**Find:** New slots for the flights and an assignment of airports to RTMs such that
- At most $S$ flights are moved
- Each flight is moved by at most $\Delta$
- No conflicting airports are assigned to the same RTM
- At most $MAP$ airports are assigned per module
- At most $M$ modules are used
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

Given:
- Flight slots in a set of airports (the matrix F)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, S
- Maximum number of airports per RTM, MAP
- Total number of modules, M

Find: New slots for the flights and an assignment of airports to RTMs such that
- At most S flights are moved
- Each flight is moved by at most Δ
- No conflicting airports are assigned to the same RTM
- At most MAP airports are assigned per module
- At most M modules are used
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

**Given:**
- Flight slots in a set of airports (the matrix F)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, S
- Maximum number of airports per RTM, MAP
- Total number of modules, M

**Find:** New slots for the flights and an assignment of airports to RTMs such that
- At most S flights are moved
- Each flight is moved by at most Δ
- No conflicting airports are assigned to the same RTM
- At most MAP airports are assigned per module
- At most M modules are used

---

**Decision problem**

For optimisation problem: Move one constraint in objective function
Formal problem definition:

**Flights Rescheduling and Airport-to-Module Assignment (FRAMA)**

**Given:**
- Flight slots in a set of airports (the matrix $F$)
- Maximum allowable shift of a flight
- Maximum total number of allowable shifts, $S$
- Maximum number of airports per RTM, MAP
- Total number of modules, $M$

**Find:** New slots for the flights and an assignment of airports to RTMs such that
- At most $S$ flights are moved
- Each flight is moved by at most $\Delta$
- No conflicting airports are assigned to the same RTM
- At most MAP airports are assigned per module
- At most $M$ modules are used

**Decision problem**

For optimisation problem: Move one constraint in objective function
For us: Minimize number $M$ of used RTMs, while respecting the bounds $\Delta$, $S$, MAP
Problem Complexity
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.
Proof: Reduction from Partition into Triangles (PIT)
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.
Proof: Reduction from Partition into Triangles (PIT)
• Graph $G = (V,E)$ (of maximum degree four)
• Can $V$ be partitioned into triples $V_1, V_2, \ldots, V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and $MAP=3$.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1, V_2, \ldots, V_{|V|/3}$ such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:
**Theorem:** FRAMA is NP-complete, even if Δ=0 and MAP=3.

Proof: Reduction from **Partition into Triangles (PIT)**

- Graph \( G = (V,E) \) (of maximum degree four)
- Can \( V \) be partitioned into triples \( V_1, V_2, \ldots, V_{|V|/3} \), such that each \( V_i \) forms a triangle in \( G \)
  (for each triple of vertices \( V_i \) each vertex in \( V_i \) is connected to both other vertices in \( V_i \))?

Given an instance of PIT (graph \( G = (V,E) \) with max degree four) we construct the matrix \( F \), the input of FRAMA:
- One airport per vertex \( \rightarrow \) \( F \) has \( |V| \) rows
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1, V_2, \ldots, V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$'s complement)
Theorem: FRAMA is NP-complete, even if $\Delta=0$ and MAP=3.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1, V_2, \ldots, V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$, each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$'s complement) $G^c = (V, E^c)$ complete graph on $V$
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1, V_2, \ldots, V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$’s complement) $G^c=(V,E^c)$ complete graph on $V$
  $\rightarrow$ $|E^c\setminus E|$ time slots
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V, E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1, V_2, ..., V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V, E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$'s complement) $G^c=(V, E^c)$ complete graph on $V$$\rightarrow$ $|E^c\setminus E|$ time slots
- For time slot corresponding to $e^c=\{v, w\} \in E^c\setminus E$ we add two 1's to the time slot column: to the airports of $v$ and $w$, all other entries in that column are 0's.
**Theorem**: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.

**Proof**: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1, V_2, \ldots, V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$'s complement) $G^c = (V,E^c)$ complete graph on $V$
  $\rightarrow$ $|E^c \setminus E|$ time slots
- For time slot corresponding to $e^c = \{v,w\} \in E^c \setminus E$ we add two 1's to the time slot column: to the airports of $v$ and $w$, all other entries in that column are 0's.

Any solution to FRAMA with $\Delta = 0$ and MAP=3 groups the airports (vertices) into triples, such that there are no conflicts between any of the three airports in a triple, that is, such that there is an edge between any of the three vertices in the triple.
**Theorem:** FRAMA is NP-complete, even if $\Delta=0$ and MAP=3.

Proof: Reduction from **Partition into Triangles (PIT)**

- **Graph $G=(V,E)$** (of maximum degree four)
- Can $V$ be partitioned into triples $V_1,V_2,…,V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G=(V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$'s complement) $G^c=(V,E^c)$ complete graph on $V$
  $\rightarrow$ $|E^c\setminus E|$ time slots
- For time slot corresponding to $e^c=\{v,w\}\in E^c\setminus E$ we add two 1's to the time slot column: to the airports of $v$ and $w$, all other entries in that column are 0's.

Any solution to FRAMA with $\Delta=0$ and MAP=3 groups the airports (vertices) into triples, such that there are no conflicts between any of the three airports in a triple, that is, such that there is an edge between any of the three vertices in the triple.
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1,V_2,\ldots,V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$’s complement) $G^c = (V,E^c)$ complete graph on $V$ $\rightarrow$ $|E^c\setminus E|$ time slots
- For time slot corresponding to $e^c = \{v,w\} \in E^c\setminus E$ we add two 1’s to the time slot column: to the airports of $v$ and $w$, all other entries in that column are 0’s.

Any solution to FRAMA with $\Delta = 0$ and MAP=3 groups the airports (vertices) into triples, such that there are no conflicts between any of the three airports in a triple, that is, such that there is an edge between any of the three vertices in the triple.
Theorem: FRAMA is NP-complete, even if $\Delta = 0$ and $\text{MAP} = 3$.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1, V_2, \ldots, V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$ (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$'s complement) $G^c=(V,E^c)$ complete graph on $V$
  $\rightarrow$ $|E^c\setminus E|$ time slots
- For time slot corresponding to $e^c=\{v,w\} \in E^c\setminus E$ we add two 1's to the time slot column: to the airports of $v$ and $w$, all other entries in that column are 0's.

Any solution to FRAMA with $\Delta = 0$ and $\text{MAP} = 3$ groups the airports (vertices) into triples, such that there are no conflicts between any of the three airports in a triple, that is, such that there is an edge between any of the three vertices in the triple.

$\Rightarrow$ We would obtain a solution to PIT
Theorem: FRAMA is NP-complete, even if $\Delta=0$ and MAP=3.

Proof: Reduction from Partition into Triangles (PIT)

- Graph $G = (V,E)$ (of maximum degree four)
- Can $V$ be partitioned into triples $V_1, V_2, \ldots, V_{|V|/3}$, such that each $V_i$ forms a triangle in $G$
  (for each triple of vertices $V_i$ each vertex in $V_i$ is connected to both other vertices in $V_i$)?

Given an instance of PIT (graph $G = (V,E)$ with max degree four) we construct the matrix $F$, the input of FRAMA:

- One airport per vertex $\rightarrow$ $F$ has $|V|$ rows
- Time slot per non-existing edge in $G$ (that is per edge in the $G$'s complement)
  $G^c=(V,E^c)$ complete graph on $V$
  $\rightarrow |E^c\setminus E|$ time slots
- For time slot corresponding to $e^c=\{v,w\} \in E^c\setminus E$ we add two 1’s to the time slot column: to the airports of $v$ and $w$, all other entries in that column are 0’s.

Any solution to FRAMA with $\Delta=0$ and MAP=3 groups the airports (vertices) into triples, such that there are no conflicts between any of the three airports in a triple, that is, such that there is an edge between any of the three vertices in the triple.

$\Rightarrow$ We would obtain a solution to PIT

Solution to FRAMA with $\Delta=0$ (and, thus, $S=0$) and MAP=3 can be verified in polytime.
Theorem: For $\Delta = 0$ and MAP=2 Minimizing the number of modules is equivalent to finding a maximum matching in the *airport conflict graph* (vertex for every airport and an edge between two airports if they can be put into the same module).
Theorem: For $\Delta = 0$ and MAP=2 Minimizing the number of modules is equivalent to finding a maximum matching in the *airport conflict graph* (vertex for every airport and an edge between two airports if they can be put into the same module).

Maximum matching can be found in polynomial time.
Theorem: For $\Delta = 0$ and MAP=2 Minimizing the number of modules is equivalent to finding a maximum matching in the airport conflict graph (vertex for every airport and an edge between two airports if they can be put into the same module).

Maximum matching can be found in polynomial time.
Theorem: For $\Delta = 0$ and $\text{MAP}=2$ Minimizing the number of modules is equivalent to finding a maximum matching in the *airport conflict graph* (vertex for every airport and an edge between two airports if they can be put into the same module).

Maximum matching can be found in polynomial time.

No edge, as AP1 and AP2 are in conflict
Theorem: For $\Delta = 0$ and MAP=2 Minimizing the number of modules is equivalent to finding a maximum matching in the *airport conflict graph* (vertex for every airport and an edge between two airports if they can be put into the same module).

Maximum matching can be found in polynomial time.

No edge, as AP1 and AP2 are in conflict
Theorem: For $\Delta = 0$ and MAP=2 Minimizing the number of modules is equivalent to finding a maximum matching in the *airport conflict graph* (vertex for every airport and an edge between two airports if they can be put into the same module).

Maximum matching can be found in polynomial time.

No edge, as AP1 and AP2 are in conflict
Theorem: For $\Delta=0$ and $\text{MAP}=2$ Minimizing the number of modules is equivalent to finding a maximum matching in the *airport conflict graph* (vertex for every airport and an edge between two airports if they can be put into the same module).

Maximum matching can be found in polynomial time.

No edge, as AP1 and AP2 are in conflict
Theorem: For $\Delta = 0$ and MAP=2 Minimizing the number of modules is equivalent to finding a maximum matching in the *airport conflict graph* (vertex for every airport and an edge between two airports if they can be put into the same module).

Maximum matching can be found in polynomial time.

No edge, as AP1 and AP2 are in conflict
Complexity for $\Delta > 0$ and MAP=2 unknown.
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
• First remove all conflicts
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:

- First remove all conflicts
- Then assign airports to RTMs
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:

- First remove all conflicts
- Then assign airports to RTMs

⇒ Solve rescheduling and assignment problem separately

Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)
Complexity for $\Delta > 0$ and MAP=2 unknown.
Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs
➡ Solve rescheduling and assignment problem separately
Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)
➡ How to deconflict flight schedule?
Complexity for $\Delta > 0$ and $\text{MAP}=2$ unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

- Solve rescheduling and assignment problem separately
  Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)
- How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
• First remove all conflicts
• Then assign airports to RTMs

➡ Solve rescheduling and assignment problem separately

Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

➡ How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
• Bipartite graph: all flights in one part and all slots in the other part
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

⇒ Solve rescheduling and assignment problem separately
Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

⇒ How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
**Complexity for $\Delta > 0$ and MAP=2 unknown.**

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

➡ Solve rescheduling and assignment problem separately

Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

➡ How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
Complexity for $\Delta > 0$ and $\text{MAP}=2$ unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

- Solve rescheduling and assignment problem separately
  Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

- How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
  - $0$, for edge between flight $f$ and its original slot (black edges)
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

➡ Solve rescheduling and assignment problem separately
  Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

➡ How to deconflict flight schedule?
  We can reduce deconfliction problem to matching:
  - Bipartite graph: all flights in one part and all slots in the other part
  - Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
  - Edge weight:
    - 0, for edge between flight $f$ and its original slot (black edges)
    - 1, otherwise (gray edges)
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:

- First remove all conflicts
- Then assign airports to RTMs

⇒ Solve rescheduling and assignment problem separately

Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

⇒ How to deconflict flight schedule?

We can reduce deconfliction problem to matching:

- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
  - 0, for edge between flight $f$ and its original slot (black edges)
  - 1, otherwise (gray edges)
- Find the minimum-weight matching in the graph that matches all flights
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

⇒ Solve rescheduling and assignment problem separately

Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

⇒ How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
  - 0, for edge between flight $f$ and its original slot (black edges)
  - 1, otherwise (gray edges)
- Find the minimum-weight matching in the graph that matches all flights
- If no such matching exists, $\Delta$ must be increased
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

⇒ Solve rescheduling and assignment problem separately

Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

⇒ How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
  - 0, for edge between flight $f$ and its original slot (black edges)
  - 1, otherwise (gray edges)
- Find the minimum-weight matching in the graph that matches all flights
- If no such matching exists, $\Delta$ must be increased
- We can also minimize the total amount of shifted minutes: set the weight of each edge equal to the length of the shift
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs
  - Solve rescheduling and assignment problem separately
  - Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)
  - How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
  - 0, for edge between flight $f$ and its original slot (black edges)
  - 1, otherwise (gray edges)
- Find the minimum-weight matching in the graph that matches all flights
- If no such matching exists, $\Delta$ must be increased
- We can also minimize the total amount of shifted minutes: set the weight of each edge equal to the length of the shift

Runs in polynomial time, but may find suboptimal solutions to FRAMA (not necessary to remove all the conflicts)
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

Solve rescheduling and assignment problem separately
Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

How to deconflict flight schedule?
We can reduce deconfliction problem to matching:
- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
  - 0, for edge between flight $f$ and its original slot (black edges)
  - 1, otherwise (gray edges)
- Find the minimum-weight matching in the graph that matches all flights
- If no such matching exists, $\Delta$ must be increased
- We can also minimize the total amount of shifted minutes: set the weight of each edge equal to the length of the shift

Runs in polynomial time, but may find suboptimal solutions to FRAMA (not necessary to remove all the conflicts)
For a small number of airports: enumerate all pairs of airports
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:

- First remove all conflicts
- Then assign airports to RTMs

solve rescheduling and assignment problem separately

Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

How to deconflict flight schedule?

We can reduce deconfliction problem to matching:

- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
  - 0, for edge between flight $f$ and its original slot (black edges)
  - 1, otherwise (gray edges)

Find the minimum-weight matching in the graph that matches all flights

- If no such matching exists, $\Delta$ must be increased
- We can also minimize the total amount of shifted minutes: set the weight of each edge equal to the length of the shift

Runs in polynomial time, but may find suboptimal solutions to FRAMA (not necessary to remove all the conflicts)

For a small number of airports: enumerate all pairs of airports completely eliminate all conflicts for the given pairs (matching) with a given $\Delta > 0$
Complexity for $\Delta > 0$ and MAP=2 unknown.

Possible heuristic:
- First remove all conflicts
- Then assign airports to RTMs

⇒ Solve rescheduling and assignment problem separately
  Assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module)

⇒ How to deconflict flight schedule?

We can reduce deconfliction problem to matching:
- Bipartite graph: all flights in one part and all slots in the other part
- Flight $f$ is connected to all slots within distance $\Delta/5$ from its original slot
- Edge weight:
  - 0, for edge between flight $f$ and its original slot (black edges)
  - 1, otherwise (gray edges)
- Find the minimum-weight matching in the graph that matches all flights
- If no such matching exists, $\Delta$ must be increased
- We can also minimize the total amount of shifted minutes: set the weight of each edge equal to the length of the shift

Runs in polynomial time, but may find suboptimal solutions to FRAMA (not necessary to remove all the conflicts)

For a small number of airports: enumerate all pairs of airports completely eliminate all conflicts for the given pairs (matching) with a given $\Delta > 0$ chose combination with minimum possible number of modules
IP for FRAMA
Decision variables

\( x_{am} \): airport \( a \) assigned to module \( m \)
\( z_m \): module \( m \) is used
\( y_{atf} \): flight \( f \) arrives/departs at/from airport \( a \) in time slot \( t \)
\( w_{ab} \): conflict between airport \( a \) and airport \( b \) (some \( t \))

\( A \) = set of airports
\( M \) = set of modules
\( T \) = set of time slots
\( V_a \) = flights at airport \( a \)
\( p_{atf} \) = cost to move flight \( f \) at airport \( a \) to time slot \( t \)
\( s_{af} \) = scheduled time for flight \( f \) at airport \( a \)
\( \delta \) = maximum shift distance for scheduled aircraft in terms of time slots: \( \delta = \Delta / 5 \).
Decision variables

\( x_{am} \): airport \( a \) assigned to module \( m \)

\( z_m \): module \( m \) is used

\( y_{atf} \): flight \( f \) arrives/departs at/from airport \( a \) in time slot \( t \)

\( w_{ab} \): conflict between airport \( a \) and airport \( b \) (some \( t \))

\[
\text{min # shifts: } p_{atf} = 1 \text{ if } t \neq s_{af}; \quad p_{atf} = 0 \text{ if } t = s_{af}
\]

\[
\text{min total amount of shifts: } p_{atf} = |t - s_{af}|
\]

\( A \) = set of airports

\( M \) = set of modules

\( T \) = set of time slots

\( V_a \) = flights at airport \( a \)

\( p_{atf} \) = cost to move flight \( f \) at airport \( a \) to time slot \( t \)

\( s_{af} \) = scheduled time for flight \( f \) at airport \( a \)

\( \delta \) = maximum shift distance for scheduled aircraft in terms of time slots: \( \delta = \Delta / 5 \).
Decision variables

\( x_{am} \): airport \( a \) assigned to module \( m \)

\( z_m \): module \( m \) is used

\( y_{atf} \): flight \( f \) arrives/departs at/from airport \( a \) in time slot \( t \)

\( w_{ab} \): conflict between airport \( a \) and airport \( b \) (some \( t \))

\[
\begin{align*}
\min & \text{# shifts: } p_{atf} = 1 \text{ if } t \neq s_{af}; \quad p_{atf} = 0 \text{ if } t = s_{af} \\
\min & \text{total amount of shifts: } p_{atf} = |t - s_{af}| \\
\min & c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t.} & \quad x_{am} \leq z_m \quad \forall (a, m) \in A \times M \quad (2) \\
\quad & \quad \sum_{m \in M} x_{am} = 1 \quad \forall a \in A \quad (3) \\
\quad & \quad \sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T \quad (4) \\
\quad & \quad \sum_{t = \max(1, s_{af} - \delta)}^{\min(|T|, s_{af} + \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a \quad (5) \\
\quad & \quad \sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b \quad (6) \\
\quad & \quad x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b \quad (7) \\
\quad & \quad \sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M \quad (8) \\
\quad & \quad x, y, w, z \quad \text{binary} \quad (9)
\end{align*}
\]

\( A \) = set of airports

\( M \) = set of modules

\( T \) = set of time slots

\( V_a \) = flights at airport \( a \)

\( p_{atf} \) = cost to move flight \( f \) at airport \( a \) to time slot \( t \)

\( s_{af} \) = scheduled time for flight \( f \) at airport \( a \)

\( \delta \) = maximum shift distance for scheduled aircraft in terms of time slots: \( \delta = \Delta / 5. \)
Decision variables

\( x_{am} \): airport \( a \) assigned to module \( m \)
\( z_m \): module \( m \) is used
\( y_{atf} \): flight \( f \) arrives/departs at/from airport \( a \) in time slot \( t \)
\( w_{ab} \): conflict between airport \( a \) and airport \( b \) (some \( t \))

\[
\begin{align*}
\min \text{ # shifts: } p_{atf} &= 1 \text{ if } t \neq s_{af} ; \quad p_{atf} = 0 \text{ if } t = s_{af} \\
\min \text{ total amount of shifts: } p_{atf} &= |t - s_{af}| \\
\end{align*}
\]

\[
\begin{align*}
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t. } x_{am} & \leq z_m \quad \forall (a, m) \in A \times M \\
\sum_{m \in M} x_{am} & = 1 \quad \forall a \in A \\
\sum_{f \in V_a} y_{atf} & \leq 1 \quad \forall (a, t) \in A \times T \\
\min(|T|, s_{af} + \delta) & \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a \\
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} & \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b \\
x_{am} + x_{bm} & \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b \\
\sum_{a \in A} x_{am} & \leq \text{MAP} \quad \forall m \in M \\
x, y, w, z & \quad \text{binary} \\
\end{align*}
\]
Decision variables

- $x_{am}$: airport $a$ assigned to module $m$
- $z_m$: module $m$ is used
- $y_{atf}$: flight $f$ arrives/departs at/from airport $a$ in time slot $t$
- $w_{ab}$: conflict between airport $a$ and airport $b$ (some $t$)

To minimize:

1. Number of shifts: $p_{atf}=1$ if $t \neq s_{af}$; $p_{atf}=0$ if $t = s_{af}$
2. Total amount of shifts: $p_{atf}=|t-s_{af}|$

$$\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf}$$

Subject to:

1. $x_{am} \leq z_m \quad \forall (a, m) \in A \times M$
2. $\sum_{m \in M} x_{am} = 1 \quad \forall a \in A$
3. $\sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T$
4. $\min(|T|, s_{af} + \delta) \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a$
5. $\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{bef} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b$
6. $x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b$
7. $\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M$
8. $x, y, w, z \text{ binary}$

$A = \text{set of airports}$
$M = \text{set of modules}$
$T = \text{set of time slots}$
$V_a = \text{flights at airport } a$
$p_{atf} = \text{cost to move flight } f \text{ at airport } a \text{ to time slot } t$
$s_{af} = \text{scheduled time for flight } f \text{ at airport } a$
$\delta = \text{maximum shift distance for scheduled aircraft in terms of time slots}$

(1) $c_1 \# \text{modules} + c_2 \sum \text{of shifts}$

Some airport assigned to module $m$
**Decision variables**

- $x_{am}$: airport $a$ assigned to module $m$
- $z_m$: module $m$ is used
- $y_{atf}$: flight $f$ arrives/departs at/from airport $a$ in time slot $t$
- $w_{ab}$: conflict between airport $a$ and airport $b$ (some $t$)

**Objective Functions**

- min # shifts: $p_{atf} = 1$ if $t \neq s_{af}$; $p_{atf} = 0$ if $t = s_{af}$
- min total amount of shifts: $p_{atf} = |t-s_{af}|$

\[
\begin{align*}
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf}
\end{align*}
\]

**Constraints**

- $x_{am}$: airport $a$ assigned to module $m$
  \[
  \sum_{m \in M} x_{am} \leq z_m \quad \forall (a, m) \in A \times M
  \]
  \[
  \sum_{m \in M} x_{am} = 1 \quad \forall a \in A
  \]
- $y_{atf}$: flight $f$ arrives/departs at/from airport $a$ in time slot $t$
  \[
  \sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T
  \]
- $\min(|T|, s_{af} + \delta)$
  \[
  \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a
  \]
- $\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b$
- $x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b$
- $\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M$
- $x, y, w, z$ binary

A = set of airports
M = set of modules
T = set of time slots
$V_a$= flights at airport $a$
$p_{atf}$ = cost to move flight $f$ at airport $a$ to time slot $t$
$s_{af}$ = scheduled time for flight $f$ at airport $a$
$\delta$ maximum shift distance for scheduled aircraft in terms of time slots: $\delta = \Delta / 5$. 

1. $c_1 \# \text{modules} + c_2 \sum \text{of shifts}$
2. Some airport assigned to module $m$ implies $\rightarrow$ module $m$ used
3. $\min \# \text{shifts}$
4. $\min \text{total amount}$
5. $\min(|T|, s_{af} + \delta)$
6. $\sum y_{atf} + \sum y_{btf} \leq 1 + w_{ab}$
7. $x_{am} + x_{bm} \leq 2 - w_{ab}$
8. $\sum x_{am} \leq \text{MAP}$
9. $x, y, w, z$ binary
Decision variables
\( x_{am} \): airport \( a \) assigned to module \( m \)
\( z_m \): module \( m \) is used
\( y_{atf} \): flight \( f \) arrives/departs at/from airport \( a \) in time slot \( t \)
\( w_{ab} \): conflict between airport \( a \) and airport \( b \) (some \( t \))

\[
\min \text{# shifts: } p_{atf} = 1 \text{ if } t \neq s_{af}; \quad p_{atf} = 0 \text{ if } t = s_{af} \\
\min \text{ total amount of shifts: } p_{atf} = |t - s_{af}|
\]

\[
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf}
\]

\[
s.t. \quad x_{am} \leq z_m \quad \forall (a, m) \in A \times M \\
\sum_{m \in M} x_{am} = 1 \quad \forall a \in A \\
\sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T \\
\min(|T|, s_{af} + \delta) \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a
\]

\[
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a \prec b
\]

\[
x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a \prec b
\]

\[
\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M
\]

\[
x, y, w, z \quad \text{binary}
\]

A = set of airports
M = set of modules
T = set of time slots
V_a = flights at airport \( a \)
p_{atf} = cost to move flight \( f \) at airport \( a \) to time slot \( t \)
s_{af} = scheduled time for flight \( f \) at airport \( a \)
\( \delta \) maximum shift distance for scheduled aircraft in terms of time slots: \( \delta = \Delta / 5 \).

(1) \( c_1 \# \text{ modules} + c_2 \# \text{ shifts} \)
(2) \( \rightarrow \text{module } m \text{ used} \)
(3) Each airport assigned to 1 module

Some airport assigned to module \( m \)
Decision variables

\( x_{am} \): airport \( a \) assigned to module \( m \)

\( z_m \): module \( m \) is used

\( y_{atf} \): flight \( f \) arrives/departs at/from airport \( a \) in time slot \( t \)

\( w_{ab} \): conflict between airport \( a \) and airport \( b \) (some \( t \))

\[
\min \# \text{ shifts}: p_{atf} = 1 \text{ if } t \neq s_{af}; \quad p_{atf} = 0 \text{ if } t = s_{af}
\]

\[
\min \text{ total amount of shifts}: p_{atf} = |t - s_{af}|
\]

\[
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf}
\]

\[
\text{s.t. } x_{am} \leq z_m \quad \forall (a, m) \in A \times M
\]

\[
\sum_{m \in M} x_{am} = 1 \quad \forall a \in A
\]

\[
\sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T
\]

\[
\min(|T|, s_{af} + \delta)
\]

\[
\sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a
\]

\[
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b
\]

\[
x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b
\]

\[
\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M
\]

\[
x, y, w, z \quad \text{binary}
\]

A = set of airports

M = set of modules

T = set of time slots

\( V_a \) = flights at airport \( a \)

\( p_{atf} \) = cost to move flight \( f \) at airport \( a \) to time slot \( t \)

\( s_{af} \) = scheduled time for flight \( f \) at airport \( a \)

\( \delta \) = maximum shift distance for scheduled aircraft in terms of time slots: \( \delta = \Delta/5 \).

1. \( c_1 \# \text{modules} + c_2 \sum \text{ of shifts} \)
2. \( \rightarrow \text{module } m \text{ used} \)
3. Each airport assigned to 1 module
4. At most 1 flight arrives/departs at airport time slot \( t \)
Decision variables
\(x_{am}:\) airport \(a\) assigned to module \(m\)
\(z_m:\) module \(m\) is used
\(y_{atf}:\) flight \(f\) arrives/departs at/from airport \(a\) in time slot \(t\)
\(w_{ab}:\) conflict between airport \(a\) and airport \(b\) (some \(t\))

\[
\min \# \text{ shifts: } p_{atf} = 1 \text{ if } t \neq s_{af}; \quad p_{atf} = 0 \text{ if } t = s_{af}
\]
\[
\min \text{ total amount of shifts: } p_{atf} = |t - s_{af}|
\]

\[
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf}
\]

s.t. \(x_{am} \leq z_m \quad \forall (a, m) \in A \times M\)
\[
\sum_{m \in M} x_{am} = 1 \quad \forall a \in A
\]
\[
\sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T
\]
\[
\min(|T|, s_{af} + \delta) \quad \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a
\]
\[
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \quad \forall (a, b, t) \in A \times A \times T, a < b
\]
\[
x_{am} + x_{bm} \leq 2 - w_{ab} \quad \forall (a, b, m) \in A \times A \times M, a < b
\]
\[
\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M
\]
\[
x, y, w, z \quad \text{binary}
\]

A = set of airports
\(M = \) set of modules
\(T = \) set of time slots
\(V_a = \) flights at airport \(a\)
\(p_{atf} = \) cost to move flight \(f\) at airport \(a\) to time slot \(t\)
\(s_{af} = \) scheduled time for flight \(f\) at airport \(a\)
\(\delta = \) maximum shift distance for scheduled aircraft in terms of time slots: \(\delta = \Delta / 5\).
Decision variables

\( x_{am} \): airport \( a \) assigned to module \( m \)

\( z_m \): module \( m \) is used

\( y_{atf} \): flight \( f \) arrives/departs at/from airport \( a \) in time slot \( t \)

\( w_{ab} \): conflict between airport \( a \) and airport \( b \) (some \( t \))

min # shifts: \( p_{atf} = 1 \) if \( t \neq s_{af} \); \( p_{atf} = 0 \) if \( t = s_{af} \)

min total amount of shifts: \( p_{atf} = \vert t - s_{af} \vert \)

\[
\begin{align*}
\text{min } & c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t. } & x_{am} \leq z_m \quad \forall (a, m) \in A \times M \\
& \sum_{m \in M} x_{am} = 1 \quad \forall a \in A \\
& \sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T \\
& \min(\vert T \vert, s_{af} + \delta) \\
& \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a \\
& \sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b \\
& x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b \\
& \sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M \\
& x, y, w, z \quad \text{binary}
\end{align*}
\]

A = set of airports
M = set of modules
T = set of time slots
\( V_a \) = flights at airport \( a \)
\( p_{atf} \) = cost to move flight \( f \) at airport \( a \) to time slot \( t \)
\( s_{af} \) = scheduled time for flight \( f \) at airport \( a \)
\( \delta \) = maximum shift distance for scheduled aircraft in terms of time slots: \( \delta = \Delta / 5 \).

(1) \( c_1 \times \text{#modules} + c_2 \times \text{sum of shifts} \)

Some airport assigned to module \( m \)

(2) \( \rightarrow \) module \( m \) used

Each airport assigned to 1 module

(3) Each airport assigned to 1 module

At most 1 flight arrives/departs at airport \( a \) time slot \( t \)

(4) Each flight \( \pm \delta \) from scheduled time

(5) Two a/c at same slot at airports \( a \) and \( b \)

(6) Two a/c at same slot at airports \( a \) and \( b \)
Decision variables
\(x_{am}:\) airport \(a\) assigned to module \(m\)
\(z_m:\) module \(m\) is used
\(y_{atf}:\) flight \(f\) arrives/departs at/from airport \(a\) in time slot \(t\)
\(w_{ab}:\) conflict between airport \(a\) and airport \(b\) (some \(t\))

\[
\begin{align*}
\text{min } \# \text{ shifts: } p_{atf} &= 1 \quad \text{if } t \neq s_{af}, \quad p_{atf} = 0 \quad \text{if } t = s_{af} \\
\text{min total amount of shifts: } p_{atf} &= |t - s_{af}|
\end{align*}
\]

\[
\begin{align*}
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf}
\end{align*}
\]

s.t. \(x_{am}\)
\[
\sum_{m \in M} x_{am} \leq z_m \quad \forall (a, m) \in A \times M
\]
\[
\sum_{m \in M} x_{am} = 1 \quad \forall a \in A
\]
\[
\sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T
\]
\[
\min (|T|, s_{af} + \delta)
\]
\[
\sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a
\]
\[
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b
\]
\[
x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b
\]
\[
\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M
\]
\[
x, y, w, z \quad \text{binary}
\]

A = set of airports
M = set of modules
T = set of time slots
\(V_a\) = flights at airport \(a\)
\(p_{atf}\) = cost to move flight \(f\) at airport \(a\) to time slot \(t\)
\(s_{af}\) = scheduled time for flight \(f\) at airport \(a\)
\(\delta\) = maximum shift distance for scheduled aircraft in terms of time slots: \(\delta = \Delta / 5\).

(1) \(c_1\)\#modules + \(c_2\) sum of shifts

Some airport assigned to module \(m\)
(2) \(\rightarrow\) module \(m\) used
(3) Each airport assigned to 1 module
At most 1 flight arrives/departs at airport time slot \(t\)
(4) Each flight \(\pm \delta\) from scheduled time
(5) Two a/c at same slot at airports \(a\) and \(b\) \(\rightarrow\) two airports in conflict
(6) Two a/c at same slot at airports \(a\) and \(b\)
Decision variables

\( x_{am} \): airport \( a \) assigned to module \( m \)

\( z_m \): module \( m \) is used

\( y_{atf} \): flight \( f \) arrives/departs at/from airport \( a \) in time slot \( t \)

\( w_{ab} \): conflict between airport \( a \) and airport \( b \) (some \( t \))

\[
\begin{align*}
\text{min} \ # \text{ shifts: } & \quad p_{atf} = 1 \text{ if } t \neq s_{af}; \quad p_{atf} = 0 \text{ if } t = s_{af} \\
\text{min total amount of shifts: } & \quad p_{atf} = |t - s_{af}|
\end{align*}
\]

\[
\begin{align*}
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t. } x_{am} & \leq z_m \quad \forall (a, m) \in A \times M \\
\sum_{m \in M} x_{am} & = 1 \quad \forall a \in A \\
\sum_{f \in V_a} y_{atf} & \leq 1 \quad \forall (a, t) \in A \times T \\
\min (|T|, s_{af} + \delta) & \quad \forall (a, f) \in A \times V_a \\
\sum_{t = \max(1, s_{af} - \delta)} y_{atf} & = 1 \quad \forall (a, f) \in A \times V_a \\
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} & \leq 1 + w_{ab} \quad \forall (a, b, t) \in A \times A \times T, a < b \\
x_{am} + x_{bm} & \leq 2 - w_{ab} \quad \forall (a, b, m) \in A \times A \times M, a < b \\
\sum_{a \in A} x_{am} & \leq \text{MAP} \quad \forall m \in M \\
x, y, w, z & \text{ binary}
\end{align*}
\]

\( A \) = set of airports

\( M \) = set of modules

\( T \) = set of time slots

\( V_a \) = flights at airport \( a \)

\( p_{atf} \) = cost to move flight \( f \) at airport \( a \) to time slot \( t \)

\( s_{af} \) = scheduled time for flight \( f \) at airport \( a \)

\( \delta \) = maximum shift distance for scheduled aircraft in terms of time slots: \( \delta = \Delta / 5. \)

(1) \( c_1 \# \text{modules} + c_2 \text{ sum of shifts} \)

(2) \( \rightarrow \text{module } m \) used

(3) Each airport assigned to 1 module

(4) At most 1 flight arrives/departs at airport time slot \( t \)

(5) Each flight \( \pm \delta \) from scheduled time

(6) Two a/c at same slot at airports \( a \) and \( b \) \( \rightarrow \) two airports in conflict

(7) If \( \exists \) conflict \( \rightarrow \) airports not same module

(8)

(9)
Decision variables

- $x_{am}$: airport $a$ assigned to module $m$
- $z_m$: module $m$ is used
- $y_{atf}$: flight $f$ arrives/departs at/from airport $a$ in time slot $t$
- $w_{ab}$: conflict between airport $a$ and airport $b$ (some $t$)

\[ \min \text{# shifts: } p_{atf} = \begin{cases} 1 & \text{if } t \neq s_{af} \\ 0 & \text{if } t = s_{af} \end{cases} \]

\[ \min \text{ total amount of shifts: } p_{atf} = |t - s_{af}| \]

\[ \min \sum_{m \in M} c_1 z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \]

s.t.

\[ x_{am} \leq z_m \quad \forall (a, m) \in A \times M \]

\[ \sum_{m \in M} x_{am} = 1 \quad \forall a \in A \]

\[ \sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T \]

\[ \min(|T|, s_{af} + \delta) \leq \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a \]

\[ \sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \quad \forall (a, b, t) \in A \times A \times T, a < b \]

\[ x_{am} + x_{bm} \leq 2 - w_{ab} \quad \forall (a, b, m) \in A \times A \times M, a < b \]

\[ \sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M \]

\[ x, y, w, z \text{ binary} \]

A = set of airports
M = set of modules
T = set of time slots
$V_a$ = flights at airport $a$
$p_{atf}$ = cost to move flight $f$ at airport $a$ to time slot $t$
s$_{af}$ = scheduled time for flight $f$ at airport $a$
\[ \delta \] = maximum shift distance for scheduled aircraft in terms of time slots: $\delta = \Delta / 5$.  

(1) $c_1 \times \text{#modules} + c_2 \times \text{sum of shifts}$

(2) Some airport assigned to module $m$ → module $m$ used

(3) Each airport assigned to 1 module

(4) At most 1 flight arrives/departs at airport $a$ in time slot $t$

(5) Each flight $\pm \delta$ from scheduled time

(6) Two a/c at same slot at airports $a$ and $b$ → two airports in conflict

(7) If $\exists$ conflict → airports not same module

(8) Max MAP airports to each module

(9)
\[
\begin{align*}
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t. } x_{am} & \leq z_m & \forall (a, m) \in A \times M \\
\sum_{m \in M} x_{am} & = 1 & \forall a \in A \\
\sum_{f \in V_a} y_{atf} & \leq 1 & \forall (a, t) \in A \times T \\
\min(|T|, s_{af} + \delta) & \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 & \forall (a, f) \in A \times V_a \\
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} & \leq 1 + w_{ab} & \forall (a, b, t) \in A \times A \times T, a < b \\
x_{am} + x_{bm} & \leq 2 - w_{ab} & \forall (a, b, m) \in A \times A \times M, a < b \\
\sum_{a \in A} x_{am} & \leq \text{MAP} & \forall m \in M \\
x, y, w, z & \text{ binary} 
\end{align*}
\]
IP formulation of FRAMA optimises $c_1*M + c_2*S$ (could move one in constraint)

$$
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t. } x_{am} \leq z_m \quad \forall (a, m) \in A \times M \\
\quad = 1 \quad \forall a \in A \\
\quad \leq 1 \quad \forall (a, t) \in A \times T \\
\quad \leq 1 \quad \forall (a, f) \in A \times V_a \\
\quad \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b \\
\quad \leq \text{MAP} \quad \forall m \in M \\
\text{x, y, w, z binary}$$
IP formulation of FRAMA optimises $c_1*M + c_2*S$ (could move one in constraint)
We choose $c_1$ and $c_2$ such that minimizing the modules is the primary objective: $c_1>>c_2$

\[
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \tag{1}
\]

\[
s.t. \quad x_{am} \leq z_m \quad \forall (a, m) \in A \times M \tag{2}
\]

\[
\sum_{m \in M} x_{am} = 1 \quad \forall a \in A \tag{3}
\]

\[
\sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T \tag{4}
\]

\[
\min(|T|, s_{af} + \delta) \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a \tag{5}
\]

\[
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b \tag{6}
\]

\[
x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b \tag{7}
\]

\[
\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M \tag{8}
\]

\[
x, y, w, z \quad \text{binary} \tag{9}
\]
IP formulation of FRAMA optimises \( c_1 \cdot M + c_2 \cdot S \) (could move one in constraint)

We choose \( c_1 \) and \( c_2 \) such that minimizing the modules is the primary objective: \( c_1 \gg c_2 \)

IP computes new slots for flights and assigns airports to RTMs, such that:

\[
\begin{align*}
\min & \quad c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t.} & \quad x_{am} \quad \leq z_m \quad \forall (a, m) \in A \times M \\
& \quad \sum_{m \in M} x_{am} \quad = 1 \quad \forall a \in A \\
& \quad \sum_{f \in V_a} y_{atf} \quad \leq 1 \quad \forall (a, t) \in A \times T \\
& \quad \min(|T|, s_{af} + \delta) \sum_{t = \max(1, s_{af} - \delta)} y_{atf} \quad = 1 \quad \forall (a, f) \in A \times V_a \\
& \quad \sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b \\
& \quad x_{am} + x_{bm} \quad \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b \\
& \quad \sum_{a \in A} x_{am} \quad \leq \text{MAP} \quad \forall m \in M \\
& \quad x, y, w, z \quad \text{binary}
\end{align*}
\]
IP formulation of FRAMA optimises $c_1*M + c_2*S$ (could move one in constraint)
We choose $c_1$ and $c_2$ such that minimizing the modules is the primary objective: $c_1>>c_2$
IP computes new slots for flights and assigns airports to RTMs, such that:

- Each flight is moved by at most $\Delta$

\[
\begin{align*}
\min & \quad c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t.} & \quad x_{am} \leq z_m \quad \forall (a, m) \in A \times M \\
& \quad \sum_{m \in M} x_{am} = 1 \quad \forall a \in A \\
& \quad \sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T \\
& \quad \min(|T|, s_{af} + \delta) \leq \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a \\
& \quad \sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b \\
& \quad x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b \\
& \quad \sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M \\
& \quad x, y, w, z \quad \text{binary}
\end{align*}
\]
IP formulation of FRAMA optimises $c_1*M + c_2*S$ (could move one in constraint)

We choose $c_1$ and $c_2$ such that minimizing the modules is the primary objective: $c_1\gg c_2$

IP computes new slots for flights and assigns airports to RTMs, such that:

- Each flight is moved by at most $\Delta$
- No conflicting airports are assigned to the same RTM

\[
\begin{align*}
\min & \quad c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \\
\text{s.t.} & \quad x_{am} \leq z_m \quad \forall (a, m) \in A \times M \\
& \quad \sum_{m \in M} x_{am} = 1 \quad \forall a \in A \\
& \quad \sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T \\
& \quad \min(|T|, s_{af} + \delta) \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a \\
& \quad \sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \quad \forall (a, b, t) \in A \times A \times T, a < b \\
& \quad x_{am} + x_{bm} \leq 2 - w_{ab} \quad \forall (a, b, m) \in A \times A \times M, a < b \\
& \quad \sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M \\
x, y, w, z & \quad \text{binary}
\end{align*}
\]
IP formulation of FRAMA optimises $c_1^*M + c_2^*S$ (could move one in constraint)

We choose $c_1$ and $c_2$ such that minimizing the modules is the primary objective: $c_1^{>>}c_2$

IP computes new slots for flights and assigns airports to RTMs, such that:

- Each flight is moved by at most $\Delta$
- No conflicting airports are assigned to the same RTM
- At most MAP airports are assigned per module

\[
\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \quad (1)
\]

\[s.t.\]
\[
\sum_{m \in M} x_{am} \leq z_m \quad \forall (a, m) \in A \times M \quad (2)
\]
\[
\sum_{m \in M} x_{am} = 1 \quad \forall a \in A \quad (3)
\]
\[
\sum_{f \in V_a} y_{atf} \leq 1 \quad \forall (a, t) \in A \times T \quad (4)
\]
\[
\min(|T|, s_{af} + \delta) \sum_{t = \max(1, s_{af} - \delta)} y_{atf} = 1 \quad \forall (a, f) \in A \times V_a \quad (5)
\]
\[
\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leq 1 + w_{ab} \quad \forall (a, b, t) \in A \times A \times T, a < b \quad (6)
\]
\[
x_{am} + x_{bm} \leq 2 - w_{ab} \quad \forall (a, b, m) \in A \times A \times M, a < b \quad (7)
\]
\[
\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M \quad (8)
\]
\[
x, y, w, z \quad \text{binary} \quad (9)
\]
IP formulation of FRAMA optimises $c_1*M + c_2*S$ (could move one in constraint)
We choose $c_1$ and $c_2$ such that minimizing the modules is the primary objective: $c_1 >> c_2$
IP computes new slots for flights and assigns airports to RTMs, such that:
• Each flight is moved by at most $\Delta$
• No conflicting airports are assigned to the same RTM
• At most MAP airports are assigned per module

$$\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{a,f} y_{a,f}$$  \hspace{1cm} (1)

s.t. $x_{am}$ $\leq z_m \quad \forall (a, m) \in A \times M$  \hspace{1cm} (2)

$$\sum_{m \in M} x_{am} = 1 \quad \forall a \in A$$  \hspace{1cm} (3)

$$\sum_{f \in V_a} y_{a,f} \leq 1 \quad \forall (a, t) \in A \times T$$  \hspace{1cm} (4)

$$\min(|T|, s_{a,f} + \delta)$$

$$\sum_{t = \max(1, s_{a,f} - \delta)} \sum_{f \in V_a} y_{a,f} = 1 \quad \forall (a, f) \in A \times V_a$$  \hspace{1cm} (5)

$$\sum_{f \in V_a} y_{a,f} + \sum_{f \in V_b} y_{b,f} \leq 1 + w_{ab} \forall (a, b, t) \in A \times A \times T, a < b$$  \hspace{1cm} (6)

$$x_{am} + x_{bm} \leq 2 - w_{ab} \forall (a, b, m) \in A \times A \times M, a < b$$  \hspace{1cm} (7)

$$\sum_{a \in A} x_{am} \leq \text{MAP} \quad \forall m \in M$$  \hspace{1cm} (8)

$x, y, w, z$ binary  \hspace{1cm} (9)

$\Rightarrow$ IP formulation solves FRAMA!
Experimental Study
Additional airports considered for remote operation in Sweden:
Additional airports considered for remote operation in Sweden:

- **Airport 1 (AP1):** Small airport with low traffic, few scheduled flights per hour, non-regular helicopter traffic, sometimes special testing activities.

- **Airport 2 (AP2):** Low to medium-sized airport, multiple scheduled flights per hour, regular special traffic flights (usually open 24/7, with exceptions).

- **Airport 3 (AP3):** Small regional airport with regular scheduled flights (usually open 24/7, with exceptions)

- **Airport 4 (AP4):** Small airport with significant seasonal variations.

- **Airport 5 (AP5):** Small airport with low scheduled traffic, non-regular helicopter flights.
Additional airports considered for remote operation in Sweden:

- **Airport 1 (AP1):** Small airport with low traffic, few scheduled flights per hour, non-regular helicopter traffic, sometimes special testing activities.
- **Airport 2 (AP2):** Low to medium-sized airport, multiple scheduled flights per hour, regular special traffic flights (usually open 24/7, with exceptions).
- **Airport 3 (AP3):** Small regional airport with regular scheduled flights (usually open 24/7, with exceptions)
- **Airport 4 (AP4):** Small airport with significant seasonal variations.
- **Airport 5 (AP5):** Small airport with low scheduled traffic, non-regular helicopter flights.

We use traffic data from October 19, 2016—the day with highest traffic in 2016
Additional airports considered for remote operation in Sweden:

- **Airport 1 (AP1):** Small airport with low traffic, few scheduled flights per hour, non-regular helicopter traffic, sometimes special testing activities.
- **Airport 2 (AP2):** Low to medium-sized airport, multiple scheduled flights per hour, regular special traffic flights (usually open 24/7, with exceptions).
- **Airport 3 (AP3):** Small regional airport with regular scheduled flights (usually open 24/7, with exceptions)
- **Airport 4 (AP4):** Small airport with significant seasonal variations.
- **Airport 5 (AP5):** Small airport with low scheduled traffic, non-regular helicopter flights.

We use traffic data from October 19, 2016—the day with highest traffic in 2016. 286 flight movements were scheduled on this day for the five airports.
Additional airports considered for remote operation in Sweden:

- **Airport 1 (AP1):** Small airport with low traffic, few scheduled flights per hour, non-regular helicopter traffic, sometimes special testing activities.
- **Airport 2 (AP2):** Low to medium-sized airport, multiple scheduled flights per hour, regular special traffic flights (usually open 24/7, with exceptions).
- **Airport 3 (AP3):** Small regional airport with regular scheduled flights (usually open 24/7, with exceptions).
- **Airport 4 (AP4):** Small airport with significant seasonal variations.
- **Airport 5 (AP5):** Small airport with low scheduled traffic, non-regular helicopter flights.

We use traffic data from October 19, 2016—the day with highest traffic in 2016. 286 flight movements were scheduled on this day for the five airports. For first set of experiments: without self-conflicts → 233 movements.
Additional airports considered for remote operation in Sweden:

- **Airport 1 (AP1):** Small airport with low traffic, few scheduled flights per hour, non-regular helicopter traffic, sometimes special testing activities.
- **Airport 2 (AP2):** Low to medium-sized airport, multiple scheduled flights per hour, regular special traffic flights (usually open 24/7, with exceptions).
- **Airport 3 (AP3):** Small regional airport with regular scheduled flights (usually open 24/7, with exceptions)
- **Airport 4 (AP4):** Small airport with significant seasonal variations.
- **Airport 5 (AP5):** Small airport with low scheduled traffic, non-regular helicopter flights.

We use traffic data from October 19, 2016—the day with highest traffic in 2016
286 flight movements were scheduled on this day for the five airports
For first set of experiments: without self-conflicts $\rightarrow$ 233 movements
One optimization problem for each pair ($\Delta$, MAP)
We have $12 \times 24 = 288$ slots for flight movements

⇒ with sufficiently large shifts 233 flight movements in single module
### Original Traffic

**MAP=5**

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts $= S$</th>
<th>maximum shift (in mins) $= \Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>32</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>26</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>118</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>108</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>99</td>
<td>45</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>91</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>85</td>
<td>55</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>83</td>
<td>60</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>81</td>
<td>65</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>79</td>
<td>70</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>78</td>
<td>75</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>75</td>
<td>80</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>75</td>
<td>85</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>75</td>
<td>90</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>74</td>
<td>95</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>74</td>
<td>100</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>73</td>
<td>105</td>
</tr>
</tbody>
</table>

No rescheduling allowed: need 5 RTMs

We have $12 \times 24 = 288$ slots for flight movements

⇒ with sufficiently large shifts 233 flight movements in single module
Original Traffic

MAP=5

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts $= S$</th>
<th>maximum shift (in mins) $= \Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>32</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>26</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>118</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>108</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>99</td>
<td>45</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>91</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>85</td>
<td>55</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>83</td>
<td>60</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>81</td>
<td>65</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>79</td>
<td>70</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>78</td>
<td>75</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>75</td>
<td>80</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>75</td>
<td>85</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>75</td>
<td>90</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>74</td>
<td>95</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>74</td>
<td>100</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>73</td>
<td>105</td>
</tr>
</tbody>
</table>

No rescheduling allowed: need 5 RTMs
Reschedule at most ±5 minutes: 2 RTMs

We have $12 \times 24 = 288$ slots for flight movements

⇒ with sufficiently large shifts 233 flight movements in single module
Original Traffic

MAP=5

<table>
<thead>
<tr>
<th>δ</th>
<th># of modules</th>
<th># of shifts $= S$</th>
<th>maximum shift (in mins) $= \Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>32</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>26</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>118</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>108</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>99</td>
<td>45</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>91</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>85</td>
<td>55</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>83</td>
<td>60</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>81</td>
<td>65</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>79</td>
<td>70</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>78</td>
<td>75</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>75</td>
<td>80</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>75</td>
<td>85</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>75</td>
<td>90</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>74</td>
<td>95</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>74</td>
<td>100</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>73</td>
<td>105</td>
</tr>
</tbody>
</table>

No rescheduling allowed: need 5 RTMs
Reschedule at most $\pm 5$ minutes: 2 RTMs
For 1 RTM: we need to reschedule by $\pm 35$ mins

We have $12 \times 24 = 288$ slots for flight movements
⇒ with sufficiently large shifts 233 flight movements in single module
Original Traffic

Shows tradeoffs: more shifts — larger shifts (more minutes) — more APs/module

#shifts

max shift (in minutes)
### Original Traffic

#### MAP=4

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$M$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>26</td>
</tr>
</tbody>
</table>

#### MAP=3

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$M$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>32</td>
</tr>
</tbody>
</table>

#### MAP=2

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>
All 286 movements

In case of a self-induced conflict: model shifts either of them
All 286 movements

In case of a self-induced conflict: model shifts either of them

➡️ we start with possible more than one flight movement per time slot and airport
All 286 movements

In case of a self-induced conflict: model shifts either of them
- we start with possible more than one flight movement per time slot and airport
- \( \delta = 0 \) infeasible by definition
All 286 movements

In case of a self-induced conflict: model shifts either of them

➡️ we start with possible more than one flight movement per time slot and airport

➡️ \( \delta = 0 \) infeasible by definition

MAP=5

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>( M )</th>
<th>( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
</tr>
</tbody>
</table>
In case of a self-induced conflict: model shifts either of them

- we start with possible more than one flight movement per time slot and airport
- \( \delta = 0 \) infeasible by definition

MAP = 5

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>( M )</th>
<th>( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
</tr>
</tbody>
</table>

For 233 movements 2 RTMs were enough for \( \delta = 1 \), now \( \delta = 2 \)
All 286 movements

In case of a self-induced conflict: model shifts either of them

- we start with possible more than one flight movement per time slot and airport
- \( \delta = 0 \) infeasible by definition

MAP=5

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>( M )</th>
<th>( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
</tr>
</tbody>
</table>

For 233 movs 2 RTMs were enough for \( \delta = 1 \), now \( \delta = 2 \)

For 233 movs 1RTM was enough for \( \delta = 7 \), now \( \delta = 37 \)
All 286 movements

In case of a self-induced conflict: model shifts either of them

- we start with possible more than one flight movement per time slot and airport
- \( \delta = 0 \) infeasible by definition

\[ \text{MAP} = 5 \]

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>( M )</th>
<th>( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
</tr>
</tbody>
</table>

For 233 movs 2 RTMs were enough for \( \delta = 1 \), now \( \delta = 2 \)

\[ \text{MAP} = 4 \]

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>( M )</th>
<th>( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>79</td>
</tr>
</tbody>
</table>
In case of a self-induced conflict: model shifts either of them
➡ we start with possible more than one flight movement per time slot and airport
➡ $\delta=0$ infeasible by definition

MAP=5

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$M$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
</tr>
</tbody>
</table>

For 233 movs 2 RTMs were enough for $\delta=1$, now $\delta=2$

For 233 movs 1RTM was enough for $\delta=7$, now $\delta=37$

MAP=4

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$M$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>79</td>
</tr>
</tbody>
</table>

MAP=3

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$M$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>79</td>
</tr>
</tbody>
</table>
All 286 movements

In case of a self-induced conflict: model shifts either of them:
→ we start with possible more than one flight movement per time slot and airport
→ $\delta=0$ infeasible by definition

MAP=5

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$M$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>infeasible</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
</tr>
</tbody>
</table>

For 233 movs 2 RTMs were enough for $\delta=1$, now $\delta=2$

For 233 movs 1RTM was enough for $\delta=7$, now $\delta=37$

MAP=4

MAP=3

MAP=2
Computation times: Solve in two steps
Computation times: Solve in two steps

We solve two optimisation with $c_2 = 0$ and $c_1 = 0$ respectively and fix the $\Sigma z_k$ to the optimal number of modules used when solving the second optimization problem.
We solve two optimisation with $c_2 = 0$ and $c_1 = 0$ respectively and fix the $\sum z_k$ to the optimal number of modules used when solving the second optimization problem.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts $S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1.40</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1.26</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1.79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
<td>7.97</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
<td>8.42</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
<td>9.34</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>151</td>
<td>40.84</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>149</td>
<td>46.61</td>
</tr>
<tr>
<td>41</td>
<td>1</td>
<td>147</td>
<td>45.12</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>144</td>
<td>38.10</td>
</tr>
<tr>
<td>43</td>
<td>1</td>
<td>141</td>
<td>40.20</td>
</tr>
<tr>
<td>44</td>
<td>1</td>
<td>139</td>
<td>43.57</td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>137</td>
<td>9.24</td>
</tr>
<tr>
<td>46</td>
<td>1</td>
<td>136</td>
<td>106.31</td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>135</td>
<td>148.79</td>
</tr>
<tr>
<td>48</td>
<td>1</td>
<td>134</td>
<td>100.03</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>133</td>
<td>94.08</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>132</td>
<td>479.12</td>
</tr>
<tr>
<td>51</td>
<td>1</td>
<td>130</td>
<td>433.79</td>
</tr>
<tr>
<td>52</td>
<td>1</td>
<td>128</td>
<td>348.83</td>
</tr>
<tr>
<td>53</td>
<td>1</td>
<td>126</td>
<td>11.65</td>
</tr>
<tr>
<td>288</td>
<td>1</td>
<td>126</td>
<td>46.49</td>
</tr>
</tbody>
</table>
We solve two optimisation with $c_2 = 0$ and $c_1 = 0$ respectively and fix the $\sum z_k$ to the optimal number of modules used when solving the second optimization problem.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1,40</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1,26</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1,79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
<td>7,97</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
<td>8,42</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
<td>9,34</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>151</td>
<td>40,84</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>149</td>
<td>46,61</td>
</tr>
<tr>
<td>41</td>
<td>1</td>
<td>147</td>
<td>45,12</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>144</td>
<td>38,10</td>
</tr>
<tr>
<td>43</td>
<td>1</td>
<td>141</td>
<td>40,20</td>
</tr>
<tr>
<td>44</td>
<td>1</td>
<td>139</td>
<td>43,57</td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>137</td>
<td>9,24</td>
</tr>
<tr>
<td>46</td>
<td>1</td>
<td>136</td>
<td>106,31</td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>135</td>
<td>148,79</td>
</tr>
<tr>
<td>48</td>
<td>1</td>
<td>134</td>
<td>100,03</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>133</td>
<td>94,08</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>132</td>
<td>479,12</td>
</tr>
<tr>
<td>51</td>
<td>1</td>
<td>130</td>
<td>433,79</td>
</tr>
<tr>
<td>52</td>
<td>1</td>
<td>128</td>
<td>348,83</td>
</tr>
<tr>
<td>53</td>
<td>1</td>
<td>126</td>
<td>11,65</td>
</tr>
<tr>
<td>288</td>
<td>1</td>
<td>126</td>
<td>46,49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1,31</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1,06</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1,22</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>79</td>
<td>60,92</td>
</tr>
</tbody>
</table>
Computation times: Solve in two steps

We solve two optimisation with $c_2 = 0$ and $c_1 = 0$ respectively and fix the $\Sigma z_k$ to the optimal number of modules used when solving the second optimization problem.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1.40</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1.26</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1.79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
<td>7.97</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
<td>8.42</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
<td>9.34</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>151</td>
<td>40.84</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>149</td>
<td>46.61</td>
</tr>
<tr>
<td>41</td>
<td>1</td>
<td>147</td>
<td>45.12</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>144</td>
<td>38.10</td>
</tr>
<tr>
<td>43</td>
<td>1</td>
<td>141</td>
<td>40.20</td>
</tr>
<tr>
<td>44</td>
<td>1</td>
<td>139</td>
<td>43.57</td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>137</td>
<td>9.24</td>
</tr>
<tr>
<td>46</td>
<td>1</td>
<td>136</td>
<td>106.31</td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>135</td>
<td>148.79</td>
</tr>
<tr>
<td>48</td>
<td>1</td>
<td>134</td>
<td>100.03</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>133</td>
<td>94.08</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>132</td>
<td>479.12</td>
</tr>
<tr>
<td>51</td>
<td>1</td>
<td>130</td>
<td>433.79</td>
</tr>
<tr>
<td>52</td>
<td>1</td>
<td>128</td>
<td>348.83</td>
</tr>
<tr>
<td>53</td>
<td>1</td>
<td>126</td>
<td>11.65</td>
</tr>
<tr>
<td>288</td>
<td>1</td>
<td>126</td>
<td>46.49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1.31</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1.06</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1.22</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>79</td>
<td>60.92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\delta$</th>
<th># of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1.36</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1.28</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1.09</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>79</td>
<td>51.79</td>
</tr>
</tbody>
</table>
We solve two optimisation with $c_2 = 0$ and $c_1 = 0$ respectively and fix the $\sum z_k$ to the optimal number of modules used when solving the second optimization problem.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>MAP=5 # of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1.40</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1.26</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1.79</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>79</td>
<td>7.97</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>158</td>
<td>8.42</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>154</td>
<td>9.34</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>151</td>
<td>40.84</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>149</td>
<td>46.61</td>
</tr>
<tr>
<td>41</td>
<td>1</td>
<td>147</td>
<td>45.12</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>144</td>
<td>38.10</td>
</tr>
<tr>
<td>43</td>
<td>1</td>
<td>141</td>
<td>40.20</td>
</tr>
<tr>
<td>44</td>
<td>1</td>
<td>139</td>
<td>43.57</td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>137</td>
<td>9.24</td>
</tr>
<tr>
<td>46</td>
<td>1</td>
<td>136</td>
<td>106.31</td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>135</td>
<td>148.79</td>
</tr>
<tr>
<td>48</td>
<td>1</td>
<td>134</td>
<td>100.03</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>133</td>
<td>94.08</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>132</td>
<td>479.12</td>
</tr>
<tr>
<td>51</td>
<td>1</td>
<td>130</td>
<td>433.79</td>
</tr>
<tr>
<td>52</td>
<td>1</td>
<td>128</td>
<td>348.83</td>
</tr>
<tr>
<td>53</td>
<td>1</td>
<td>126</td>
<td>11.65</td>
</tr>
<tr>
<td>288</td>
<td>1</td>
<td>126</td>
<td>46.49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>MAP=4 # of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1.31</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1.06</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1.22</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>79</td>
<td>60.92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>MAP=3 # of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>103</td>
<td>1.36</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>80</td>
<td>1.28</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>79</td>
<td>1.09</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>79</td>
<td>51.79</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>MAP=2 # of modules</th>
<th># of shifts $= S$</th>
<th>computation time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>infeasible</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>61</td>
<td>0.55</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>61</td>
<td>1.09</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>60</td>
<td>0.98</td>
</tr>
<tr>
<td>288</td>
<td>3</td>
<td>60</td>
<td>100.30</td>
</tr>
</tbody>
</table>
Increased Traffic Volume

Duplicate each of the original flight movements
Increased Traffic Volume

Duplicate each of the original flight movements
Shift randomly by plus/minus one hour
Increased Traffic Volume

Duplicate each of the original flight movements
Shift randomly by plus/minus one hour
Shift again, randomly, by plus/minus 15 minutes
Increased Traffic Volume

Duplicate each of the original flight movements
Shift randomly by plus/minus one hour
Shift again, randomly, by plus/minus 15 minutes
If two flight movements end up in the same slot, one of the movements is deleted
Increased Traffic Volume

Duplicate each of the original flight movements
Shift randomly by plus/minus one hour
Shift again, randomly, by plus/minus 15 minutes
If two flight movements end up in the same slot, one of the movements is deleted
“2x” data created from all data of the year 2016
Increased Traffic Volume

Duplicate each of the original flight movements
Shift randomly by plus/minus one hour
Shift again, randomly, by plus/minus 15 minutes
If two flight movements end up in the same slot, one of the movements is deleted
“2x” data created from all data of the year 2016
➡ shifted duplicates of flights from October 18, 2016 and October 20, 2016 may now happen on October 19, 2016
Duplicate each of the original flight movements
Shift randomly by plus/minus one hour
Shift again, randomly, by plus/minus 15 minutes
If two flight movements end up in the same slot, one of the movements is deleted
“2x” data created from all data of the year 2016
➡ shifted duplicates of flights from October 18, 2016 and October 20, 2016 may now happen on October 19, 2016
➡ Not exactly twice the number of movements
Duplicate each of the original flight movements
Shift randomly by plus/minus one hour
Shift again, randomly, by plus/minus 15 minutes
If two flight movements end up in the same slot, one of the movements is deleted
“2x” data created from all data of the year 2016
➡ shifted duplicates of flights from October 18, 2016 and October 20, 2016 may now happen on October 19, 2016
➡ Not exactly twice the number of movements
- October 19: data set has 416 flight movements (after deleting double movements in time slots) out of 575 flight movements (all of the movements from 2016 that the duplication and shifting process produces)
### Increased Traffic Volume

<table>
<thead>
<tr>
<th>δ</th>
<th># of modules</th>
<th>S</th>
<th>Δ</th>
<th>S for 3RTMs (1-3AP/RTM)</th>
<th>S for 3RTMs (1-2AP/RTM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>30</td>
<td>5</td>
<td>30</td>
<td>33</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>24</td>
<td>10</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>23</td>
<td>15</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>23</td>
<td>20</td>
<td>-</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>111</td>
<td>25</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>101</td>
<td>30</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>96</td>
<td>35</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>92</td>
<td>40</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>88</td>
<td>45</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>87</td>
<td>50</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>84</td>
<td>55</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>81</td>
<td>60</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>81</td>
<td>65</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>81</td>
<td>70</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>81</td>
<td>75</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>80</td>
<td>80</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

![Graph showing traffic volume data](chart.png)
Increased Traffic Volume

For MAP=2 we get the optimum of 3RTMs for $\delta=1$
33 shifts  $\leftrightarrow$ 7 shifts for original traffic
Increased Traffic Volume

Same tradeoffs: more shifts — larger shifts (more minutes) — more APs/module

<table>
<thead>
<tr>
<th>δ</th>
<th># of modules</th>
<th>S</th>
<th>Δ</th>
<th>S for 3RTMs (1-3AP/RTM)</th>
<th>S for 3RTMs (1-2AP/RTM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>30</td>
<td>5</td>
<td>30</td>
<td>33</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>24</td>
<td>10</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>23</td>
<td>15</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>23</td>
<td>20</td>
<td>-</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>111</td>
<td>25</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>101</td>
<td>30</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>96</td>
<td>35</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>92</td>
<td>40</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>88</td>
<td>45</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>87</td>
<td>50</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>84</td>
<td>55</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>81</td>
<td>60</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>81</td>
<td>65</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>81</td>
<td>70</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>81</td>
<td>75</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>80</td>
<td>80</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

For MAP=2 we get the optimum of 3RTMs for $\delta=1$
33 shifts ↔ 7 shifts for original traffic
Conclusion/Future Work
Conclusion

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC

Future Work
Conclusion

- Optimization problem for remote towers (FRAMA):
  - Shifts flights to other, nearby, slots
  - To minimize the total number of modules in the RTC
- Discussed computational complexity

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports
  ➔ Show applicability of our approach

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports
  ➡ Show applicability of our approach
  ➡ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module

Future Work
Conclusion

- Optimization problem for remote towers (FRAMA):
  - Shifts flights to other, nearby, slots
  - To minimize the total number of modules in the RTC
- Discussed computational complexity
- Presented different solution approaches
- Experiments for IP for five Swedish airports
  ➡ Show applicability of our approach
  ➡ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➡ Minor shifts (few minutes) can significantly reduce necessary number of modules

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports
  ➡ Show applicability of our approach
  ➡ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➡ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➡ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports
  ➤ Show applicability of our approach
  ➤ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➤ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➤ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

• Our conflict definition may be too conservative/precautionary
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports
  ➡ Show applicability of our approach
  ➡ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➡ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➡ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

• Our conflict definition may be too conservative/precautionary
• They cannot be discarded, and will influence staff planning
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC

• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports
  ➤ Show applicability of our approach
  ➤ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➤ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➤ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

• Our conflict definition may be too conservative/precautionary
• They cannot be discarded, and will influence staff planning
  ➤ Continues discussion with operations
Conclusion

- Optimization problem for remote towers (FRAMA):
  - Shifts flights to other, nearby, slots
  - To minimize the total number of modules in the RTC
- Discussed computational complexity
- Presented different solution approaches
- Experiments for IP for five Swedish airports
  ➤ Show applicability of our approach
  ➤ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➤ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➤ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

- Our conflict definition may be too conservative/precautionary
- They cannot be discarded, and will influence staff planning
  ➤ Continues discussion with operations
  ➤ Possibly: distinguish arrival/departures
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports
  ➪ Show applicability of our approach
  ➪ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➪ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➪ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

• Our conflict definition may be too conservative/precautionary
• They cannot be discarded, and will influence staff planning
  ➪ Continues discussion with operations
  ➪ Possibly: distinguish arrival/departures
  ➪ Possibly: consider uncertainty
Conclusion

- Optimization problem for remote towers (FRAMA):
  - Shifts flights to other, nearby, slots
  - To minimize the total number of modules in the RTC
- Discussed computational complexity
- Presented different solution approaches
- Experiments for IP for five Swedish airports
  ➪ Show applicability of our approach
  ➪ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➪ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➪ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

- Our conflict definition may be too conservative/precautionary
- They cannot be discarded, and will influence staff planning
  ➪ Continues discussion with operations
  ➪ Possibly: distinguish arrival/departures
  ➪ Possibly: consider uncertainty
- Computational complexity of FRAMA with $\Delta > 0$ and even MAP=2 is open
Conclusion

- Optimization problem for remote towers (FRAMA):
  - Shifts flights to other, nearby, slots
  - To minimize the total number of modules in the RTC
- Discussed computational complexity
- Presented different solution approaches
- Experiments for IP for five Swedish airports
  ➡ Show applicability of our approach
  ➡ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➡ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➡ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

- Our conflict definition may be too conservative/precautionary
- They cannot be discarded, and will influence staff planning
  ➡ Continues discussion with operations
  ➡ Possibly: distinguish arrival/departures
  ➡ Possibly: consider uncertainty
- Computational complexity of FRAMA with $\Delta > 0$ and even MAP=2 is open
- Currently we do not care which airlines affected by shift (possibly all to a single airline)
Conclusion

• Optimization problem for remote towers (FRAMA):
  • Shifts flights to other, nearby, slots
  • To minimize the total number of modules in the RTC
• Discussed computational complexity
• Presented different solution approaches
• Experiments for IP for five Swedish airports
  ➪ Show applicability of our approach
  ➪ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➪ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➪ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

• Our conflict definition may be too conservative/precautionary
• They cannot be discarded, and will influence staff planning
  ➪ Continues discussion with operations
  ➪ Possibly: distinguish arrival/departures
  ➪ Possibly: consider uncertainty
• Computational complexity of FRAMA with $\Delta > 0$ and even MAP=2 is open
• Currently we do not care which airlines affected by shift (possibly all to a single airline)
  ➪ Take equity into account (2 airlines, airline A operating 150 flights, airline B operating 75; reassign slot for 60 flights→ aim for 40 new slots for airline A, 20 new slots for airline B)
Conclusion

- Optimization problem for remote towers (FRAMA):
  - Shifts flights to other, nearby, slots
  - To minimize the total number of modules in the RTC
- Discussed computational complexity
- Presented different solution approaches
- Experiments for IP for five Swedish airports
  ➤ Show applicability of our approach
  ➤ Tradeoffs: more shifts — larger shifts (more minutes) — more APs/module
  ➤ Minor shifts (few minutes) can significantly reduce necessary number of modules
  ➤ Cooperation between airlines, airport owners and ANSPs may help in reduction of RTC operation costs

Future Work

- Our conflict definition may be too conservative/precautionary
- They cannot be discarded, and will influence staff planning
  ➤ Continues discussion with operations
  ➤ Possibly: distinguish arrival/departures
  ➤ Possibly: consider uncertainty
- Computational complexity of FRAMA with $\Delta > 0$ and even $\text{MAP}=2$ is open
- Currently we do not care which airlines affected by shift (possibly all to a single airline)
  ➤ Take equity into account (2 airlines, airline A operating 150 flights, airline B operating 75; reassign slot for 60 flights→ aim for 40 new slots for airline A, 20 new slots for airline B)

Thank you.