Integer Programming Models for the Delay Management Problem

Anita Schöbel

Universität Kaiserslautern

and

Fraunhofer Institut für Techno-und Wirtschaftsmathematik

AMORE Research Seminar 2002

Plan of the talk

The delay management problem

Application

Introduction

Minimizing the total delay

The model with constant weights

Algorithms for (TDM-const)

Conclusions

Our Application

Why delay management?

Project: together with two traffic associations (VRN and VGS) and DB Regio

supported by: Stiftung Innovation Rheinland-Pfalz

Application 1 :

On-line-decision for bus-drivers at the “Lautertalbahn”: wait for delayed trains or not?

Application 2 :

Find perturbed timetables in case of known disturbances (construction areas, etc.) in the Saarland.

Size of the problem

Testregion

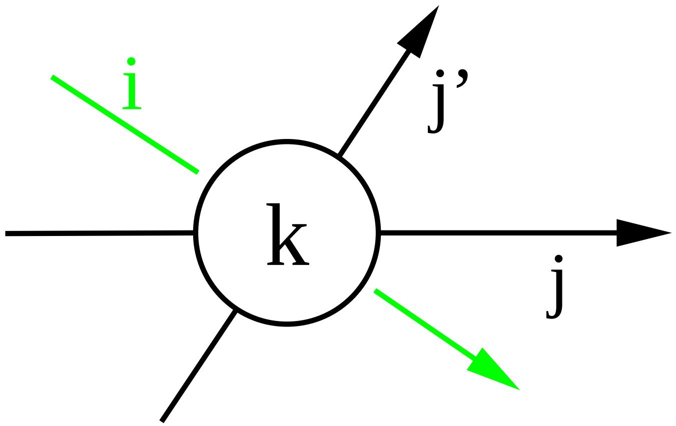
- **1100** stations
- **1350** vehicles within a day
- **11700** changing possibilities

resulting in a graph with

- **23000** nodes
- **25000** edges

Delay Management

Suppose that train i arrives at some station k with a delay. What should the connecting buses do?



Each bus j may either

WAIT and therefore cause delay for

- the customers within the bus j,
- the customers who want to get on the bus j, and,
- for subsequent other buses, or

DEPART and therefore accept that customers who want to change from the delayed train i onto the bus j miss their connection.

QUESTION: **WAIT** or **DEPART**?

Wait or depart?

Two extreme solutions:

All vehicles wait :

- No connection is missed!
- Many vehicles are delayed.

All vehicles depart in time :

- Minimizes the number of delayed vehicles.
- Many connections are missed.

A solution between these extremes:

Minimize the total delay!

What is the total delay?

Total delay is the sum of all delays over all customers.

Assumptions:

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

Consider a customer traveling on a path

$$p = (\text{station, vehicle, station, vehicle}, \ldots, \text{station}).$$

- If all connections are maintained: Delay of the customer is the arrival delay of its last vehicle at its last station
- If at least one connection is missed, the delay is T.

Consequence: To calculate the total delay, a *perturbed timetable* has also to be found.

(TDM)

Find **wait-depart decisions** and **a feasible perturbed timetable** for all vehicles at all stations such that the **total delay** over all customers is minimized.

- What is a feasible perturbed tinetable?
- How to calculate the total delay?

Literature review

For finding timetables in public transportation: Many papers, e.g., Adamski, Bookbinder, Bowman, Burkhard, Brucker, Cai, Carey, Ceder, Chmiel, Daduna, Desilet, Domschke, Egmont, Fleischmann, Goh, Goverde, Higgins, Holz, Hüttmann, Hurink, Klemt, Krista, Nachtigall, Odijk, Serafini, Stemme, Turnquist, Ukovich, Voget, Voß, Weigand . . .

(Stochastic) Investigation of delays: Chen and Harker (1990), Dauber (1986), Firpo and Savio (1997), Hall (1987), Higgins and Kozan (1998), Kohler and Letzbach (1974), Engelhardt-Funke (2002)

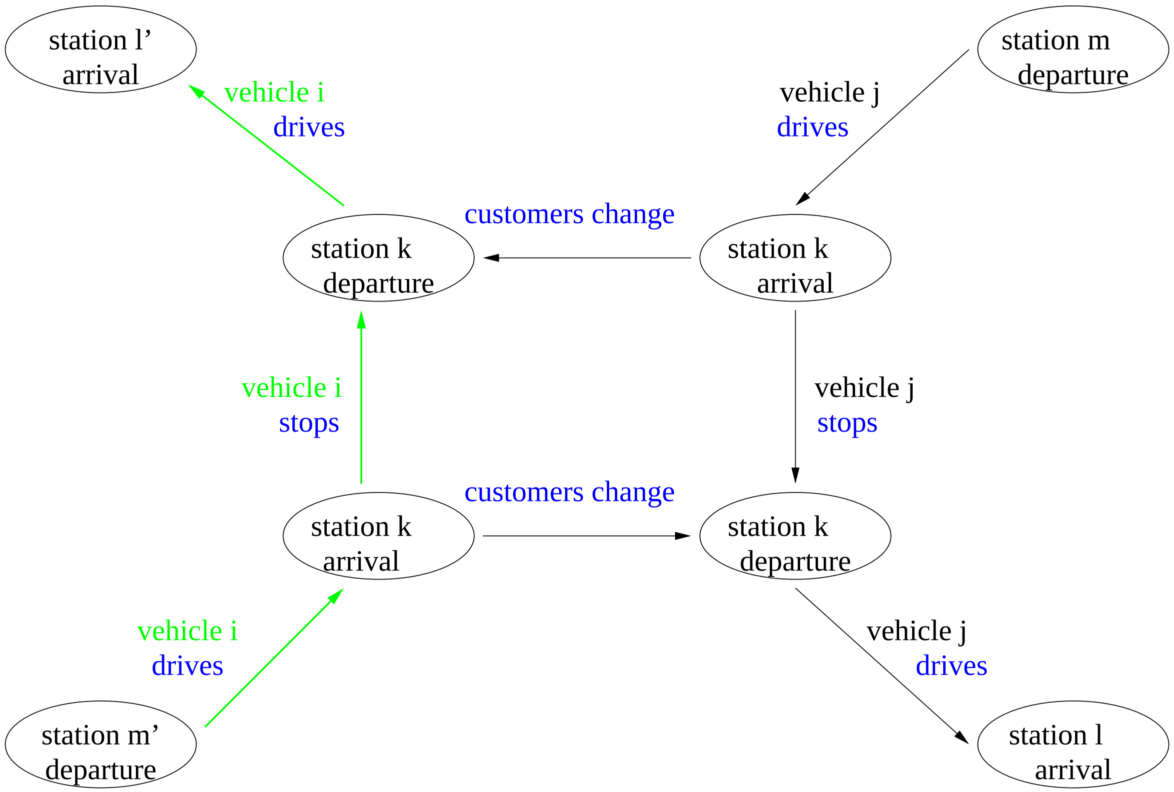
Delay management through simulation: Petersen and Taylor (1982), Ackermann (1999), Fay (1999), Suhl and Mellouli (1997), Suhl and Mellouli (1999), Kolonko et al. (1996), Kolonko and Engelhardt-Funke (1999)

Delay management through optimization: Adenso-Diaz et al. (1999), Kliewer (2000), Suhl et al (2001), Anderegg et al (2002)

Event-activity networks

see Nachtigall (1998)

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



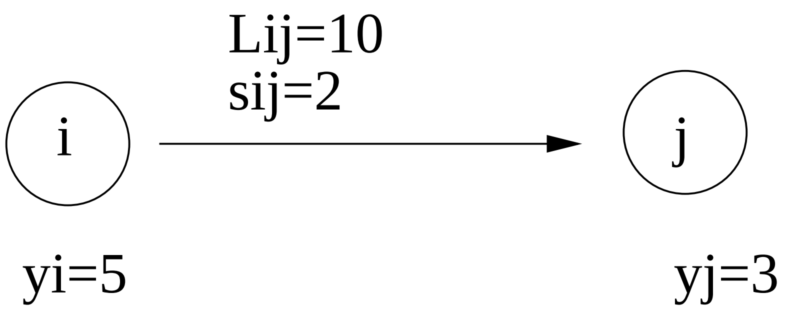
Variables and Paramters

Variables:

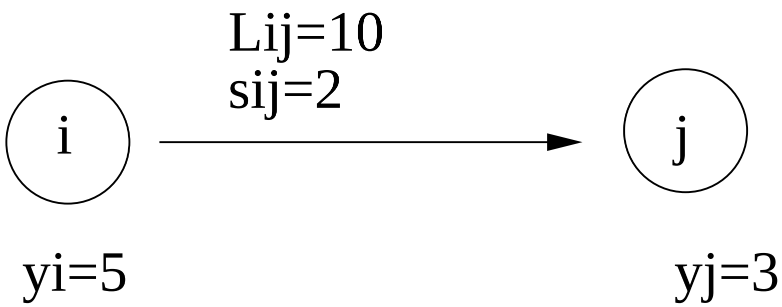
- y_i is delay of event $i \in \mathcal{E}$.
- $z_a = \begin{cases} 0 & \text{if connection } a \text{ is maintained} \\ 1 & \text{if connection } a \text{ is missed} \end{cases}$

Parameters :

- s_a slack time for all activities $a \in \mathcal{A}$.
- d_i source delays for all $i \in \mathcal{E}$ (maybe 0)
- $\mathcal{P}$ set of customers paths, given as a sequence of events.



Feasible perturbed timetable



A perturbed timetable is feasible, if :

- for all $a = (i, j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive}$ **and** all $a \in \mathcal{A}_{change}$ with $z_a = 0$:

$$y_i - y_j \leq s_a$$

- for all events:

$$y_i \geq d_i$$

Calculating the total delay

Idea: Calculate the delay on each activity and sum over all activities.

- The additional delay on activity a is given by the tension $y_j - y_i$ (can be positive or negative)
- The additional delay, if a is missed is $T - y_i = (T - y_j) + (y_j - y_i)$
- The delay of a path is the sum of all additional delays over its activities.

Define: w_a as the number of customers using activity a

The total delay is

$$\mathbf{f_{TDM}} = \sum_{a=(i,j)\in\mathcal{A}^s} w_a(y_j - y_i) + \sum_{a=(i,j)\in\mathcal{A}_{change}} w_a \bar{z}_a (T - y_j).$$

The Model (TDM)

$$\min \sum_{a=(i,j)\in\mathcal{A}^s} w_a(y_j - y_i) + \sum_{a=(i,j)\in\mathcal{A}_{change}} w_a \bar{z}_a (T - y_j)$$

$$
\begin{array}{lrcll}
\text{s.t.} & y_i & \geq & d_i & \text{for all } i \in \mathcal{D} \\
& y_i - y_j & \leq & s_a & \text{for all } a = (i,j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive} \\
& -M\bar{z}_a + y_i - y_j & \leq & s_a & \text{for all } a = (i,j) \in \mathcal{A}_{change} \\
& \tilde{z}_a^p + \displaystyle\sum_{\substack{\tilde{a}\in p\cap\mathcal{A}_{change}: \\ \tilde{a}\prec a}} \bar{z}_{\tilde{a}} & \geq & 1 & \text{for all } p \in \mathcal{P}^s \text{ and } a \in p \\
& \tilde{z}_a^p + \bar{z}_{\tilde{a}} & \leq & 1 & \text{for all } p \in \mathcal{P}^s \text{ and } a, \tilde{a} \in p \text{ with } \tilde{a} \prec a \\
& w_a & = & \displaystyle\sum_{p\in\mathcal{P}^s : a\in p} w_p \tilde{z}_a^p & \text{for all } a \in \mathcal{A}^s \\
& y_i, w_a & \in & \mathbb{N}^0 & \\
& z_a, \tilde{z}_a^p & \in & \{0,1\} &
\end{array}
$$

Model 2 is too complicated!!!

Idea: Fix the weights w_a as paramters by defining

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

i.e., assume that all customers can travel as they have planned.

(TDM-const)

$$\min \sum_{a=(i,j)\in\mathcal{A}^s} w_a(y_j - y_i) + \sum_{a=(i,j)\in\mathcal{A}_{change}} w_a \bar{z}_a (T - y_j)$$

such that

$$\begin{array}{lrcll}
\text{s.t.} & y_i & \geq & d_i & \text{for all } i \in \mathcal{D} \\
& y_i - y_j & \leq & s_a & \text{for all } a = (i,j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive} \\
& -M\bar{z}_a + y_i - y_j & \leq & s_a & \text{for all } a = (i,j) \in \mathcal{A}_{change} \\
& y_i & \in & \mathbb{N}^0 & \text{for all } i \in \mathcal{E} \\
& \bar{z}_a & \in & \{0,1\} & \text{for all } a \in \mathcal{A}^s
\end{array}$$

Note that

$$\sum_{a=(i,j)\in\mathcal{A}^s} w_a(y_j - y_i) = \sum_{i\in\mathcal{E}} w_i y_i$$

if we define

$$w_i = \sum_{\substack{p\in\mathcal{P}:\\ p \text{ ends at } i}} w_p.$$

This yields:

$$f_{TDM} = \sum_{i\in\mathcal{E}} w_i y_i + \sum_{a=(i,j)\in\mathcal{A}_{change}} w_a \bar{z}_a (T - \mathbf{y_j})$$

and forget about subtracting $\mathbf{y_i}$.

Consequence: A linear model!!

(TDM-const)

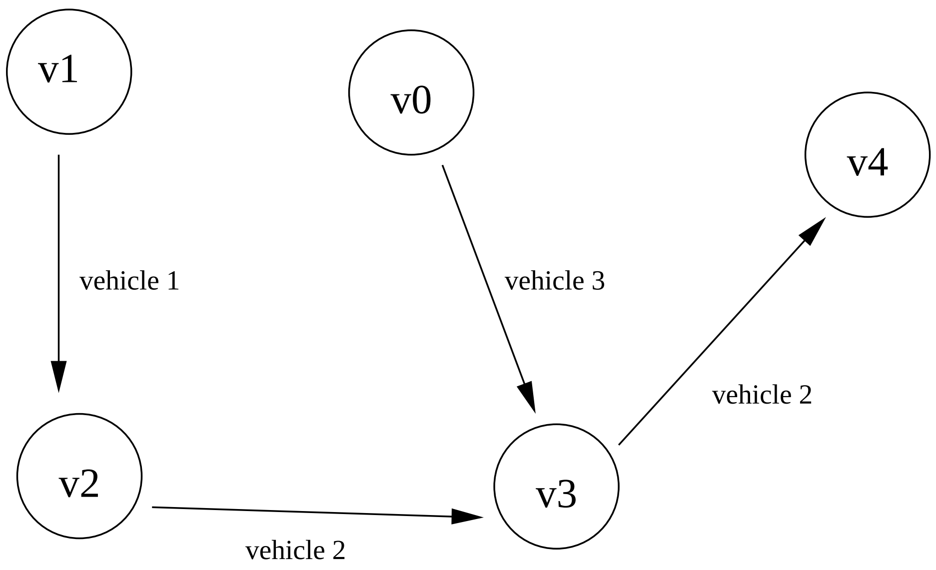
$$\min f_{TDM-const} = \sum_{i \in \mathcal{E}} w_i y_i + \sum_{a=(i,j) \in \mathcal{A}_{change}} w_a \bar{z}_a T$$

such that

$$\begin{array}{rcl} y_i & \geq & d_i \text{ for all } i \in \mathcal{D} \\ y_i - y_j & \leq & s_a \text{ for all } a = (i,j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive} \\ -M\bar{z}_a + y_i - y_j & \leq & s_a \text{ for all } a = (i,j) \in \mathcal{A}_{change} \\ y_i & \in & \mathbb{N}^0 \text{ for all } i \in \mathcal{E} \\ \bar{z}_a & \in & \{0,1\} \text{ for all } a \in \mathcal{A}^s \end{array}$$

The model (TDM-const) is in general wrong.

. . . not really surprisingly!

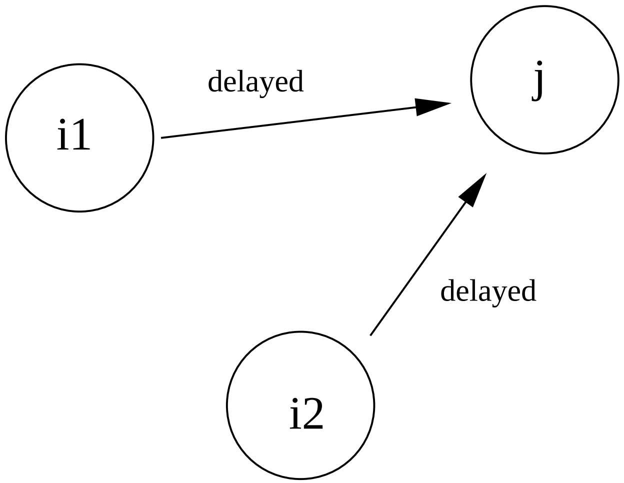


The never-meet-property

The delay management problem has the **never-meet-property** if in each (time-minimal) feasible solution with **zero slack times** for all $a \in \mathcal{A}$:

For all $j \in \mathcal{E}$:

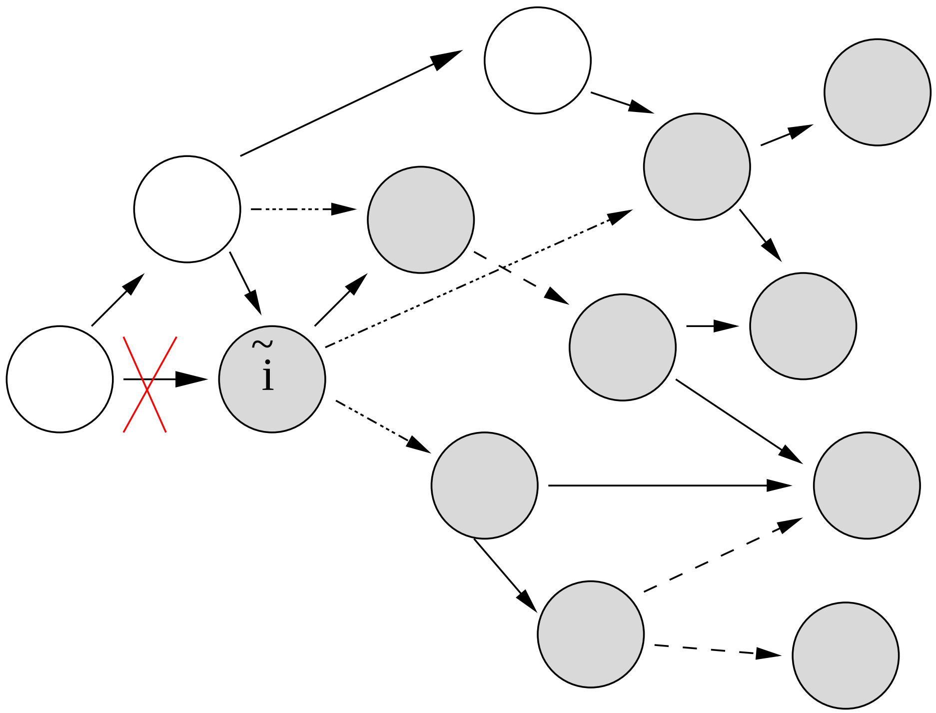
1. If $(i_1, j), (i_2, j) \in \mathcal{A}$, and $y_{i_2} > 0$ then $y_{i_1} = 0$.
2. If $(i_1, j) \in \mathcal{A}$, and $d_j > 0$ then $y_{i_1} = 0$.



Consequence of the never-meet-property

In all optimal solutions we have: If $\tilde{a} = (\tilde{i}, \tilde{j})$ and $z_{\tilde{a}} = 1$, then

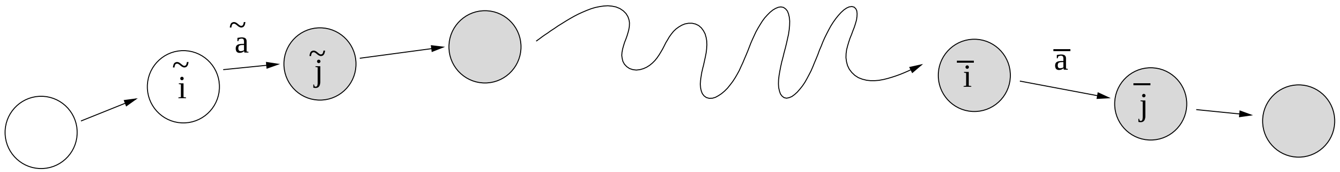
1. $y_i = 0$ for all i which can be reached from $\tilde{i}$.
2. $\bar{z}_a = 0$ for all a which can be reached from $\tilde{i}$.



Good news for (TDM-const)

Theorem: (TDM-const) is correct if the never-meet-property holds

Proof: If w_a is not the correct weight, the additional delay on a is zero.



More good news for (TDM-const)

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

In our data (1100 stations, 1350 vehicles, $\geq$ 10000 changing possibilities)

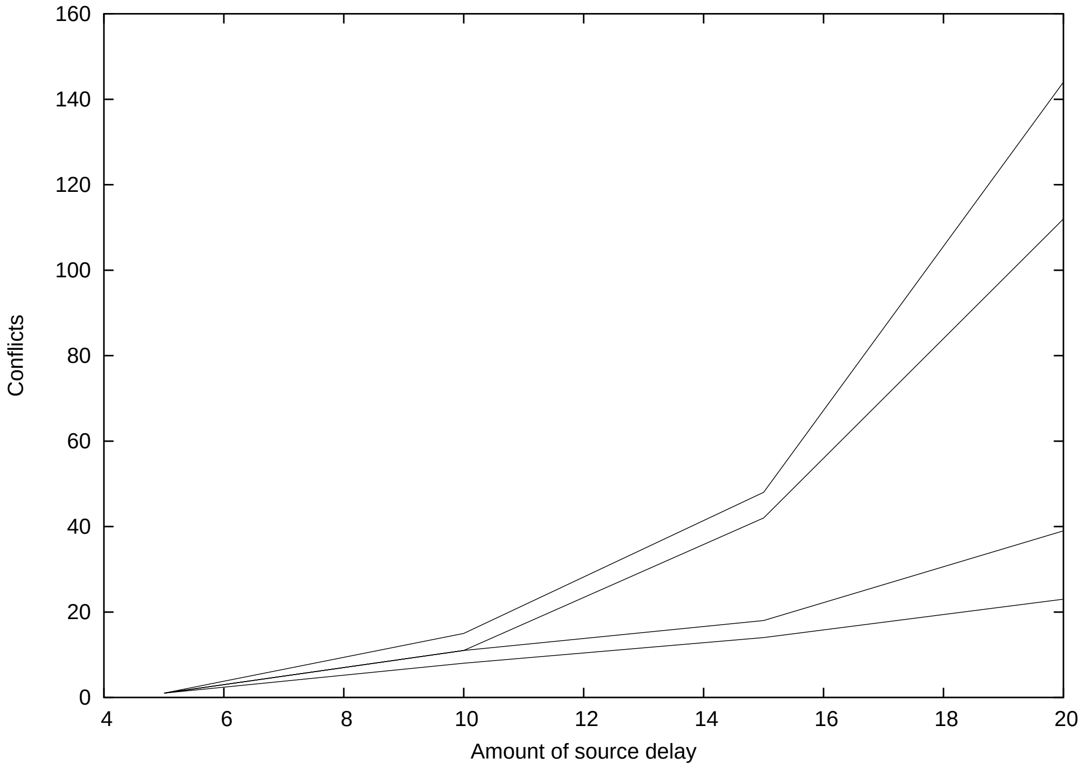
e.g.,

- 120 delayed vehicles
- sorce delay of 10 minutes each

results **in only 148 conflicts**!

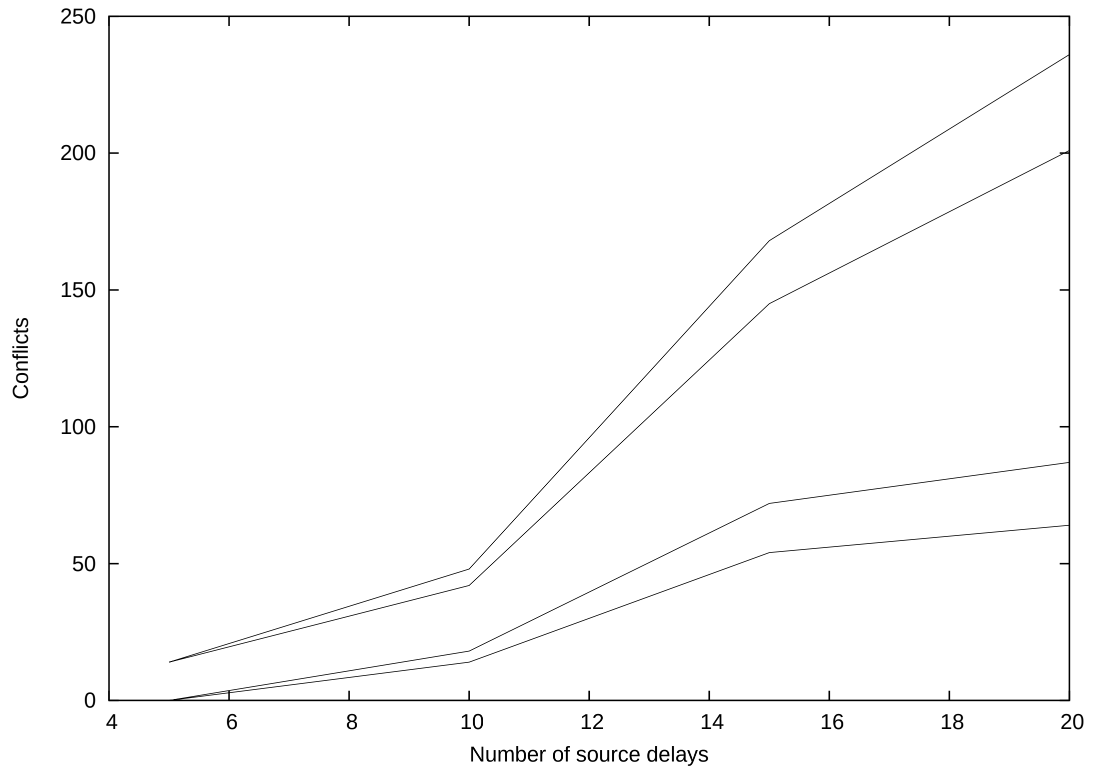
The never-meet-property in practice

Example: 10 delayed vehicles



The never-meet-property in practice

Example: source delay of 10 minutes



Solving (TDM-const) for zero slack times

see [Scholl, 01]

- coefficient matrix is totally unimodular
- Solve a linear program

Numerical results

- on a 750 MHz Pentium III processor, 128 MB RAM
- with XPRESS (optimiser version 11.14)
- Size of LP: 70730 variables, 68250 constraints
- Even for **200** delayed vehicles not more than 3 minutes!

Solving (TDM-const) for arbitrary slack times

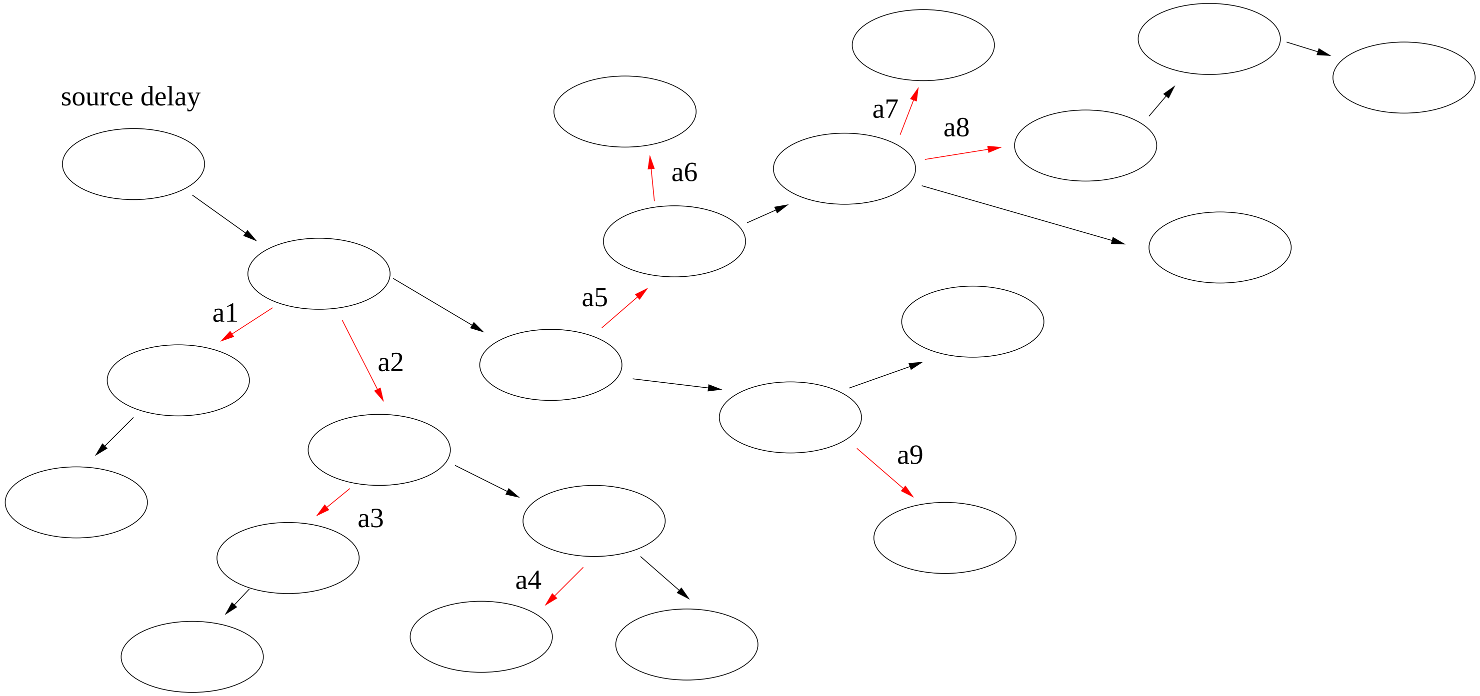
Theorem: (TDM-const) can be solved in O($|\mathcal{A}|$) time if the never-meet property holds.

Idea of the enumeration-algorithm

1. **Decompose:** At each changing activity a decompose into problems $a_1, \ldots, a_L$ where there is no other changing activites 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 $a_1, \ldots, a_L$, i.e.,
   - missing a costs $w_a T$
   - maintaining a costs $f_{a_1} + \ldots f_{a_L} + C(a)$.


source delay
a1
a2
a3
a4
a5
a6
a7
a8
a9

Summary

1. **Complicated** Model (TDM) to minimize the total delay
2. Simplification to a **linear integer program** (TDM-const), **but wrong?**
3. (TDM-const) **is correct** if the never-meet property holds!
4. The never-meet-property **can be tested efficiently**
5. (TDM-const) can be **solved in linear time** if the never-meet property holds.

Other aspects of delay management

- The enumeration algorithm can be extended to Branch and Bound for the general case (Schöbel, 2002)
- There exists a more general linear model for (TDM) (see Schöbel, 2001)
- Bicriterial delay management (see Ginkel and Schöbel, 2002) using methods of the DTCTP in the event-activity network
- Modeling railway specific requirements as fixed connections in alternative graphs, see (Schöbel, 2002)
- Allowing different time periods (see Schöbel and Scholl, 2002)