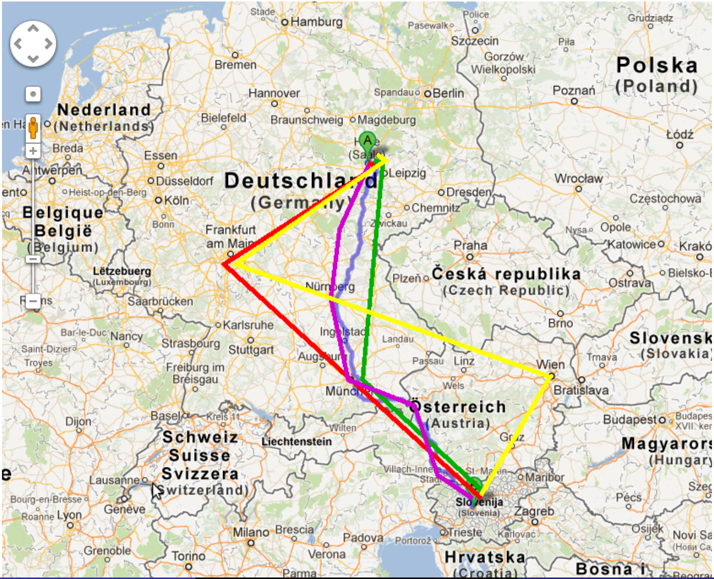
Algorithm Engineering of Timetable Information

Matthias Müller-Hannemann

Institut für Informatik
Martin-Luther-Universität Halle-Wittenberg

September 13, 2012

How To Travel?



Public Transport = Schedule-Based Travelling

Public Transport = Schedule-Based Travelling






- My own experience based on cooperation with Deutsche Bahn AG,
- so most of my examples will be about railways
- but application area covers all of public transport

From Printed Schedule Books to Fully Electronic Timetable Information

Timetable Information Systems: The Classical Use

Pre-trip information and selling:

- Selling of tickets without a timetable information system is nowadays impossible.
- Used by all distribution channels:
  at ticket counter, by ticket vending machines, by travel agencies, for Internet ticket sales.

Timetable Information Systems: The Classical Use

Pre-trip information and selling:

- Selling of tickets without a timetable information system is nowadays impossible.
- Used by all distribution channels:
  at ticket counter, by ticket vending machines, by travel agencies, for Internet ticket sales.
- Quality of recommended connection is crucial:

  Connections which the search engine does not offer will not be sold.

  Examples:

  - special fares, campaigns
  - extra trains
  - night trains
  - trains with seat reservation

Usage of Timetable Information Systems

Commercial state-of-the-art:

- (train) timetable information in Europe
  HAFAS: computes more than 60 million connections per day
  [www.hacon.de/hafas]
- $\rightarrow$ high relevance
- very fast, but
- only heuristic solutions
  often suboptimal, even with respect to travel time

Algorithmic Engineering enters ...

The driving question in 1999 was:

Can we do exact timetable information?
And can we do it efficiently enough?

Algorithmic Engineering enters ...

The driving question in 1999 was:

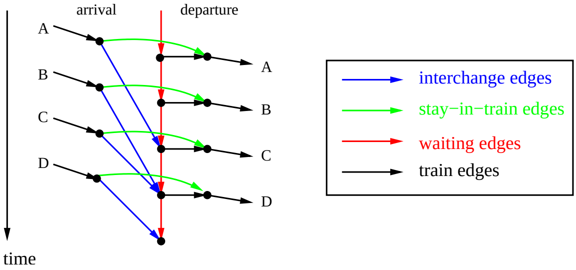
Can we do exact timetable information?
And can we do it efficiently enough?

Seminal paper by Frank Schulz, Dorothea Wagner, and Karsten Weihe, WAE 1999:

"Dijkstra's Algorithm On-Line: An Empirical Case Study from Public Railroad Transport."

Event-Activity Networks
Time-Expanded Network Model

Small excerpt at some station:



Properties:

- arrival/departure events correspond to vertices
- feasible connections correspond to directed paths

Earliest Arrival Problem

Earliest arrival problem

Input: source station, destination station, start time, event-activity network

Task: Find the connection with earliest arrival time at destination

In other words:

- have to solve an s-t-shortest path problem in acyclic digraph
- solvable in linear time $O(m+n)$
  (m number of arcs, n number of vertices)

Earliest Arrival Problem

Earliest arrival problem

Input: source station, destination station, start time, event-activity network

Task: Find the connection with earliest arrival time at destination

In other words:

- have to solve an s-t-shortest path problem in acyclic digraph
- solvable in linear time $O(m+n)$ (m number of arcs, n number of vertices)
- but graphs are fairly large (several millions of vertices)
- we definitely need sublinear algorithms
- $\rightarrow$ development of speed-up techniques

Success Story of Algorithm Engineering: Classical *s*-*t* Shortest Paths

Road networks:

- millions of nodes (continental size)
- query times of few microseconds achievable
- many new techniques:
  overlay graphs, contraction hierarchies, shortcuts and arc-flags, transit node routing, hub labeling, ...

Timetable information in public transport:

- considerable speed-ups, although much less effective
- but achievable milliseconds are fine from a practical point of view

Example

Query: from Halle (Saale) Hbf to Stuttgart Hbf at 08:00 on Wednesday September 12, 2012

Earliest arrival connection:

Halle (Saale) Hbf	departure	8:00
Naumburg (Saale) Hbf	arrival	8:29
Naumburg (Saale) Hbf	departure	8:35
Fulda	arrival	10:42
Fulda	departure	10:47
Stuttgart Hbf	arrival	13:08

Earliest arrival time: 5h 08 minutes

Example: Real Query

DB BAHN

Contact | Help | Sitemap a a+ a++

Home | Offers | Destinations | Services | Trains | Tourism | About DB Bahn My Bahn

Search | Selection | Ticket&Reservation | Payment | Booking | Confirmation

Your trip information

Outward journey from Halle(Saale)Hbf We, 12.09.12
to Stuttgart Hbf 07:10 Departure Arrival

Refresh

Travellers 1 adult, 2nd class

Change other data New query

Print your ticket at home

With the DB online ticket, it is quick and easy to print your ticket on your own PC up until shortly before your train departs.

Further information

Please choose a connection.

Selection outward journey - Selection Departure

Print view

Station/Stop	Date	Time / prognosis	Duration	Chg.	Products	Price for all travellers*	
		Earlier				Standard fare	
Halle(Saale)Hbf Stuttgart Hbf	We, 12.09.12 We, 12.09.12	dep 07:07 arr 13:08	6:01	1	IC, ICE	Connection is in the past	Choose return trip
Halle(Saale)Hbf Stuttgart Hbf	We, 12.09.12 We, 12.09.12	dep 08:00 arr 13:08	5:08	2	ICE, IC	102,00 EUR Purchase	Choose return trip
Halle(Saale)Hbf Stuttgart Hbf	We, 12.09.12 We, 12.09.12	dep 08:00 arr 13:53	5:53	1	ICE, IC	89,00 EUR Purchase	Choose return trip
Show details for all		Later					

Mobility Check **Environmental Mobility Check**

Change query

* All of the travellers and reduced rate tickets were taken into consideration in the price shown.

Important information about this connection is currently available (e.g. connection is threatened, train was cancelled). Please note the information in the details.

Not bookable via the internet.

All information is issued without liability.

For further offers, visit our ticket shop .

Example: Real Query

Selection outward journey - Selection Departure

Station/Stop	Date	Time / prognosis		Duration	Chg.	Products	Price for all travellers*
		↑ Earlier					Standard fare
Halle(Saale)Hbf Stuttgart Hbf	We, 12.09.12 We, 12.09.12	dep arr	07:07 13:08	6:01	1	IC, ICE	Connection is in the past
Halle(Saale)Hbf Stuttgart Hbf	We, 12.09.12 We, 12.09.12	dep arr	08:00 13:08	5:08	2	ICE, IC	102,00 EUR → Purchase
Halle(Saale)Hbf Stuttgart Hbf	We, 12.09.12 We, 12.09.12	dep arr	08:00 13:53	5:53	1	ICE, IC	89,00 EUR → Purchase
Show details for all		↓ Later					

Example: Real Query

What is Wrong?

- Too simple model:
  - start time interval
  - further search attributes
  - missing footpaths
- too space-consuming model
- only single-criterion search
- no alternatives
- no delays

Overview

1. Realistic models
2. Multi-criteria search
3. Realtime information
4. Reliable connections (robustness, stochastic forecasts)
5. Extensions
   - train disposition
   - multi-modal search

Towards More Realistic Models

Start time intervals:

- easy extension (standard trick in network flows)
- just add an artificial source vertex s
- add arcs, connecting the source with all departure vertices within the departure interval
- give these arcs zero length
- do shortest path computation from s
- Warning:
  some subtleties in combination with multi-criteria search!

Towards More Realistic Models

Further search attributes:

- bicycle transportation, on-board restaurant, wheel chair access, seat reservation possible, ...
- again an easy extension possible
- equip each travel arc with a bit vector of the attributes it serves
- during search simply ignore all arcs which don't have the required attributes

Alternative Graph Models

- time-expanded graph model uses one vertex per event

Can we do better if we use only one vertex per station (stop, airport)?

Alternative Graph Models

- time-expanded graph model uses one vertex per event

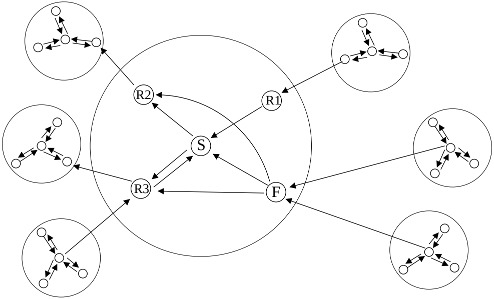
Can we do better if we use only one vertex per station (stop, airport)?

(Simple) Time-Dependent Graph Model:

- every vertex represents a station
- two vertices are connected by an arc if the corresponding stations are connected by a direct non-stop connection
- lengths on the arcs are determined "on-the-fly":
  the length of an arc depends on the time in which the particular arc will be used

Time-Dependent Graph Model

Small excerpt:



- different node types:
  train route nodes (Rx), station node (S), foot-node (F)
- train edges have variable length
  (depending on the moment of time when they are used)

Brief Discussion of Models

Time-expanded model:

- pro: easier to extend to realistic setting
- contra: higher memory + slower performance

Time-dependent model:

- pro: smaller memory + better performance
- contra: fairly complicated extension to multi-criteria

For more details see survey by M.-H., Schulz, Wagner, Zaroliagis 2007.

Overview

1. Realistic models
2. **Multi-criteria search**
3. Realtime information
4. Reliable connections (robustness, stochastic forecasts)
5. Extensions
   - train disposition
   - multi-modal search

Multi-Criteria Search

Typical criteria:

- travel time
- number of transfers
- fare
- reliability (maximize minimum buffer time)

Multi-Criteria Search

Typical criteria:

- travel time
- number of transfers
- fare
- reliability (maximize minimum buffer time)

Difficulties with advanced criteria:

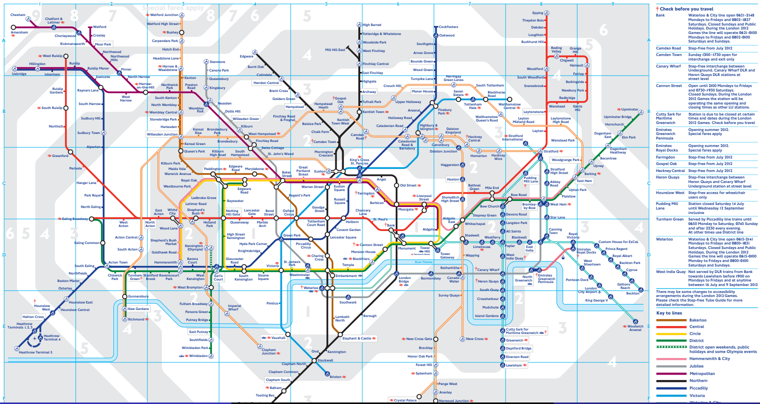
- non-linear or non-additive (for example, fares)
- expensive to evaluate
  black-box optimization (for example, fares)
- strong negative correlation (for example, travel time vs. reliability)

Fare Zones in London

Fare-optimal paths:

- bit-vectors indicate which zones are used
- apply dominance rules during search [DPW12]

Tube map



Pareto-Optimality

- Standard notion of dominance:
  Definition: A dominates B if A is at least as good as B in all criteria and strictly better in at least one.
- Pareto-optimal solutions are the non-dominated ones.
- Basic algorithmic problem with several objectives:

Find all Pareto-optimal solution values.

or

Find all Pareto-optimal paths.

Size of Pareto Set

- In general, the Pareto set can be exponentially large.
- In practice, depending on the kind of criteria, it is often manageable [M.-H., Weihe 2006].
- Some Pareto optima irrelevant for practical purposes.
- But what is an appropriate size?
  Who wants to see 17 alternatives?

Attractive Alternatives

- Pareto-optimality excludes "near-optimal" paths which often are very reasonable and attractive alternatives.
- Consequence: we need some kind of relaxed Pareto-dominance. [M.-H., Schnee 2004]
- Further consequence: using relaxed Pareto-dominance restricts the set of applicable speed-up techniques.

Multi-Criteria Dijkstra Algorithm

- generalization of Dijkstra's algorithm to several criteria
- uses multi-dimensional labels (representing partial solutions)
- efficiency depends on effective dominance rules

Multi-Criteria Dijkstra Algorithm

Remarks:

- Can also be adapted to time-dependent network model
- Main difficulty: need to ensure “substructure optimality”
- otherwise, domination goes wrong!
- See [Annabell Berger, M.-H. 2008]

A Dilemma

- preprocessing tries to compute additional information (at most linear extra memory) which can be used to reduce the search space for queries
- most speed-up technique work only well if we look for a single optimal solution
- the more aggressive the speed-up technique is able to reduce the search space, the less likely it is that it can find good alternatives

Techniques that do not work as well as expected ...

We tested several preprocessing techniques in a multi-criteria setting for time-dependent networks: [BDGM09]

- short-cuts
- arc-flags

$\rightarrow$ they yield only comparably small speed-ups

Transfer Patterns (Bast et al., ESA 2010)

What is special in public transportation networks?

- Even when you take a very long trip, the number of transfers is almost always a very small number
- And more than that, for a given source and destination, there is only a very limited number of "patterns" where it makes sense to transfer
- "Pattern" $=$ sequence of stations where a transfer occurs

Transfer Patterns (Bast et al., ESA 2010)

Idea: for each pair of stations, precompute all transfer patterns of all optimal paths (at all times) and store them

Query algorithm: do a time-dependent Dijkstra computation on this so-called query graph, where each arc evaluation is again a shortest path query, but restricted to direct connections

Such direct-connection queries are easy

Transfer Patterns (Bast et al., ESA 2010)

Idea: for each pair of stations, precompute all transfer patterns of all optimal paths (at all times) and store them

Query algorithm: do a time-dependent Dijkstra computation on this so-called query graph, where each arc evaluation is again a shortest path query, but restricted to direct connections

Such direct-connection queries are easy

Remarks:

- Preprocessing is quite expensive, but doable
- successful tests with networks in Switzerland, New York area, and a large part of North America
- (You may try yourself: http://www.google.com/transit)

Round-Based Search

RAPTOR - Round-bAsed Public Transit Optimized Router
(Delling, Pajor, Werneck, ALENEX 2012)

Idea:

- routes = lines
  (sequence of stops served by the same means of transport)
- operates in rounds, one per transfer
- round k computes the fastest way of getting to every stop with at most $k-1$ transfers
- computes arrival times by traversing every route at most once per round

Round-Based Search

RAPTOR - Round-bAsed Public Transit Optimized Router
(Delling, Pajor, Werneck, ALENEX 2012)

Advantages:

- simple data structures and excellent memory locality
- easy to parallelize
- no preprocessing necessary

Tested successfully for London's public transport system

Overview

1. Realistic models
2. Multi-criteria search
3. **Realtime information**
4. Reliable connections (robustness, stochastic forecasts)
5. Extensions
    - multi-modal search
    - train disposition

Delays and Cancellations

Various causes













Realtime Train Information

Considers current traffic situation:

- train cancellations, extra trains
- delay messages for all trains
- dispositions of the central train traffic management
  (connection train will wait/will not wait)

Realtime Train Information

Considers current traffic situation:

- train cancellations, extra trains
- delay messages for all trains
- dispositions of the central train traffic management
  (connection train will wait/will not wait)

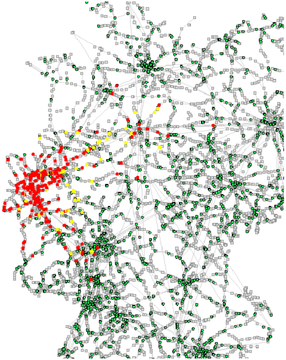
"On-Trip Scenario":

You are already on the way —
How to find the best continuation
to your destination?

chaos on a strike day

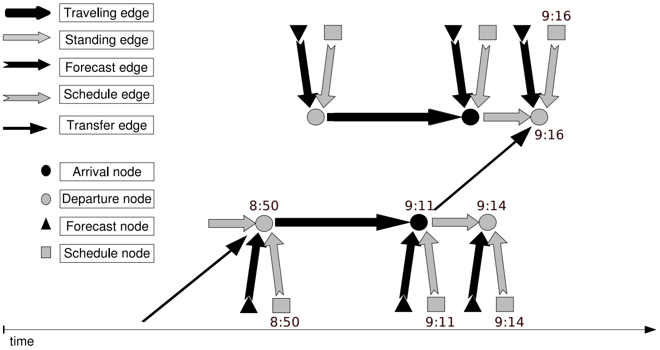
Secondary Delays

- Decision "train A waits for train B" generates cascade of secondary delays for other trains
- Depending on waiting policy possibly many additional delays
- Revocation of wait decisions requires undo process



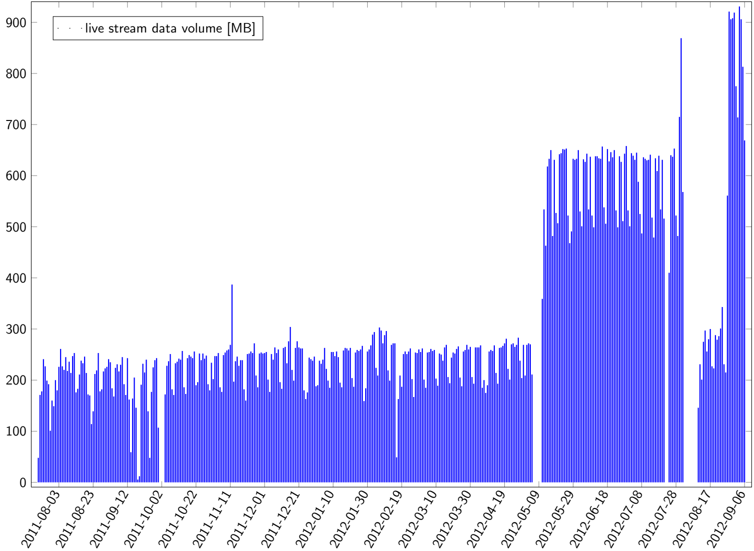
Static Delay Propagation

Event-Activity-Based Dependency Model
[M.-H., Schnee 2009]



“Realtime” Timestamp of event = maximum over all incoming dependencies

Massive Delay Streams



Typical Delay Message (simplified)

```xml
<Paket TOut="20120602195644533">
 <ListNachricht>
  <Nachricht>
   <Ist>
    <Service Id="80031551" IdZNr="1744" IdZGattung="ICE"
             IdBf="DH" IdBfEvaNr="8010085" IdZeit="20120602142300">
     <ListZug>
      <Zug Nr="1744" Gattung="ICE">
       <ListZE>
        <ZE Typ="Ab">
         <Bf Code="HB" EvaNr="8000050" Name="Bremen␣Hbf" />
         <Zeit Soll="20120602195500" Ist="20120602195600" />
        </ZE>
       </ListZE>
      </Zug>
     </ListZug>
    </Service>
   </Ist>
  </Nachricht>
 </ListNachricht>
</Paket>
```

Delay Propagation

Our prototype: (M.-H., Schnee 2009)

- graph size: 1,000,000 nodes (German train schedule)
- can handle massive data streams (6 million messages per day)
- time per operation: $< 1ms$ per update

Conclusion:

Handling (static!) delay information is not the computational bottleneck

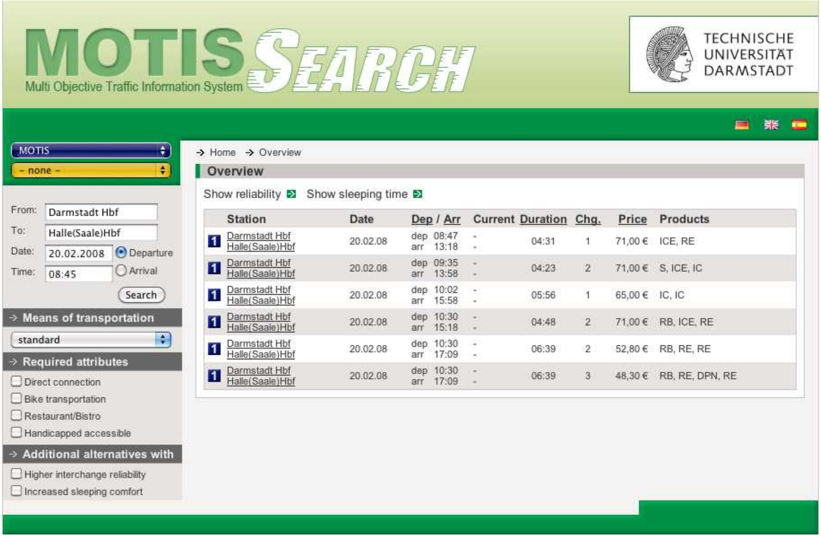
MOTIS (developed with Mathias Schnee)

- MOTIS is an abbreviation for
  Multi-Objective Traffic Information System
- is fully realistic
- its core includes a Dijkstra-based multi-criteria search algorithm

MOTIS (developed with Mathias Schnee)

- MOTIS is an abbreviation for
  Multi-Objective Traffic Information System
- is fully realistic
- its core includes a Dijkstra-based multi-criteria search algorithm
- minimizes exactly travel time and number of interchanges
- delivers many attractive alternatives using our concept of relaxed Pareto optimality
- MOTIS is easily extendable to further objectives
  (like train reservation, buffer times between train changes, ...)
- $>$ 100k lines of C++ code

MOTIS — Example



Station	Date	Dep / Arr	Current	Duration	Chg.	Price	Products
Darmstadt Hbf Halle(Saale)Hbf	20.02.08	dep 08:47 arr 13:18	- -	04:31	1	71,00 €	ICE, RE
Darmstadt Hbf Halle(Saale)Hbf	20.02.08	dep 09:35 arr 13:58	- -	04:23	2	71,00 €	S, ICE, IC
Darmstadt Hbf Halle(Saale)Hbf	20.02.08	dep 10:02 arr 15:58	- -	05:56	1	65,00 €	IC, IC
Darmstadt Hbf Halle(Saale)Hbf	20.02.08	dep 10:30 arr 15:18	- -	04:48	2	71,00 €	RB, ICE, RE
Darmstadt Hbf Halle(Saale)Hbf	20.02.08	dep 10:30 arr 17:09	- -	06:39	2	52,80 €	RB, RE, RE
Darmstadt Hbf Halle(Saale)Hbf	20.02.08	dep 10:30 arr 17:09	- -	06:39	3	48,30 €	RB, RE, DPN, RE

MOTIS — Example

Applications of Realtime Train Information

Application I: at service points

- Used by experienced staff to guide passengers.

Application II: feasibility check and rerouting

- Service provider constantly checks feasibility of planned connections.
- Of course, this assumes that a customer is willing to tell his travel plans to the provider.
- If some planned connection becomes infeasible (for what reason so ever) the customer is informed by a short message (SMS).
- The message does not only tell this fact but gives a recommendation for alternative routes.
- This recommendation can be tailored to the preferences of the customer.

MOTIS — Example

CoCoAS
Connection Controller & Alternatives System

TECHNISCHE UNIVERSITÄT DARMSTADT

Logout

→ Home

Original connection

Station	Date	Time	Current	Dur	Changes	Products	Status	State at
Kaiserslautern Hbf Mönchengladbach Hbf	21.02.2008	07:37 12:16	07:38	04:38	2	RB, IC, RE	✗	11:11

Alternatives

Station	Date	Time	Current	Dur	Changes	Products	Status	State at
Kaiserslautern Hbf Mönchengladbach Hbf	21.02.2008	07:37 12:26	07:38 12:27	04:49	3	RB, IC, RE, S	!	11:11
Kaiserslautern Hbf Mönchengladbach Hbf	21.02.2008	07:37 12:20	07:38 12:28	04:50	2	RB, IC, RE	!	11:11
Kaiserslautern Hbf Mönchengladbach Hbf	21.02.2008	07:37 12:46	07:38	05:08	2	RB, IC, S	✓	11:11
Kaiserslautern Hbf Mönchengladbach Hbf	21.02.2008	07:37 12:48	07:38	05:10	2	RB, IC, RE	✓	11:11

Server time: 11:11

Transfer into Practice

DB BAHN

Kontakt | Hilfe | Sitemap a a+ a++

Startseite | Angebotsberatung | **Fahrplan & Buchung** | Services | BahnCard | Urlaub   Meine Bahn

Ermittelte Verbindungen gemäß Live Auskunft.

Betaversion

Ihr Feedback

Ihre Verbindungen wurden auf Basis der aktuellen Verkehrslage berechnet, die sich jederzeit ändern kann. Wir empfehlen, die Verbindung ggf. kurz vor Reisebeginn nochmals anzufragen. Alle Angaben sind ohne Gewähr.
Bitte beachten Sie, dass Ihr Ticket (z.B. mit Zugbindung) nicht grundsätzlich die Nutzung aller Alternativen einschließt. Bitte wenden Sie sich ggf. an das Servicepersonal vor Ort.
Teilen Sie uns Ihre Erfahrungen und Wünsche zu unserem neuen Service mit. → Mehr Informationen

Ihre gewählte Verbindung nach Fahrplan

Bahnhof/Haltestelle	Datum	Zeit / Prognose		Umst.	Produkte	Bewertung
Halle(Saale)Hbf Stuttgart Hbf	Fr, 07.09.12 Fr, 07.09.12	ab an	08:00 +10 13:08 +0	2	ICE, IC	

Bahnhof/Haltestelle	Datum	Zeit / Prognose		Gleis	Produkte	
Halle(Saale)Hbf	Fr, 07.09.12	ab	08:00 +10	3	ICE 1105	Intercity-Express
Naumburg(Saale)Hbf	Fr, 07.09.12	an	08:29 +10	2		Bordbistro
			Der Anschluss wird vsl. nicht erreicht.			
Naumburg(Saale)Hbf	Fr, 07.09.12	ab	08:35 +0	3	IC 2252	Intercity
Fulda	Fr, 07.09.12	an	10:42 +0	3		Fahrradmitnahme reservierungspflichtig, Fahrradmitnahme begrenzt möglich, Bordbistro
			Es liegen derzeit keine ausreichenden Informationen zum Anschluss vor.			
Fulda	Fr, 07.09.12	ab	10:47 +1	4	ICE 595	Intercity-Express
Stuttgart Hbf	Fr, 07.09.12	an	13:08 +0	16		Bordrestaurant

Zwischenhalte einblenden

fährt nicht täglich, Verkehrstage

Am Bahnhof   Karte anzeigen

Transfer into Practice

Beta-version of realtime information of German Railways

Disclaimer

Your connections were calculated on the basis of the current traffic situation, which can change at any time.

...

Please be aware that your ticket (e.g. with the requirement to use a specific train) does not necessarily include the use of all alternatives.

Transfer into Practice

DB BAHN

Kontakt | Hilfe | Sitemap a a+ a++

Startseite | Angebotsberatung | **Fahrplan & Buchung** | Services | BahnCard | Urlaub Meine Bahn

Ermittelte Verbindungen gemäß Live Auskunft.

Betaversion

Ihr Feedback

Ihre Verbindungen wurden auf Basis der aktuellen Verkehrslage berechnet, die sich jederzeit ändern kann. Wir empfehlen, die Verbindung ggf. kurz vor Reisebeginn nochmals anzufragen. Alle Angaben sind ohne Gewähr.
Bitte beachten Sie, dass Ihr Ticket (z.B. mit Zugbindung) nicht grundsätzlich die Nutzung aller Alternativen einschließt. Bitte wenden Sie sich ggf. an das Servicepersonal vor Ort.
Teilen Sie uns Ihre Erfahrungen und Wünsche zu unserem neuen Service mit. → Mehr Informationen

Ihre gewählte Verbindung nach Fahrplan

Bahnhof/Haltestelle	Datum	Zeit / Prognose	Umst.	Produkte	Bewertung
Halle(Saale)Hbf Stuttgart Hbf	Fr, 07.09.12 Fr, 07.09.12	ab 08:00 +10 an 13:08 +0	2	ICE, IC	

Bahnhof/Haltestelle	Datum	Zeit / Prognose		Gleis	Produkte	
Halle(Saale)Hbf	Fr, 07.09.12	ab	08:00 +10	3	ICE 1105	Intercity-Express
Naumburg(Saale)Hbf	Fr, 07.09.12	an	08:29 +10	2		Bordbistro
			Der Anschluss wird vsl. nicht erreicht.			
Naumburg(Saale)Hbf	Fr, 07.09.12	ab	08:35 +0	3	IC 2252	Intercity
Fulda	Fr, 07.09.12	an	10:42 +0	3		Fahrradmitnahme reservierungspflichtig, Fahrradmitnahme begrenzt möglich, Bordbistro
			Es liegen derzeit keine ausreichenden Informationen zum Anschluss vor.			
Fulda	Fr, 07.09.12	ab	10:47 +1	4	ICE 595	Intercity-Express
Stuttgart Hbf	Fr, 07.09.12	an	13:08 +0	16		Bordrestaurant

Zwischenhalte einblenden

fährt nicht täglich, Verkehrstage

Am Bahnhof **Karte anzeigen**

Example: Real Query

Overview

1. Realistic models
2. Multi-criteria search
3. Realtime information
4. **Reliable connections (robustness, stochastic forecasts)**
5. Extensions
   - train disposition
   - multi-modal search

Robust Timetable Information

[ATMOS 2011, jointly with Marc Goerigk, Marie Schmidt, Anita Schöbel (Göttingen) and Martin Knoth (MLU)]

- **Goal:** plan your journey such that you will reach all your transfers regardless of potential delays
- **Scenarios:** we have to define scenario sets specifying potential delays (uncertainty sets)
- **Strictly robust optimization:** determine the fastest route such that all included transfers will be reached for **every** scenario

Uncertainty Sets

- scenario described as a vector d of dimension # travel and waiting arcs
- each entry specifies by how much the travel or waiting activity is delayed (primary delays)
- our model: very few large delays (parameter K), arbitrarily many small delays
- small delays: $\leq \epsilon_a$
- large delays on arc a: between ϵ_a and $d_a^{\max}$
- $$U := U_\epsilon^K := \{d \in \mathbb{R}^{|A_{wait} \cup A_{drive}|} : 0 \leq d_a \leq d_a^{\max} \text{ for all } a \in A_{wait} \cup A_{drive}, \\ |\{a \in A : d_a > \epsilon_a\}| \leq K\}.$$

Results

Given: a set of delay scenarios and an event-activity network (i.e., the schedule)

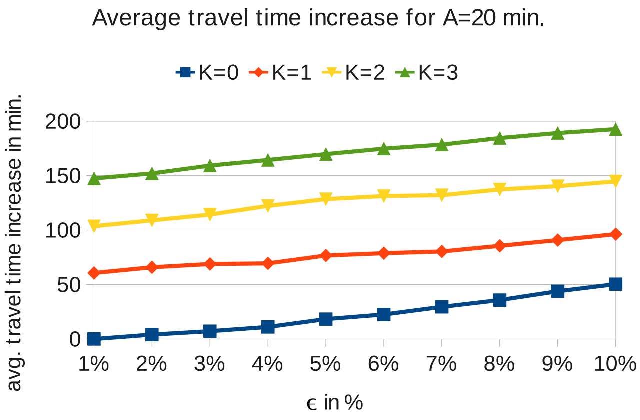
Goal: identify all transfer activities which will never fail in the given scenario set (i.e., "strictly robust transfers")

Afterwards: Do timetable information subject to the network restricted to strictly robust transfers (i.e., forbid all other transfers)

Main results:

- To decide whether a transfer arc is strictly robust is NP-hard
- Dynamic programming can be used to solve the problem heuristically (may classify some transfer arcs erroneously as non-robust)

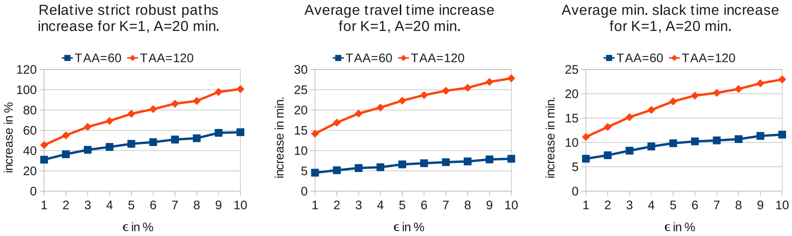
Strict Robustness



Way too restrictive!

Price of “Light Robustness”

Goal: minimize the number of “non-robust” transfers subject to an upper bound on the travel time: earliest arrival time $+ x$ minutes (in the figure: $x =$ TAA)



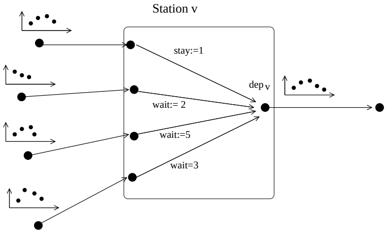
increase of strictly robust paths (in %, left),
increase of average travel time (center) and
increase of minimum buffer time for transfers in the chosen light robust path in comparison to the nominal scenario (right)

Stochastic Delay Propagation

[ATMOS 2011, with Annabell Berger, Andreas Gebhardt and Martin Ostrowski]

Drawbacks of static propagation:

- Assumes constant travel times (as scheduled!)
- Fluctuations and catch-up potential ignored (different speeds, track conflicts ...)
- Stochastic online model



Main achievement:

- stochastic forecasts possible within few seconds,
- but further investigations with empirical delay data necessary

Model Assumptions

General scenario:

- arbitrary discrete distributions
- stream of online messages about the delay status of trains
- "is"-messages about what has been realized

Model Assumptions

General scenario:

- arbitrary discrete distributions
- stream of online messages about the delay status of trains
- "is"-messages about what has been realized

Assumption 1: With respect to status messages, a train can arrive or depart at any time after the planned arrival or departure time, respectively.

Assumption 2: With respect to our forecasts of arrival and departure time distributions, no train departs before its scheduled time or arrives at a station before its planned arrival time.

Model Assumptions

Assumption 3: We assume that the distributions of arrival times of all feeder trains of a given train are stochastically independent.

Model Assumptions

Assumption 3: We assume that the distributions of arrival times of all feeder trains of a given train are stochastically independent.

Discussion:

- standard assumption used throughout in all previous work
- makes the computation tractable from a practical point of view

Model Assumptions

Assumption 3: We assume that the distributions of arrival times of all feeder trains of a given train are stochastically independent.

Discussion:

- standard assumption used throughout in all previous work
- makes the computation tractable from a practical point of view
- is not so unrealistic as it seems:
  - suppose train A is heavily delayed (primary delay) and
  - train B and train C both catch a secondary delay from train A (that means, their arrival and departure distributions are dependent)
  - but this dependency is reflected in our predictions since the information about A's primary delay is fully used, as soon as it becomes known

Model Assumptions

Assumption 3: We assume that the distributions of arrival times of all feeder trains of a given train are stochastically independent.

Discussion:

- standard assumption used throughout in all previous work
- makes the computation tractable from a practical point of view
- is not so unrealistic as it seems:
  - suppose train A is heavily delayed (primary delay) and
  - train B and train C both catch a secondary delay from train A (that means, their arrival and departure distributions are dependent)
  - but this dependency is reflected in our predictions since the information about A's primary delay is fully used, as soon as it becomes known
- what about track conflicts?
  here is indeed a problem — but the required data is not available

Model Assumptions

Assumption 4: Waiting rules are defined for any pair of arriving and departing trains for which a transfer arc is defined.

Remark:
degree of freedom to define "planned transfer arcs"

for simplicity, we did not implement new transfer possibilities due to other delayed trains

Experiment: Predictions over time

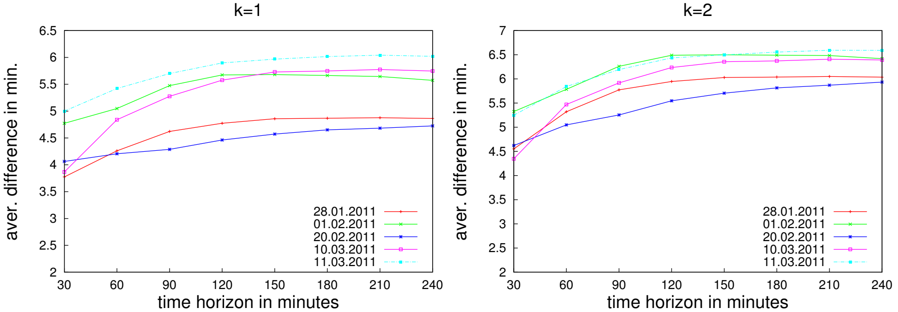
Setup: all delays before 11:59 a.m. incorporated in predictions

Question: How (fast) does the quality of our predictions decrease?

Experiment: Predictions over time

Setup: all delays before 11:59 a.m. incorporated in predictions

Question: How (fast) does the quality of our predictions decrease?



average distance of our predicted expectation values in minutes from the realized ones

Overview

1. Realistic models
2. Multi-criteria search
3. Realtime information
4. Reliable connections (robustness, stochastic forecasts)
5. Extensions
   - **train disposition**
   - multi-modal search

General Goal: Passenger-Oriented Train Disposition

Basic question (train disposition):
Shall a train wait for delayed feeder trains or not?

Why passenger orientation?

- increase of passenger satisfaction and therefore indirectly of the attractiveness of trains as means of transport
- avoidance of reimbursements or repays to customers according to customer charta in case of delays

'...die größte Freud',
ist doch die Zufriedenheit.'

'no joy is found like contentment on earth's round!'

(Wilhelm Busch)

Multi-Criteria Optimization

Possible criteria:

1. Sum of delay (in minutes) at destination, number of cases above 60 minutes, weighted according to customer groups (business travellers or others)
2. number of missed transfers
3. costs for reimbursements to customers (taxi costs, hotel fares, ...)
4. strength of deviation from published schedule (additional expenses for staff and resources; number of track changes, ...)

Status Quo

Current situation in disposition centers for passenger trains (long-distance and regional):

- only local view on effects of decisions (in particular on consequences for passengers)
- (in Germany, due to European regulations)

  only partial view on potential track conflicts
- repeated and time-consuming decision processes

Local Decisions - Global View

Our vision of an optimized disposition

The dispatcher decides

- upon waiting conflicts (shall a train wait?) in his regional disposition center, but
- under consideration of global consequences on the whole train network

Our task:
development of tools for decision support

Passenger Flows

Theoretical ideal: each planned connection for each individual passenger is known (on a daily basis)

Two problems:

- data on this detailed level not available
- data volume fairly large:
  seems infeasible to represent 4 million passengers (and their routes!) individually

Our **model:** group passengers with same destination together, i.e.

for each train we assume to know for each driving section (approximately) how many passengers are heading to which destination

What is needed?

Our goals:

- decision support for train dispatchers
- decisions shall be based on effect on passenger flow
- real-time capability

Algorithm Engineering Challenges:

- modeling issues:
  How to model the objective function? $\rightarrow$ conflicting goals
- dynamic update of passenger flows (kind of multicommodity flow) in real-time

Previous Work on Delay Management

- delay management is hard (Gatto et al.)
- integer linear programming (ILP) models (Schöbel and co-workers)
  - static view (complete delay scenario is known)
  - periodic schedules
  - computational studies on comparatively small subnetworks

We consider online-scenario where the newest delay information is revealed step by step.

Passenger-Oriented Train Disposition

[ESA 2011, joint work with Annabell Berger, Christian Blaar, Andreas Gebhardt and Mathias Schnee)]

Achievements: efficient prototype with building blocks:

1. routines for the permanent update of our graph model subject to incoming delay messages,
2. routines for forecasting future arrival and departure times,
3. the update of passenger flows subject to several rerouting strategies (including dynamic shortest path queries), and
4. the simulation of passenger flows.

Module 1: Timetable Update

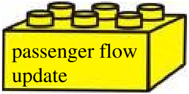
This tool

- updates the timetable with respect to a steady stream of delay messages
- always keeps predictions for future departure and arrival times
- determines critical transfers —
- with respect to standard waiting rules (of German Railways)

Module 2: Passenger Flow Update

This tool

- updates every minute the number of passengers towards their destinations for each train
- passengers who miss their planned connecting train are rerouted.
- A kind of multi-commodity flow problem (one commodity for each destination).



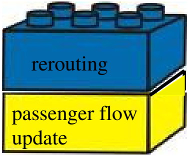
Recall: We do not consider individual passengers, but groups of passengers with the same destination

Rerouting Strategies

We apply the following rerouting strategies if necessary (in this order):

- **Rule 1:** Reroute passenger to the very next train towards his destination.
- **Rule 2:** Apply a dynamic timetable query to calculate a fastest alternative connection. Take the new connection, if acceptable.
- **Rule 3:** If neither rule 1 nor rule 2 apply, we send the passengers to "Nirvana".

  This means: Such passengers have no reasonable alternative to reach their destination!



Module 3: Optimization

Note: there is always a dilemma

- "waiting decision" usually means: many passengers catch a small delay while few "transfer passengers" are happy
- "do not wait decision" usually means: few "transfer passengers" are heavily delayed, all others are not delayed

Our interpretation of passenger-friendly disposition:

- minimize sum of delays at destinations
- minimize deviation from planned passenger flow
- minimize number of non-arriving passengers

→ we take a weighted combination as our objective

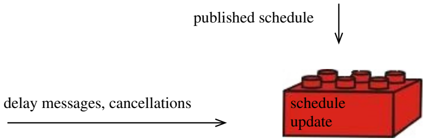
→ weight factors model dependence of the time horizon

Module 3: Optimization

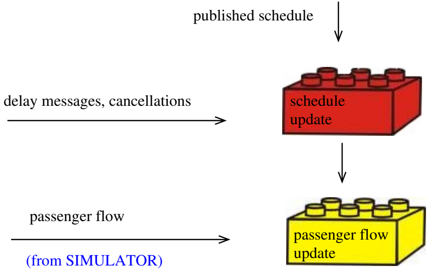
Simple optimization loop:

- order critical transfers with respect to time horizon (most urgent ones come first; consider also number of affected passengers)
- select the most important one
- evaluate objective function for alternative waiting decisions
- propose the most promising ones to dispatcher
- once the current case is settled, proceed to the next

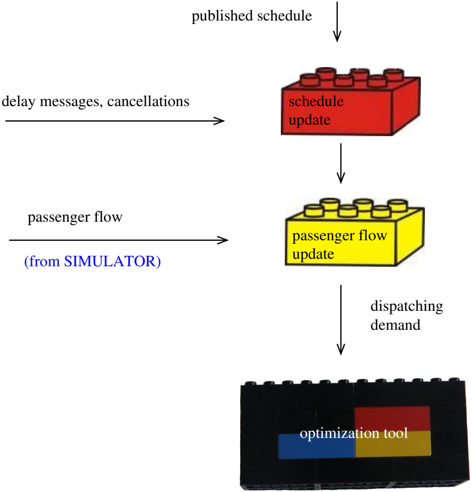
Train Disposition - Overview



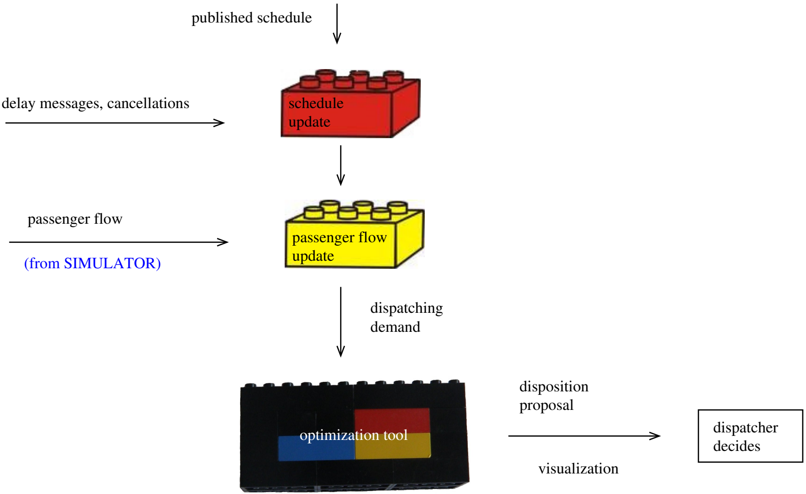
Train Disposition - Overview



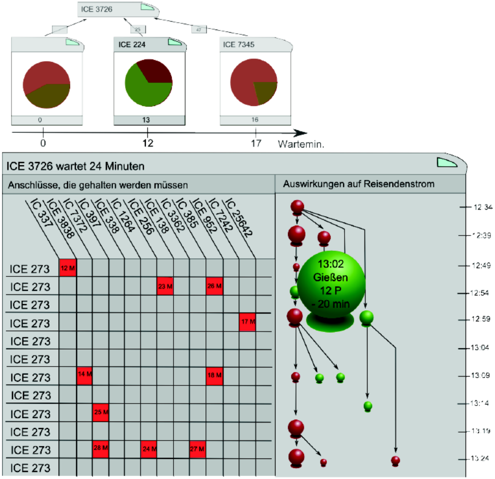
Train Disposition - Overview



Train Disposition - Overview

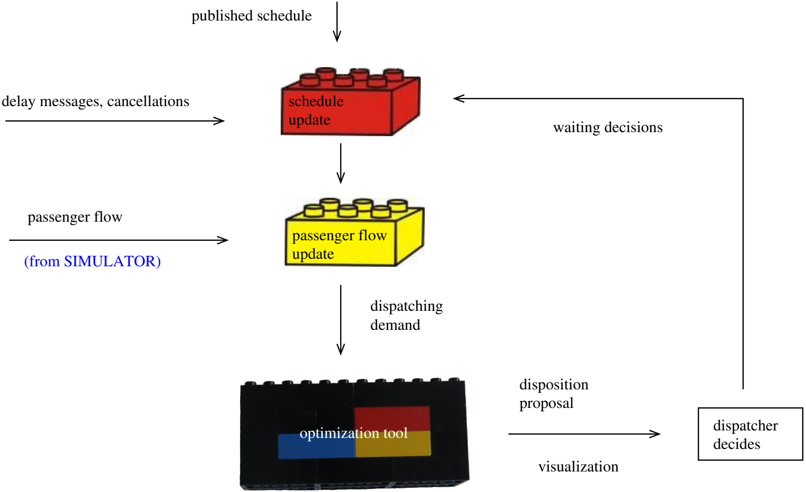


Visualization of Alternatives



Disposition alternatives and their impact on passenger flows

Train Disposition - Overview



Test Instances and Environment

Test instances:

- German train schedule
- 8800 stations, about 40000 trains, one million events per day
- stream of delay messages for whole traffic days

Test environment:

- C++ code, compiled with g++ 4.4.3 and option -O3 under ubuntu linux 10.04.2 LTS
- PC Intel(R) Xeon(R) 2.93 GHz, 4MB cache, 47GB main memory, used only one core

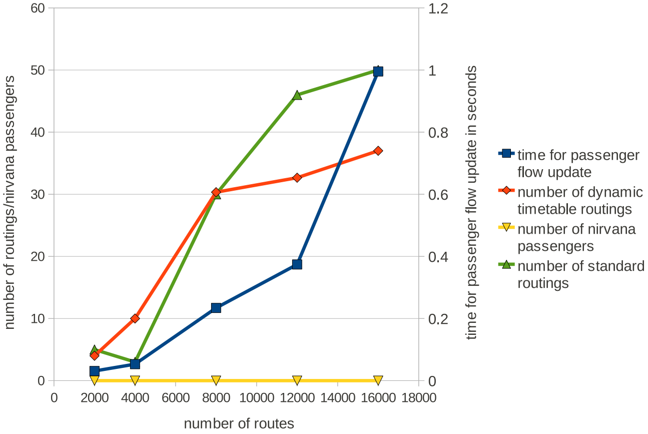
Simulation of Passenger Flows

- Problem: no actual passenger flow data available
- our solution: a SIMULATOR for passenger flows
- in this study: a fairly simple, but extendable prototype

Extensions:

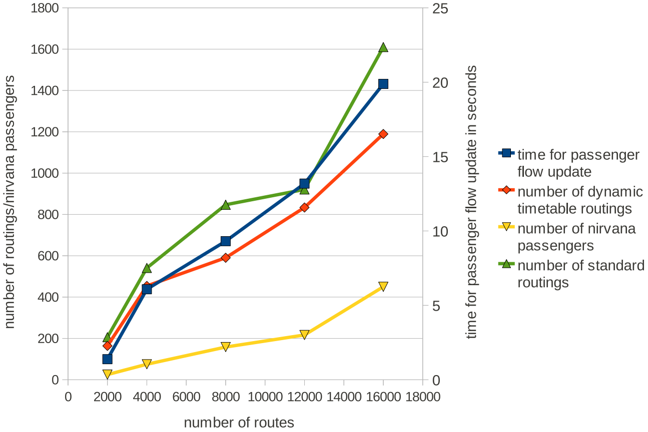
- day-time dependence (like rush hours)
- representative passenger countings
- origin-destination matrices

Flow Updates



Scenario: 20 “light” delays of 5 minutes each

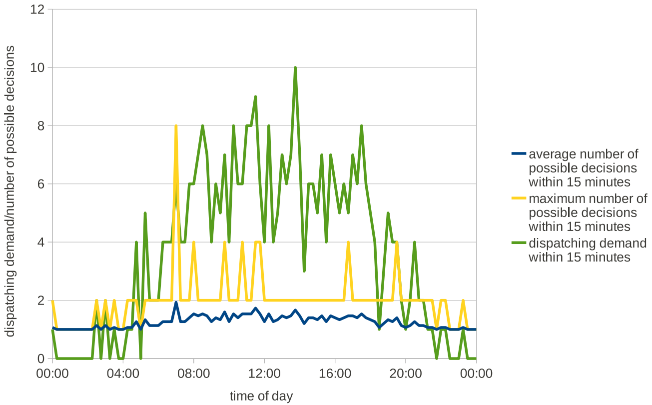
Flow Updates



Scenario: 20 “heavy” delays of 30 minutes each

Passenger-friendly Train Disposition

How many disposition decisions are needed?



Summary

- train disposition is a fairly complex, highly dynamic multi-criteria optimization problem
- real-time update of schedule and passenger flows can be achieved
- simple optimization fast enough to handle typical amount of disposition decisions on ordinary traffic days

Future Work on Train Disposition

- more realistic passenger flows
  (currently we use a simple simulation)
- more advanced optimization
- improved wide-ranging predictions of delay forecasts
- more efficient rerouting of passengers
- capacity restrictions

Final Thoughts

Success story of Algorithm Engineering

Public transport timetable information has achieved quite some progress. But many challenges remain!

Transfer into practice

There is a slow, but steadily improving impact of our research on solutions in industry.

Industry should be more cooperative

Research in our field is often hindered by lack of cooperation from industry side. In particular, availability of test data is a crucial issue.

Future Work — Challenges

Multi-modal timetable information:

- integration of several modes of transport
- exact point-to-point queries
- hybrid models for combinations of "road" and "public transport"

Future Work — Challenges

Pre-trip planning: "recovery robust" timetable information

Goal: compute a travel plan which has an acceptable alternative "backup connection" for every delay scenario

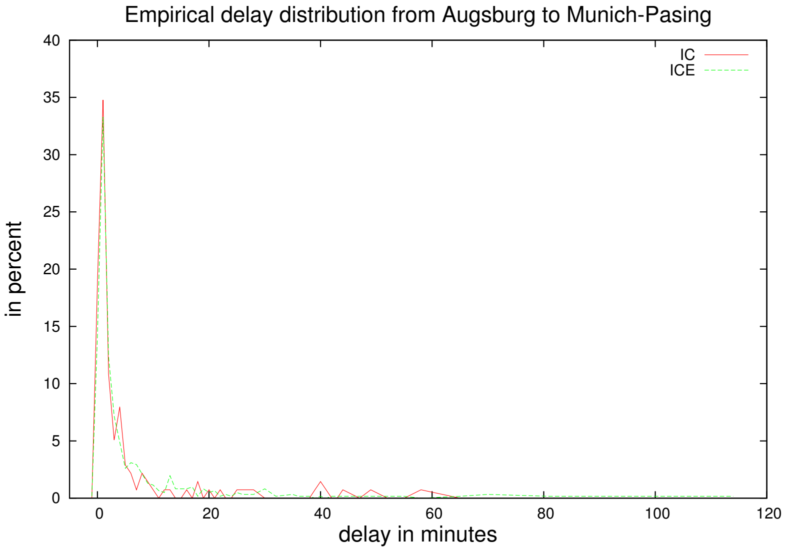
acceptable: with some guarantee on the worst-case arrival time

Future Work — Challenges

Real-time timetable information

- Improve forecasting models for the spreading of delays
- How can we exploit historical delay data?
- How to cope with volatile situations?
  Dilemma: when to inform passengers what to do?

Empirical Delay Distributions



The End?

"When people think, one is done, one has really to start working"

"Wenn die anderen glauben, man ist am Ende, so muss man erst richtig anfangen."

Konrad Adenauer