To wait or not to wait? — The delay management problem in public transportation

Anita Schöbel

Institute for Numerical and Applied Mathematics
Georg-August Universität Göttingen

29. September 2006
Contents

1 What is delay management?

2 Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3 A bicriteria model

4 Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5 Summary and Extensions
What is delay management?

To wait or not to wait?
To wait or not to wait?
What is delay management?

To wait or not to wait?
...more formally...

Vehicle $i$ arrives at a station with a delay.

What should a connecting vehicle $j$ do?

- Wait and cause further delays?
  - for customers within the vehicles
  - for customers waiting to board the vehicle later on
  - for subsequent other vehicles

- Not wait although customers who want to change from $i$ to $j$ will miss their connections.
What is delay management?

...more formally...

Vehicle $i$ arrives at a station with a delay.

What should a connecting vehicle $j$ do?

**Wait** and cause further delays?
- for customers within the vehicles
- for customers waiting to board the vehicle later on
- for subsequent other vehicles

**Not wait** although customers who want to change from $i$ to $j$ will miss their connections.
Vehicle $i$ arrives at a station with a delay.

What should a connecting vehicle $j$ do?

**Wait**
and cause further delays?
- for customers within the vehicles
- for customers waiting to board the vehicle later on
- for subsequent other vehicles

**Not wait**
although customers who want to change from $i$ to $j$ will miss their connections.
What is delay management?

... more formally ...

Vehicle $i$ arrives at a station with a delay.

What should a connecting vehicle $j$ do?

**Wait**
and cause further delays?
- for customers within the vehicles
- for customers waiting to board the vehicle later on
- for subsequent other vehicles

**Not wait**
although customers who want to change from $i$ to $j$ will miss their connections.
Vehicle $i$ arrives at a station with a delay.

What should a connecting vehicle $j$ do?

**Wait**

- and cause further delays?
  - for customers within the vehicles
  - for customers waiting to board the vehicle later on
  - for subsequent other vehicles

**Not wait**

although customers who want to change from $i$ to $j$ will miss their connections.
The delay management problem

In case of some known delays, find wait-depart decisions for all vehicles in the PTN such that the “inconvenience” for the customers is minimized.
The delay management problem

In case of some known delays, find wait-depart decisions for all vehicles in the PTN such that the "inconvenience" for the customers is minimized.
What is delay management?

How to evaluate the inconvenience?

Possible objective functions:

a) (weighted) number of missed connections
   All vehicles wait!

b) (weighted) number of delayed vehicles
   all vehicles depart in time!

c) weighted sum of a) and b)

d) bicriterial problem w.r.t. a) and b)

e) average delay of the customers
How to evaluate the inconvenience?

Possible objective functions:

a) (weighted) number of missed connections
b) (weighted) number of delayed vehicles
c) weighted sum of a) and b)
d) bicriterial problem w.r.t. a) and b)
e) average delay of the customers
How to evaluate the inconvenience?

Possible objective functions:

a) (weighted) number of missed connections  
   All vehicles wait!

b) (weighted) number of delayed vehicles  
   all vehicles depart in time!

c) weighted sum of a) and b)

d) bicriterial problem w.r.t. a) and b)

e) average delay of the customers
What is delay management?

Complexity

At a single station the problem is easy.

But:

Theorem (Gatto, Jakob, Peeters, Schöbel, 2005)

The delay management problem is NP hard.

Proof: Reduction to independent set.
Complexity

At a single station the problem is easy.

But:

Theorem (Gatto, Jakob, Peeters, Schöbel, 2005)

The delay management problem is NP hard.

Proof: Reduction to independent set.
Real-world delay management projects

Delay management for **bus companies**
Supported between 2000-2003
Data:
- timetables
- very approximate customers data
- typical delay scenarios from the planners

**Result:** Development of models and solution algorithms

Delay management for **German Rail** (with Aachen and Dresden)
Supported between 2004-2007
Data:
- timetables, including freight trains
- customers data in development
- Data about all delays in the year 2004

**Goal:** Including the capacity constraints of railways and use in practice...!
Real-world delay management projects

Delay management for **bus companies**
Supported between 2000-2003
Data:
- timetables
- very approximate customers data
- typical delay scenarios from the planners

**Result:** Development of models and solution algorithms

Delay management for **German Rail** (with Aachen and Dresden)
Supported between 2004-2007
Data:
- timetables, including freight trains
- customers data in development
- Data about all delays in the year 2004

**Goal:** Including the capacity constraints of railways and use in practice...!
What is delay management?

The basic model

Assumptions:

1. $T$ is the fixed time period before the next vehicle of the same type arrives.

2. In the next period, all vehicles are in time.

If a customer . . .

- . . . reaches all his connections: delay = arrival delay of his last vehicle at his destination station
- . . . misses a connection: delay = $T$.

Consequence: It is necessary to know the actual timetable.
The basic model

Assumptions:

1. $T$ is the fixed time period before the next vehicle of the same type arrives.
2. In the next period, all vehicles are in time.

If a customer . . .

- . . . reaches all his connections: 
  delay = arrival delay of his last vehicle at his destination station
- . . . misses a connection: delay = $T$.

Consequence: It is necessary to know the actual timetable.
The basic model

Assumptions:

1. $T$ is the fixed time period before the next vehicle of the same type arrives.
2. In the next period, all vehicles are in time.

If a customer . . .

- . . . reaches all his connections: delay = arrival delay of his last vehicle at his destination station
- . . . misses a connection: delay = $T$.

Consequence: It is necessary to know the actual timetable.
The basic model

Assumptions:

1. \( T \) is the fixed time period before the next vehicle of the same type arrives.

2. In the next period, all vehicles are in time.

If a customer . . .

- . . . reaches all his connections: delay = arrival delay of his last vehicle at his destination station
- . . . misses a connection: delay = \( T \).

Consequence: It is necessary to know the

actual timetable
What is delay management?

Event-activity network

\[ \mathcal{E} = \mathcal{E}_{arr} \cup \mathcal{E}_{dep}, \quad \mathcal{A} = \mathcal{A}_{drive} \cup \mathcal{A}_{wait} \cup \mathcal{A}_{change} \]
Contents

1. What is delay management?

2. Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3. A bicriteria model

4. Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5. Summary and Extensions
Path-based model

Parameter:
- $s_a$ slack time of activity $a \in A$, i.e. scheduled time for $a$ - technically necessary time for $a$
- $d_i$ source delay for all $i \in E$ (may be 0)
- $P$ set of customers' paths (as sequences of events)
- $w_p$ number of customers using $p \in P$.

Variables:
- $y_i = \text{delay of event } i \in E$
  - $= \text{actual time - scheduled time}$.
- $z_p = \begin{cases} 
0 & \text{if all connections of } p \text{ are maintained} \\
1 & \text{if a connection of } p \text{ is missed.}
\end{cases}$
Path-based model

Parameter:

- $s_a$ slack time of activity $a \in A$, i.e. scheduled time for $a$ - technically necessary time for $a$
- $d_i$ source delay for all $i \in E$ (may be 0)
- $P$ set of customers’ paths (as sequences of events)
- $w_p$ number of customers using $p \in P$.

Variables:

- $y_i = \text{delay of event } i \in E$
  - actual time - scheduled time.
- $z_p = \begin{cases} 0 & \text{if all connections of } p \text{ are maintained} \\ 1 & \text{if a connection of } p \text{ is missed.} \end{cases}$
Path-based formulation

\[
\min f_{TDM-A} = \sum_{p \in \mathcal{P}} w_p (y_{i(p)} (1 - z_p) + Tz_p)
\]

such that

\[
\begin{align*}
    y_i &\geq d_i & \text{for all } i \in \mathcal{E}_{del} \\
    y_i - y_j &\leq s_a & \text{for all } a = (i, j) \in A_{wait} \cup A_{drive} \\
    -Mz_p + y_i - y_j &\leq s_a & \text{for all } a = (i, j) \in A_{change} \cap p \\
    y_i &\in \mathbb{N} & \text{for all } i \in \mathcal{E} \\
    z_p &\in \{0, 1\} & \text{for all } p \in \mathcal{P}
\end{align*}
\]
Path-based formulation

\[
\min f_{\text{TDM-A}} = \sum_{p \in \mathcal{P}} w_p (y_i(p) (1 - z_p) + T z_p)
\]

such that

\[
\begin{align*}
    y_i & \geq d_i \quad \text{for all } i \in \mathcal{E}_{\text{del}} \\
    y_i - y_j & \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}} \\
    -Mz_p + y_i - y_j & \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{\text{change}} \cap p \\
    y_i & \in \mathbb{N} \quad \text{for all } i \in \mathcal{E} \\
    z_p & \in \{0, 1\} \quad \text{for all } p \in \mathcal{P}
\end{align*}
\]
Path-based formulation

\[
\begin{align*}
\min f_{TDM-A} &= \sum_{p \in \mathcal{P}} w_p(y_{i(p)}(1 - z_p) + Tz_p) \\
\text{such that} & \quad y_i \geq d_i \quad \text{for all } i \in \mathcal{E}_{del} \\
& \quad y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive} \\
& \quad -Mz_p + y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{change} \cap p \\
& \quad y_i \in \mathbb{N} \quad \text{for all } i \in \mathcal{E} \\
& \quad z_p \in \{0, 1\} \quad \text{for all } p \in \mathcal{P}
\end{align*}
\]
Path-based formulation

\[ \min f_{TDM-A} = \sum_{p \in P} w_p(y_{i(p)}(1 - z_p) + Tz_p) \]

such that

\[ y_i \geq d_i \quad \text{for all } i \in \mathcal{E}_{del} \]

\[ y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive} \]

\[ -Mz_p + y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{change} \cap p \]

\[ y_i \in \mathbb{N} \quad \text{for all } i \in \mathcal{E} \]

\[ z_p \in \{0, 1\} \quad \text{for all } p \in \mathcal{P} \]
Path-based formulation

\[
\min f_{TDM-A} = \sum_{p \in P} w_p(y_{i(p)}(1 - z_p) + Tz_p)
\]

such that
\[
\begin{align*}
  y_i & \geq d_i \quad \text{for all } i \in \mathcal{E}_{del} \\
  y_i - y_j & \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}} \\
  -Mz_p + y_i - y_j & \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{\text{change}} \cap p \\
  y_i & \in \mathbb{N} \quad \text{for all } i \in \mathcal{E} \\
  z_p & \in \{0, 1\} \quad \text{for all } p \in \mathcal{P}
\end{align*}
\]
Path-based formulation

\[
\min f_{TDM-A} = \sum_{p \in \mathcal{P}} w_p (q_p + Tz_p)
\]

such that

\[
y_i \geq d_i \quad \text{for all } i \in \mathcal{E}_{del}
\]
\[
y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive}
\]
\[
-Mz_p + y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{change} \cap p
\]
\[
y_i \in \mathbb{N} \quad \text{for all } i \in \mathcal{E}
\]
\[
z_p \in \{0, 1\} \quad \text{for all } p \in \mathcal{P}
\]
\[
-Mz_p + y_{i(p)} - q_p \leq 0 \quad \text{for all } p \in \mathcal{P}
\]
\[
q_p \geq 0 \quad \text{for all } p \in \mathcal{P}
\]
Path-based formulation

\[
\min f_{TDM-A} = \sum_{p \in \mathcal{P}} w_p (q_p + Tz_p)
\]

such that

\[
y_i \geq d_i \quad \text{for all } i \in \mathcal{E}_{del}
\]

\[
y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive}
\]

\[
-Mz_p + y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in \mathcal{A}_{change} \cap \mathcal{P}
\]

\[
y_i \in \mathbb{N} \quad \text{for all } i \in \mathcal{E}
\]

\[
z_p \in \{0, 1\} \quad \text{for all } p \in \mathcal{P}
\]

\[
-Mz_p + y_{i(p)} - q_p \leq 0 \quad \text{for all } p \in \mathcal{P}
\]

\[
q_p \geq 0 \quad \text{for all } p \in \mathcal{P}
\]

**Result:** A linear model.
Contents

1. What is delay management?

2. Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3. A bicriteria model

4. Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5. Summary and Extensions
Activity-based model

Variables:
- \( y_i = \) delay of event \( i \in \mathcal{E} \).
- \( z_a = \begin{cases} 
0 & \text{if connection } a \text{ maintained} \\
1 & \text{if connection } a \text{ missed}
\end{cases} \)
- \( w_a = \) number of customers using activity \( a \).

Parameter:
- \( s_a = \) slack time of activity \( a \in \mathcal{A} \)
- \( d_i = \) source delay for all \( i \in \mathcal{E} \) (may be 0)
- \( \mathcal{P} = \) set of customers’ paths (as sequences of events).
Calculating the average delay

Idea: The delay of a customer is the sum of its additional delays over all his activities.

The additional delay of an activity \( a = (i, j) \) is the tension \( y_j - y_i \) (can be positive or negative!)

If \( a \) is missed, the additional delay is \( T - y_i = (T - y_j) + (y_j - y_i) \)

Let \( w_a = \) number of customers using activity \( a \).

sum of all delays=

\[
f_{TDM} = \sum_{a=(i,j) \in \mathcal{A}^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in \mathcal{A}_{\text{change}}} w_a z_a(T - y_j).
\]
Activity-based model

\[
\min \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a(T - y_j)
\]

s.t. \[y_i \geq d_i \text{ for all delayed } i\]

\[y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{\text{wait}} \cup A_{\text{drive}}\]

\[ -Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{\text{change}}\]

\[\tilde{z}_a^p + \sum_{\tilde{a} \in p \cap A_{\text{change}}: \tilde{a} \prec a} z_{\tilde{a}} \geq 1 \text{ for all paths } p \text{ and } a \in p\]

\[\tilde{z}_a^p + z_{\tilde{a}} \leq 1 \text{ for all paths } p \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a\]

\[w_a = \sum_{\text{paths } p: a \in p} w_p \tilde{z}_a^p \text{ for all } a \in A^s\]

\[y_i, w_a \in \mathbb{N}^0, \quad z_a, \tilde{z}_a^p \in \{0, 1\}\]
Activity-based model

\[
\min \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a (T - y_j)
\]

s.t. \( y_i \geq d_i \) for all delayed \( i \)
\( y_i - y_j \leq s_a \) for all \( a = (i, j) \in A_{\text{wait}} \cup A_{\text{drive}} \)
\( -Mz_a + y_i - y_j \leq s_a \) for all \( a = (i, j) \in A_{\text{change}} \)
\( \tilde{z}_a^p + \sum_{\tilde{a} \in p \cap A_{\text{change}}: \tilde{a} \prec a} z_{\tilde{a}} \geq 1 \) for all paths \( p \) and \( a \in p \)
\( \tilde{z}_a^p + z_{\tilde{a}} \leq 1 \) for all paths \( p \) and \( a, \tilde{a} \in p \) with \( \tilde{a} \prec a \)
\( w_a = \sum_{\text{paths } p: a \in p} w_p \tilde{z}_a^p \) for all \( a \in A^s \)

\( y_i, w_a \in \mathbb{N}^0 \), \( z_a, \tilde{z}_a^p \in \{0, 1\} \)
Activity-based model

\[ \min \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{change}} w_a z_a (T - y_j) \]

s.t. \[ y_i \geq d_i \] for all delayed \( i \)
\[ y_i - y_j \leq s_a \] for all \( a = (i, j) \in A_{wait} \cup A_{drive} \)
\[ -Mz_a + y_i - y_j \leq s_a \] for all \( a = (i, j) \in A_{change} \)
\[ \tilde{z}_a^p + \sum_{\tilde{a} \in p \cap A_{change}: \tilde{a} \prec a} z_{\tilde{a}} \geq 1 \] for all paths \( p \) and \( a \in p \)
\[ \tilde{z}_a^p + z_{\tilde{a}} \leq 1 \] for all paths \( p \) and \( a, \tilde{a} \in p \) with \( \tilde{a} \prec a \)
\[ w_a = \sum_{\text{paths} p: a \in p} w_p \tilde{z}_a^p \] for all \( a \in A^s \)
\[ y_i, w_a \in \mathbb{N}^0, \quad z_a, \tilde{z}_a^p \in \{0, 1\} \]
Activity-based model

\[
\begin{align*}
\min & \quad \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a(T - y_j) \\
\text{s.t.} & \quad y_i \geq d_i \text{ for all delayed } i \\
& \quad y_i - y_j \leq s_a \text{ for all } a = (i,j) \in A_{\text{wait}} \cup A_{\text{drive}} \\
& \quad - Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i,j) \in A_{\text{change}} \\
\tilde{z}^p_a + \sum_{\tilde{a} \in p \cap A_{\text{change}}: \tilde{a} \prec a} z_{\tilde{a}} & \geq 1 \text{ for all paths } p \text{ and } a \in p \\
\tilde{z}^p_a + z_{\tilde{a}} & \leq 1 \text{ for all paths } p \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a \\
w_a & = \sum_{\text{paths } p: a \in p} w_p \tilde{z}^p_a \text{ for all } a \in A^s \\
y_i, w_a & \in \mathbb{N}^0, \quad z_a, \tilde{z}^p_a \in \{0, 1\}
\end{align*}
\]
Activity-based model

\[
\begin{align*}
\text{min} & & \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a(T - y_j) \\
\text{s.t.} & & y_i \geq d_i \text{ for all delayed } i \\
& & y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{\text{wait}} \cup A_{\text{drive}} \\
& & -Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{\text{change}} \\
\tilde{z}_a^p & + \sum_{\tilde{a} \in p \cap A_{\text{change}}: \tilde{a} \prec a} z_{\tilde{a}} & \geq 1 \text{ for all paths } p \text{ and } a \in p \\
\tilde{z}_a^p + z_{\tilde{a}} & \leq 1 \text{ for all paths } p \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a \\
w_a & = \sum_{\text{paths } p: a \in p} w_p \tilde{z}_a^p \text{ for all } a \in A^s \\
y_i, w_a & \in \mathbb{N}^0, \quad z_a, \tilde{z}_a^p \in \{0, 1\}
\end{align*}
\]
Activity-based model

\[
\min \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{change}} w_a z_a (T - y_j)
\]

\[
\text{s.t.} \quad y_i \geq d_i \quad \text{for all delayed } i
\]

\[
y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in A_{wait} \cup A_{drive}
\]

\[
-Mz_a + y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in A_{change}
\]

\[
\tilde{z}_{a}^p + \sum_{\tilde{a} \in p \cap A_{change}: \tilde{a} \prec a} z_{\tilde{a}} \geq 1 \quad \text{for all paths } p \text{ and } a \in p
\]

\[
\tilde{z}_{a}^p + z_{\tilde{a}} \leq 1 \quad \text{for all paths } p \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a
\]

\[
w_a = \sum_{\text{paths } p: a \in p} w_p \tilde{z}_{a}^p \quad \text{for all } a \in A^s
\]

\[
y_i, w_a \in \mathbb{N}^0, \quad z_a, \tilde{z}_{a}^p \in \{0, 1\}
\]
Relation of both formulations

Theorem

- The set of feasible solutions in the path-based formulation contains the set of feasible solutions in the activity-based formulation.
- The optimal solution values of both formulations are the same.

The activity-based formulation

- has a stronger LP-relaxation
- but is cubic!

...how to solve it?
Relation of both formulations

Theorem

- The set of feasible solutions in the path-based formulation contains the set of feasible solutions in the activity-based formulation.
- The optimal solution values of both formulations are the same.

The activity-based formulation

- has a stronger LP-relaxation
- but is cubic!

... how to solve it?
Relation of both formulations

Theorem

- The set of feasible solutions in the path-based formulation contains the set of feasible solutions in the activity-based formulation.
- The optimal solution values of both formulations are the same.

The activity-based formulation

- has a stronger LP-relaxation
- but is cubic!

... how to solve it?
Simplify the activity-based formulation

**Idea:** Fix the customer weights $w_a$ as parameters by

$$w_a = \sum_{p \in P} \sum_{a \in p} w_p,$$

i.e. pretend that all customers are able to travel as they have planned.
Fixing the $w_a$

$$\begin{align*}
\min & \quad \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{change}} w_a z_a(T - y_j) \\
\text{s.t.} & \quad y_i \geq d_i \quad \text{for all delayed } i \\
& \quad y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in A_{wait} \cup A_{drive} \\
& \quad -M z_a + y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in A_{change} \\
& \quad \tilde{z}_a + \sum_{\tilde{a} \in p \cap A_{change} : \tilde{a} \prec a} z_{\tilde{a}} \geq 1 \quad \text{for all paths } p \text{ and } a \in p \\
& \quad \tilde{z}_a + z_{\tilde{a}} \leq 1 \quad \text{for all paths } p \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a \\
& \quad w_a = \sum_{\text{paths } p : a \in p} w_p \tilde{z}_a \quad \text{for all } a \in A^s \\
& \quad y_i, w_a \in \mathbb{N}^0, \quad z_a, \tilde{z}_a \in \{0, 1\}
\end{align*}$$
Fixing the $w_a$

$$\begin{align*}
\min & \quad \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a (T - y_j) \\
\text{s.t.} & \quad y_i \geq d_i \quad \text{for all delayed } i \\
& \quad y_i - y_j \leq s_a \quad \text{for all } a = (i,j) \in A_{\text{wait}} \cup A_{\text{drive}} \\
& \quad -M z_a + y_i - y_j \leq s_a \quad \text{for all } a = (i,j) \in A_{\text{change}} \\
& \quad \tilde{z}_a^p + \sum_{\tilde{a} \in p \cap A_{\text{change}}: \tilde{a} \prec a} z_{\tilde{a}} \geq 1 \quad \text{for all paths } p \text{ and } a \in p \\
& \quad \tilde{z}_a^p + z_{\tilde{a}} \leq 1 \quad \text{for all paths } p \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a \\
& \quad w_a = \sum_{\text{paths } p: a \in p} w_p \tilde{z}_a^p \quad \text{for all } a \in A^s \\
& \quad y_i, w_a \in \mathbb{N}^0, \quad z_a, \tilde{z}_a^p \in \{0, 1\}
\end{align*}$$
Fixing the $w_a$

$$\min \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{change}} w_a z_a(T - y_j)$$

s.t.

$$y_i \geq d_i \text{ for all delayed } i$$

$$y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{wait} \cup A_{drive}$$

$$-Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{change}$$

$$\tilde{z}_a^p + \sum_{\tilde{a} \in p \cap A_{change}: \tilde{a} \prec a} z_{\tilde{a}} \geq 1 \text{ for all paths } p \text{ and } a \in p$$

$$\tilde{z}_a^p + z_{\tilde{a}} \leq 1 \text{ for all paths } p \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a$$

$$w_a = \sum_{\text{paths } p: a \in p} w_p \tilde{z}_a^p \text{ for all } a \in A^s$$

$$y_i, w_a \in \mathbb{N}^0, \quad z_a, \tilde{z}_a^p \in \{0, 1\}$$
Fixing the $w_a$

$$
\begin{align*}
\min & \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a(T - y_j) \\
\text{s.t.} & y_i \geq d_i \quad \text{for all delayed } i \\
& y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in A_{\text{wait}} \cup A_{\text{drive}} \\
& -Mz_a + y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in A_{\text{change}} \\
& \tilde{z}_a^p + \sum_{\tilde{a} \in p \cap A_{\text{change}}: \tilde{a} \prec a} z_{\tilde{a}} \geq 1 \quad \text{for all paths } p \text{ and } a \in p \\
& \tilde{z}_a^p + z_{\tilde{a}} \leq 1 \quad \text{for all paths } p \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a \\
& w_a = \sum_{\text{paths } p: a \in p} w_p \tilde{z}_a^p \quad \text{for all } a \in A^s \\
& y_i, w_a \in \mathbb{N}^0, \quad z_a, \tilde{z}_a^p \in \{0, 1\}
\end{align*}
$$
Step 1: Fixing the $w_a$

$$\min \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) \ + \ \sum_{a=(i,j) \in A_{\text{change}}} w_ay_a(T - y_j)$$

s.t. $y_i \geq d_i$ for all delayed $i$

$y_i - y_j \leq s_a$ for all $a = (i,j) \in A_{\text{wait}} \cup A_{\text{drive}}$

$-Mz_a + y_i - y_j \leq s_a$ for all $a = (i,j) \in A_{\text{change}}$

$y_i \in \mathbb{N}^0$, $z_a \in \{0, 1\}$
Step 1: Fixing the $w_a$

\[
\begin{align*}
\min & \sum_{a=(i,j) \in \mathcal{A}^s} w_a (y_j - y_i) + \sum_{a=(i,j) \in \mathcal{A}_{\text{change}}} w_a z_a (T - y_j) \\
\text{s.t.} & \quad y_i \geq d_i \text{ for all delayed } i \\
& \quad y_i - y_j \leq s_a \text{ for all } a = (i, j) \in \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}} \\
& \quad -Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i, j) \in \mathcal{A}_{\text{change}} \\
& \quad y_i \in \mathbb{N}^0, \quad z_a \in \{0, 1\}
\end{align*}
\]

simpler, but unfortunately wrong . . .
Example
Example
Example
Example
Example

Models for minimizing the average delay

Activity-based model

Anita Schöbel (NAM)

Delay Management

29. September 2006
Example
Approximate linearization

\[
\begin{align*}
\min & \quad \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a (T - y_j) \\
\text{s.t.} & \quad y_i \geq d_i \text{ for all delayed } i \\
& \quad y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{\text{wait}} \cup A_{\text{drive}} \\
& \quad -Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{\text{change}} \\
& \quad y_i \in \mathbb{N}^0, \quad z_a \in \{0, 1\}
\end{align*}
\]
Approximate linearization

\[
\begin{align*}
\min & \quad \sum_{a=(i,j)\in A^s} w_a(y_j - y_i) + \sum_{a=(i,j)\in A_{\text{change}}} w_a z_a (T - y_j) \\
\text{s.t.} & \quad y_i \geq d_i \text{ for all delayed } i \\
& \quad y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{\text{wait}} \cup A_{\text{drive}} \\
& \quad -Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{\text{change}} \\
& \quad y_i \in \mathbb{N}^0, \quad z_a \in \{0, 1\}
\end{align*}
\]
Approximate linearization

\[
\begin{align*}
\min & \quad \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a (T - y_j) \\
\text{s.t.} & \quad y_i \geq d_i \text{ for all delayed } i \\
& \quad y_i - y_j \leq s_a \text{ for all } a = (i,j) \in A_{\text{wait}} \cup A_{\text{drive}} \\
& \quad -Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i,j) \in A_{\text{change}} \\
& \quad y_i \in \mathbb{N}^0, \quad z_a \in \{0, 1\}
\end{align*}
\]
Approximate linearization

\[
\begin{align*}
\min & \quad \sum_{a=(i,j) \in A^s} w_a (y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a (T - y_j) \\
\text{s.t.} & \quad y_i \geq d_i \quad \text{for all delayed } i \\
& \quad y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in A_{\text{wait}} \cup A_{\text{drive}} \\
& \quad -Mz_a + y_i - y_j \leq s_a \quad \text{for all } a = (i, j) \in A_{\text{change}} \\
& \quad y_i \in \mathbb{N}^0, \quad z_a \in \{0, 1\}
\end{align*}
\]

Result: A linear model!
Approximate linearization

\[
\min \sum_{a=(i,j) \in A^s} w_a (y_j - y_i) + \sum_{a=(i,j) \in A_{\text{change}}} w_a z_a (T - y_j)
\]

s.t. \( y_i \geq d_i \) for all delayed \( i \)

\( y_i - y_j \leq s_a \) for all \( a = (i, j) \in A_{\text{wait}} \cup A_{\text{drive}} \)

\( -Mz_a + y_i - y_j \leq s_a \) for all \( a = (i, j) \in A_{\text{change}} \)

\( y_i \in \mathbb{N}^0, \quad z_a \in \{0, 1\} \)

Result: A linear model!

\[\ldots\] \text{but how badly wrong?}
Questions:

1. How wrong is the simplified model?
2. Can we solve it efficiently?
The “never-meet” property

Definition

The delay management problem has the **never-meet property** if each (time-minimal) feasible solution with zero slack times satisfies for all $j \in E$:

1. If $(i_1, j), (i_2, j) \in A$, and $y_{i_2} > 0$ then $y_{i_1} = 0$.
2. If $(i_1, j) \in A$, and $d_j > 0$ then $y_{i_1} = 0$.
The “never-meet” property

Definition

The delay management problem has the **never-meet property** if each (time-minimal) feasible solution with zero slack times satisfies for all \( j \in E \):

1. If \((i_1, j), (i_2, j) \in A\), and \(y_{i_2} > 0\) then \(y_{i_1} = 0\).
2. If \((i_1, j) \in A\), and \(d_j > 0\) then \(y_{i_1} = 0\).
The “never-meet” property

**Definition**

The delay management problem has the **never-meet property** if each (time-minimal) feasible solution with zero slack times satisfies for all $j \in E$:

1. If $(i_1, j), (i_2, j) \in A$, and $y_{i_2} > 0$ then $y_{i_1} = 0$.
2. If $(i_1, j) \in A$, and $d_j > 0$ then $y_{i_1} = 0$. 

“never-meet”-property
Good news!

**Theorem**

*The linear model is correct if the never-meet property holds.*

**Proof:**
Good news!

Theorem

*The linear model is correct if the never-meet property holds.*

Proof:
Good news!

**Theorem**

*The linear model is correct if the never-meet property holds.*

**Proof:**

\[ \overline{i} \rightarrow \overline{a} \rightarrow \overline{j} \]

The variables that have been changed are multiplied by zero and do not contribute.
Good news!

Theorem

*The linear model is correct if the never-meet property holds.*

Proof:
Good news!

**Theorem**

*The linear model is correct if the never-meet property holds.*

**Proof:**
Good news!

Theorem

*The linear model is correct if the never-meet property holds.*

Proof:

The variables that have been changed are multiplied by zero and do not contribute.
More Good News!

- The never-meet property can be tested efficiently by the forward phase of the critical path method (CPM).
- The never-meet property is in practice often "almost" satisfied.

Typical examples with 20 source delays:

<table>
<thead>
<tr>
<th>time interval</th>
<th>conflicts</th>
</tr>
</thead>
<tbody>
<tr>
<td>up to 1 hour</td>
<td>(\sim) up to 200</td>
</tr>
<tr>
<td>up to 30 minutes</td>
<td>(\sim) up to 45</td>
</tr>
<tr>
<td>up to 10 minutes</td>
<td>(\sim) nearly always 0</td>
</tr>
</tbody>
</table>
Models for minimizing the average delay

Solution approaches

Contents

1. What is delay management?

2. Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3. A bicriteria model

4. Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5. Summary and Extensions
Solution approach in case of the never-meet property

Theorem

*The delay management problem can be solved in $O(|A|)$ time if the never-meet property holds.*
Idea of the algorithm:

1. **Decompose**: At each changing activity $a$ decompose into problems $a_1, \ldots, a_L$ where there is no other changing activities between each of the $a_1, \ldots, a_L$ and $a$.

Lemma

The problems $a_1, \ldots, a_L$ are independent of each other

2. **Compose**: The optimal solution for $a$ can be determined if we know the optimal solutions for $a_1, \ldots, a_L$. 
Idea of the algorithm:

1. Decompose: At each changing activity $a$ decompose into problems $a_1, \ldots, a_L$ where there is no other changing activities between each of the $a_1, \ldots, a_L$ and $a$.

2. Compose: The optimal solution for $a$ can be determined if we know the optimal solutions for $a_1, \ldots, a_L$.

Lemma

The problems $a_1, \ldots, a_L$ are independent of each other.
Idea of algorithm:
And if the never-meet property is not true?

- Solve the simplified model to get a sharp lower bound by a dual approach
- Solve the complete model by branch and bound
Solving the simplified model

**Idea:** Solve the LP-Relaxation by looking at the dual program:

\[
\max \sum_{i \in \mathcal{E}_{\text{del}}} d_i \xi_i - \sum_{a \in \mathcal{A}} s_a \eta_a
\]

such that

\[
\eta_a \leq \frac{w_a T}{M} \text{ for all } a \in \mathcal{A}_{\text{change}}
\]

\[
\sum_{a=(i,j)} \eta_a + \sum_{a=(j,i)} \eta_a \leq w_i \text{ for all } i \in \mathcal{E} \setminus \mathcal{E}_{\text{del}}
\]

\[
\sum_{a=(i,j)} \eta_a + \sum_{a=(j,i)} \eta_a = w_i - \xi_i \text{ for all } i \in \mathcal{E}_{\text{del}}
\]

\[
\eta_a \geq 0 \text{ for all } a \in \mathcal{A}, \quad \xi_i \geq 0 \text{ for all } i \in \mathcal{E}
\]

Generalized flow problem (see [Brandenburg, 2003])
**Performance Guarantee**

**Theorem**

Let \((y, z)\) be an optimal solution of the relaxation with objective value \(f_{\text{simple}}\). Let \(f^*\) be the optimal objective value of the simplified model. Then

\[
0 \leq f^* - f_{\text{simple}} \leq T \sum_{a: z_a > 0} w_a \left(1 - \frac{y_i - y_j - s_a}{M}\right).
\]

**Special case:** For \(s_a = 0\) we obtain \(f^* - f_{\text{simple}} \leq 0\) i.e. the relaxation solves the IP optimally!

**Improvement of the relaxation:** Replace \(M\) by \(M_a \geq y_i - s_a\) for all \(a = (i, j) \in A_{\text{change}}\)
Branch and Bound approach for the general case

Branching along the changing activities, in their natural order

In each branch & bound node:
- “early” and “late” reduction
- test if never-meet property holds, and if yes solve to optimality
- good lower bound by solving the simplified version
- upper bounds by heuristics
Branch and Bound approach for the general case

Branching along the changing activities, in their natural order

In each branch & bound node:
- “early” and “late” reduction
- test if never-meet property holds, and if yes solve to optimality
- good lower bound by solving the simplified version
- upper bounds by heuristics
Branch and Bound approach for the general case

Branching along the changing activities, in their natural order

In each branch & bound node:
- “early” and “late” reduction
- test if never-meet property holds, and if yes solve to optimality
- good lower bound by solving the simplified version
- upper bounds by heuristics
Branch and Bound approach for the general case

Branching along the changing activities, in their natural order

In each branch & bound node:
- “early” and “late” reduction
- test if never-meet property holds, and if yes solve to optimality
- good lower bound by solving the simplified version
- upper bounds by heuristics
Contents

1. What is delay management?

2. Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3. A bicriteria model

4. Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5. Summary and Extensions
Two (conflicting) objective functions

- minimize (weighted) number of missed connections
- minimize (weighted) sum of all delays of all vehicles

**Remark:** These objectives are conflicting!

**Bi-criteria problem!**

**Goal:** Find Pareto Solutions.
A bicriteria model

Two (conflicting) objective functions

- minimize (weighted) number of missed connections
- minimize (weighted) sum of all delays of all vehicles

Remark: These objectives are conflicting!

Bi-criteria problem!

Goal: Find Pareto Solutions.
Two (conflicting) objective functions

- minimize (weighted) number of missed connections
- minimize (weighted) sum of all delays of all vehicles

Remark: These objectives are conflicting!

Bi-criteria problem!

Goal: Find Pareto Solutions.
Pareto solutions
Pareto solutions
Pareto solutions
Pareto solutions
Pareto solutions
Pareto solutions
Pareto solutions
Pareto Solutions

An actual timetable is called **Pareto solution**, if there is no other feasible timetable with less delayed vehicles and less missed transfers.

**Idea:** Use methods of project planning to find Pareto solutions.
Bicriteria project planning

**Given:** A set of activities with precedence relations

**Look for:** starting times for each of the activities

**such that:** the project is completes as fast as possible

**Extension:** DTCTP
Activities can be fastened by spending more money!

**Two objective functions:**
- How much money should be spend?
- When can the project be completes?

**Also here:** Looking for Pareto solutions.
Bicriteria project planning

**Given:** A set of activities with precedence relations

**Look for:** starting times for each of the activities

**such that:** the project is completes as fast as possible

**Extension: DTCTP**
Activities can be fastened by spending more money!

Two objective functions:
- How much money should be spend?
- When can the project be completes?

Also here: Looking for Pareto solutions.
Bicriteria project planning

**Given:** A set of activities with precedence relations

**Look for:** starting times for each of the activities

**such that:** the project is completes as fast as possible

**Extension: DTCTP**
Activities can be fastened by spending more money!

**Two objective functions:**
- How much money should be spend?
- When can the project be completes?

**Also here:** Looking for Pareto solutions.
Delay management and project planning

- Spending money (penalty for missed transfers)
- Gives the changing activities a duration of $-\infty$

Obtain DTCTP and adapt solution method of Demeulemeester, Herroelen, and Elmaghraby

Difficulties:
- Network has to be transformed to a project planning network
- Minimize project length is not the same as minimize number of delayed vehicles: shrink in correct order, add a third mode
Delay management and project planning

- Spending money (\(=\)penalty for missed transfers)
- gives the changing activities a duration of \(-\infty\)

Obtain DTCTP and adapt solution method of Demeulemeester, Herroelen, and Elmaghraby

Difficulties:
- network has to be transformed to a project planning network
- minimize project length is not the same as minimize number of delayed vehicles: shrink in correct order, add a third mode
Delay management and project planning

- Spending money (penalty for missed transfers)
- Gives the changing activities a duration of $-\infty$

Obtain DTCTP and adapt solution method of Demeulemeester, Herroelen, and Elmaghraby

Difficulties:
- Network has to be transformed to a project planning network
- Minimize project length is not the same as minimize number of delayed vehicles: shrink in correct order, add a third mode
Constructing the project network

A bicriteria model

- Event 1: Departure
  - \( \Pi_1 \)

- Event 2: Arrival
  - \( \Pi_2 \)
  - \( L_{23} \)

- Event 3: Departure
  - \( \Pi_3 \)
  - \( L_{63} \)

- Event 4: Arrival
  - \( \Pi_4 \)
  - \( L_{34} \)

- Event 5: Departure
  - \( \Pi_5 \)
  - \( L_{56} \)

- Event 6: Arrival
  - \( \Pi_6 \)
  - \( L_{67} \)

- Event 7: Departure
  - \( \Pi_7 \)
  - \( L_{78} \)

- Event 8: Arrival
  - \( \Pi_8 \)
Constructing the project network
Merge Operations

Serial merge:

(L1,c1,d1) (L2,c2,d2) → (L1+L2, c1+c2, d1+d2)
Merge Operations

Parallel merge:

\[(L_1, c_1, d_1) \rightarrow (L_2, c_2, d_2)\] timetable arc

\[(\max\{L_1, L_2\}, c_1+c_2, d_1+[L_1-L_2]+)\]
Merge Operations

Node reduction:

(L1+L3, c1+c3, d1+d3)

(L2+L3, c2, d2)

vehicle f

(L3,c3,d3)

vehicle f

(L1,c1,d1)

(L2,c2,d2)

(L2+L3, c2, d2)
Finding Pareto timetables

1. Apply all possible serial and parallel merges.
2. If no merge is possible, take the earliest node for reduction (fixing the mode\textcolor{white}{\text{(wait/depart)}} of the single activity if necessary.)
3. If more than one activity remains, goto 1, otherwise 4.
4. Store the solutions and repeat with all other possible modes.
5. Evaluate the efficient solutions out of all stored ones.

Theorem

All Pareto solutions of the bicriterial delay management problem can be found by the shrinking method.
Numerical results
Interested?

Book:

“Optimization in public transportation: Stop Location, **Delay Management** and Tariff Planning from a Customer-oriented point of view”,

Springer, to appear (hopefully!) this summer.
Interested?

Book:

Contents of the talk

1. What is delay management?

2. Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3. A bicriteria model

4. Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5. Summary and Extensions
A new challenge: capacity constraints!
A new challenge: capacity constraints!
The Re-scheduling problem

The “Re-scheduling” problem

In case of known delays,
find an actual timetable
respecting the capacity constraints of the tracks and minimizing the delays of the trains.

Main point: Two trains must not use the same piece of track at the same time!
- in the stations
- between the stations

Survey article: Törnquist
The Re-scheduling problem

The “Re-scheduling” problem

In case of known delays,
find an actual timetable
respecting the capacity constraints of the tracks and minimizing the delays of the trains.

Main point: Two trains must not use the same piece of track at the same time!
- in the stations
- between the stations

Survey article: Törnquist
How to take capacity constraints into account?

Three approaches:

1. Iterative approach
2. Microscopic approach: Modeling the capacity constraints explicitly
3. Macroscopic approach: Approximating capacity constraints
Contents

1. What is delay management?

2. Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3. A bicriteria model

4. Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5. Summary and Extensions
Iterative approach

is used within DisKon together with *Deutschen Bahn AG*

- Macroscopic model
  - View of the passengers
  - University Göttingen and Dresden

- Microscopic model
  - Capacity constraints
  - University Aachen

- Current status of delays

- Wait-depart-decisions

- Disposition timetable

- Timetable
Iterative approach

is used within DisKon together with *Deutschen Bahn AG*

- Macroscopic model
  - View of the passengers
  - University Göttingen and Dresden

- Microscopic model
  - Capacity constraints
  - University Aachen

- Current status of delays

- Wait–depart–decisions

- Major conflicts with capacity constraints

- Disposition timetable
Contents

1. What is delay management?

2. Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3. A bicriteria model

4. Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5. Summary and Extensions
Microscopic approach

Possibilities:

Use event-activity network approach

- add all junctions, overtaking possibilities, (in the worst case: all signalling points and switches) as nodes
- simulate the headways between each pair of trains

Use Set Partitioning Approach
No two trains may use the same piece of infrastructure at the same time.

\[ x_{ibt} = \begin{cases} 1 & \text{if train } i \text{ uses block } b \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \]
Contents

1 What is delay management?

2 Models for minimizing the average delay
   - Path-based model
   - Activity-based model
   - Solution approaches

3 A bicriteria model

4 Delay Management for Railway Systems
   - Iterative approach
   - Microscopic approach
   - Macroscopic approach

5 Summary and Extensions
Lifting capacity constraints to the macroscopic model

**Idea:** Use the macroscopic event activity network

Determine *virtual activities* and “headways” for each macroscopic edge.

I.e. precedence constraints for events which all use the same part $b$ of the infrastructure:

$$\mathcal{E}(b) = \{ \text{events } i \text{ such that } a = (i, j) \text{ uses } b \}$$

This includes:

- simple headways constraints for each edge
- single track constraints
- dependencies between delays (Carla Conte)
Lifting capacity constraints to the macroscopic model

**Idea:** Use the macroscopic event activity network

Determine *virtual activities* and “headways” for each macroscopic edge.

I.e. precedence constraints for events which all use the same part $b$ of the infrastructure:

$$\mathcal{E}(b) = \{ \text{events } i \text{ such that } a = (i, j) \text{ uses } b \}$$

This includes:

- simple headways constraints for each edge
- single track constraints
- dependencies between delays (Carla Conte)
Lifting capacity constraints to the macroscopic model

**Idea:** Use the macroscopic event activity network

Determine *virtual activities* and “headways” for each macroscopic edge.

I.e. precedence constraints for events which all use the same part $b$ of the infrastructure:

$$\mathcal{E}(b) = \{ \text{events } i \text{ such that } a = (i, j) \text{ uses } b \}$$

This includes:

- simple headways constraints for each edge
- single track constraints
- dependencies between delays (Carla Conte)
Simple headway constraints

**Idea:** Look at edges instead of blocks.

Determine headway $Z_{ij}$ between two departures $i, j$ that prevents any block conflict.

If all trains have same speed:

$$Z(e) := \max \{ \text{driving time } b : b \text{ is block on edge } e \}$$

Add disjunctive constraints $x_j \geq x_i + Z_{ij}$ or $x_j \leq x_i - Z_{ji}$

$$|x_j - x_i - \frac{Z_{ij} - Z_{ji}}{2}| \geq \frac{Z_{ij} - Z_{ji}}{2} \text{ for all } i, j \in E(e).$$

If all trains have same speed:

$$|x_j - x_i| \geq Z(e) \text{ for all } i, j \in E(e).$$
Simple headway constraints

**Idea:** Look at edges instead of blocks.

Determine headway $Z_{ij}$ between two departures $i, j$ that prevents any block conflict

If all trains have same speed:

$$Z(e) := \max \{ \text{driving time } b : b \text{ is block on edge } e \}$$

Add disjunctive constraints $x_j \geq x_i + Z_{ij}$ or $x_j \leq x_i - Z_{ji}$

$$\left| x_j - x_i - \frac{Z_{ij} - Z_{ji}}{2} \right| \geq \frac{Z_{ij} - Z_{ji}}{2} \text{ for all } i, j \in \mathcal{E}(e).$$

If all trains have same speed:

$$\left| x_j - x_i \right| \geq Z(e) \text{ for all } i, j \in \mathcal{E}(e).$$
Simple headway constraints

**Idea:** Look at edges instead of blocks.

Determine headway $Z_{ij}$ between two departures $i, j$ that prevents any block conflict.

If all trains have same speed:

$$Z(e) := \max \{ \text{driving time } b : b \text{ is block on edge } e \}$$

Add disjunctive constraints $x_j \geq x_i + Z_{ij}$ or $x_j \leq x_i - Z_{ji}$

$$|x_j - x_i - \frac{Z_{ij} - Z_{ji}}{2}| \geq \frac{Z_{ij} - Z_{ji}}{2} \quad \text{for all } i, j \in \mathcal{E}(e).$$

If all trains have same speed:

$$|x_j - x_i| \geq Z(e) \quad \text{for all } i, j \in \mathcal{E}(e).$$
Adding headway constraints in the model

\[
\begin{align*}
\min \quad & \sum_{a=(i,j) \in A^s} w_a(y_j - y_i) + \sum_{a=(i,j) \in A_{umsteige}} Tw_ay_a \\
\text{s.d.} \quad & y_i \geq d_i \text{ for all delayed } i \\
& y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{wait} \cup A_{drive} \\
& -Mz_a + y_i - y_j \leq s_a \text{ for all } a = (i, j) \in A_{change} \\
& |y_i - y_j + \Pi_i - \Pi_j - z_{i,j}| \geq z_{i,j} \text{ for all } i, j \in E(e) \\
& y_i \in \mathbb{N}^0, \quad z_a \in \{0, 1\}
\end{align*}
\]
Graphical representation of headway constraints

In the event/activity network:

Choose exactly one of each pair of edges!
In the event-activity network:

Choose exactly one of each pair of edges!
As edge orientation problem

Other interpretation:

Orient the edges in such a way that no directed cycles appear!

Remark: Job-Shop scheduling with (variable) precedence constraints
As edge orientation problem

Other interpretation:

Orient the edges in such a way that no directed cycles appear!

**Remark:** Job-Shop scheduling with \((\text{variable})\) precedence constraints
When are the headway constraints necessary?

Order the events using the same edge $e$ according to scheduled times $\Pi_1 < \Pi_2 < \ldots \Pi_L$.

Lemma

The constraint $|y_i - y_j + \Pi_i - \Pi_j| \geq Z(e)$ is redundant, if one of the following conditions are satisfied:

- $y_i - y_{i+1} \leq P^i_e := \Pi_{i+1} - \Pi_i - Z_e$
  
  $P^i_e$ is called headway-slack

- $d_i \leq P_e := \min_{i \in \mathcal{E}(e)} P^i_e$

- $y_{i+1} \geq y_i$ für alle $i \in \mathcal{E}(e)$. 
A special case

Consider two trains using the same edge, no buffer times, all passengers get off at the next station.

\begin{itemize}
  \item train 1: departure delayed by $d$
  \item train 2: no source delay
\end{itemize}

Which train should go first?

\begin{itemize}
  \item if order is not changed: delay=$d_i + \max\{d_i - P_e^i, 0\}$
  \item if order is changed: delay=$\max\{d_i, \Pi_{i+1} - \Pi_i + Z_e\}$
\end{itemize}

Result: If $d_i \leq \Pi_{i+1} - \Pi_i$: do not change the order, otherwise change.
A special case

Consider two trains using the same edge, no buffer times, all passengers get off at the next station.

Which train should go first?

- if order is not changed: delay=$d_i + \max\{d_i - P^i, 0\}$
- if order is changed: delay=$\max\{d_i, \Pi_{i+1} - \Pi_i + Z_e\}$

Result: If $d_i \leq \Pi_{i+1} - \Pi_i$: do not change the order, otherwise change.
A special case

\[ \Pi_i - \Pi_{i+1} + Z_e \]

- do not change order
- change order

\[ p_e^i \]

\[ \Pi_i - \Pi_{i+1} \]

\[ p_e^i \]

\[ \Pi_i - \Pi_{i+1} + Z_e \]

\[ \text{delay} \]
A special case

\[ \Pi_i - \Pi_{i+1} + Z_e \]

Minimum

do not change order

change order

Delay Management for Railway Systems
Macroscopic approach
A special case

\[ p_i^e \]

\[ \Pi_i - \Pi_{i+1} \]

\[ \Pi_i - \Pi_{i+1} + Z_e \]

delay of train 1

delay of train 2
Heuristic FSFS for Re-Scheduling

If all wait-depart decisions are given:

- Remaining problem is re-scheduling problem
- Heuristic approach: “First Scheduled First Served” fixes the undirected red edges
- Solvable by simple adaption of CPM: For $i \in \mathcal{E}(e)$ do

$$x_i = \max \{\Pi_i + d_i, \max_{a=(j,i)} x_j + L_a, \max_{k \in \mathcal{E}(e)} x_k + Z_e\}.$$ 

Justification due to previous results (for delays $d_i$ small).

Include in Branch & Bound approach.
First Numerical results
### First Numerical results

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>107</td>
<td>107</td>
<td>107</td>
<td>107</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>167</td>
<td>167</td>
<td>167</td>
<td>167</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>212</td>
<td>212</td>
<td>212</td>
<td>220</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>212</td>
<td>212</td>
<td>212</td>
<td>220</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>131</td>
<td>131</td>
<td>131</td>
<td>131</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>131</td>
<td>131</td>
<td>131</td>
<td>131</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>224</td>
<td>224</td>
<td>232</td>
<td>281</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>284</td>
<td>284</td>
<td>292</td>
<td>341</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>345</td>
<td>440</td>
<td>515</td>
<td>521</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>645</td>
<td>740</td>
<td>815</td>
<td>821</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>174</td>
<td>174</td>
<td>174</td>
<td>174</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>524</td>
<td>524</td>
<td>526</td>
<td>529</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>704</td>
<td>704</td>
<td>706</td>
<td>709</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>564</td>
<td>564</td>
<td>580</td>
<td>637</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>624</td>
<td>624</td>
<td>640</td>
<td>697</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>651</td>
<td>723</td>
<td>796</td>
<td>801</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>951</td>
<td>1023</td>
<td>1096</td>
<td>1105</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>367</td>
<td>367</td>
<td>367</td>
<td>367</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>607</td>
<td>607</td>
<td>607</td>
<td>607</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>514</td>
<td>521</td>
<td>535</td>
<td>477</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>694</td>
<td>708</td>
<td>715</td>
<td>717</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>847</td>
<td>847</td>
<td>870</td>
<td>919</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1027</td>
<td>1027</td>
<td>1050</td>
<td>1099</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

**a** = no headway required  \( \geq 3 \) min

**b** = headway \( \geq 5 \) min

**c** = headway \( \geq 10 \) min

**d** = delay of vehicles

**Missed connections**

**Objective**
Arriving at the end . . .

You nearly made it!
Arriving at the end . . .

You nearly made it!
Literature

- Mixed-Integer Programming: Schöbel, Scholl, Kliewer, de Giovanni, Heilporn und Labbé
- Project networks: Ginkel und Schöbel
- Complexity: Gatto, Jakob, Peeters, Schöbel, Widmayer
- Simulation: Suhl, Kliewer, Biederbick, Mellouli, Ackermann, . . .
- Max-Plus-Algebra: van de Boom, De Schutter, Vries
Summary

What you have (hopefully!) seen in this talk:

- Integer programming model for delay management
- Bicriteria approach for delay management
- Modeling capacity constraints by headways

What else is going on?

- macroscopic level: Identify capacity constraints by stochastic investigation … Carla Conte
- microscopic level: Constraint branching … together with Ryan, Ehrgott, Larsen
- implement ideas within our project DisKon with Deutsche Bahn
- within European project ARRIVAL: Robust re-scheduling plans … Michael Schachtebeck
Summary

What you have (hopefully!) seen in this talk:

- Integer programming model for delay management
- Bicriteria approach for delay management
- Modeling capacity constraints by headways

What else is going on?

- macroscopic level: Identify capacity constraints by stochastic investigation
  
  ... Carla Conte

- microscopic level: Constraint branching
  
  ... together with Ryan, Ehrgott, Larsen

- implement ideas within our project DisKon with Deutsche Bahn

- within European project ARRIVAL: Robust re-scheduling plans
  
  ... Michael Schachtebeck
Summary

What you have (hopefully!) seen in this talk:

- Integer programming model for delay management
- Bicriteria approach for delay management
- Modeling capacity constraints by headways

What else is going on?

- macroscopic level: Identify capacity constraints by stochastic investigation ... Carla Conte
- microscopic level: Constraint branching ... together with Ryan, Ehrgott, Larsen
- implement ideas within our project DisKon with Deutsche Bahn
- within European project ARRIVAL: Robust re-scheduling plans ... Michael Schachtebeck
Good news for DisKon

DB Netz now accepted to use the first package of DisKon in the dispositive work (within a test environment).

If it works, the customer-oriented view will also be implemented in the disposition!
There is still a lot to do . . .
There is still a lot to do . . .

Thank you!