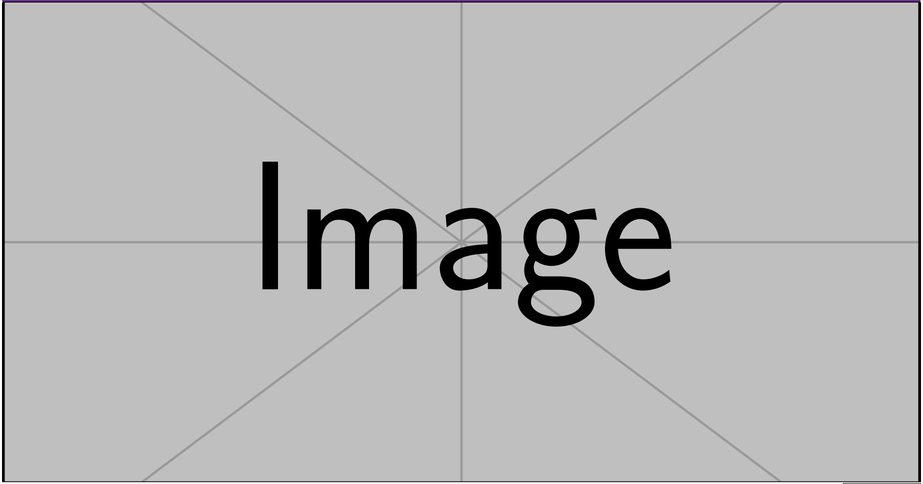
Seminar Algorithmen zum Fahrplanentwurf

TECHNISCHE UNIVERSITÄT DARMSTADT

Graphenmodelle



Fahrpläne als Graphen

- **Das** Beispiel
- Das Problem
- Naive Ansätze

Modell 1: Time-Expanded

- Umstiegszeiten
- Verfeinerungen

Modell 2: Time-Dependent

Vergleiche

Exkurs: Implementation in MOTIS

16×9

(Original size: 320×180 bp)

- Quelle: `europe.motis-project.de` [Super bei Verspätungen ;)]

16×9

Wie muss ich meinen Graph modellieren, damit ich das Earliest Arrival Problem mit einem Shortest Path Algorithmus lösen kann?

1. Griff in die Terminologie-Kiste

Züge und Stationen

Züge und Zugklassen

1. Griff in die Terminologie-Kiste

Züge und Stationen

Züge und Zugklassen

1. Griff in die Terminologie-Kiste
Züge und Stationen

Züge und Zugklassen

1. Griff in die Terminologie-Kiste
Züge und Stationen

Züge und Zugklassen

16×9

1. Griff in die Terminologie-Kiste

Züge und Stationen

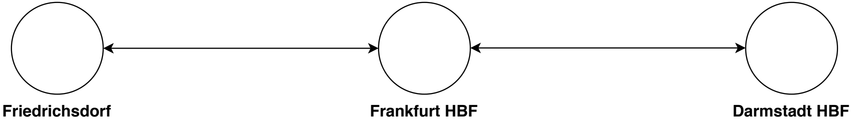
Stationen

1. Griff in die Terminologie-Kiste
Züge und Stationen

Stationen

Wie modelliere ich einen Fahrplan als Graphen?

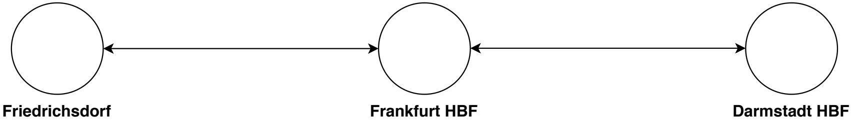
- Simple Idee: Nodes sind Stationen, Edges sind Verbindungen



16×9

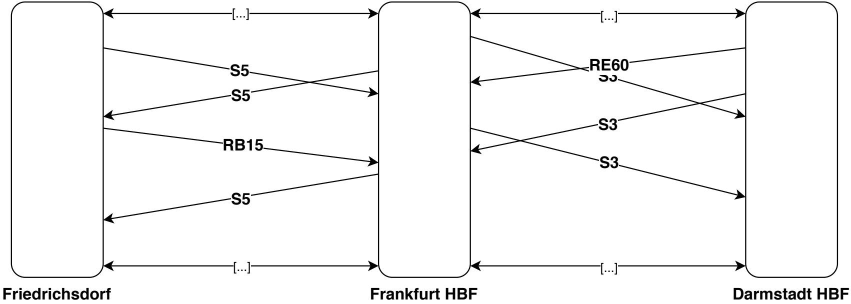
Wie modelliere ich einen Fahrplan als Graphen?

- Simple Idee: Nodes sind Stationen, Edges sind Verbindungen



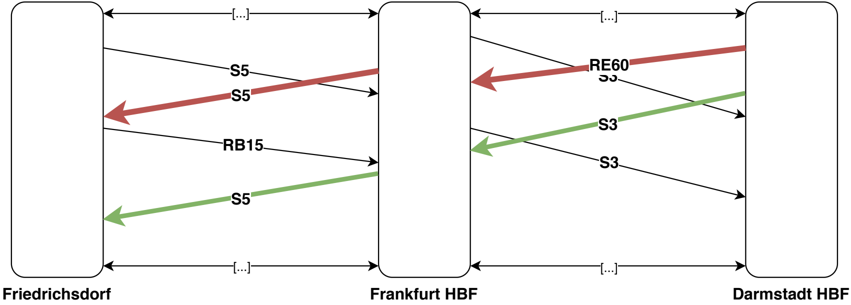
Was ist das Problem hier?

- Wir müssen irgendwie Zeitverhältnisse innerhalb des Graphen darstellen!



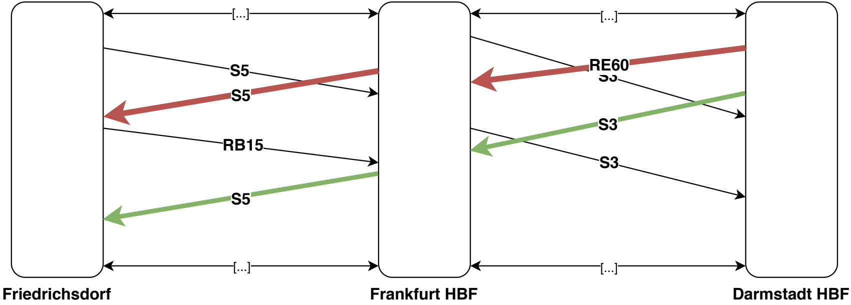
16×9

- Wir müssen irgendwie Zeitverhältnisse innerhalb des Graphen darstellen!



16×9

- Wir müssen irgendwie Zeitverhältnisse innerhalb des Graphen darstellen!



Was ist das Problem hier?

16×9

Probleme

Die es zu lösen gilt...

- Zeitverhältnisse im Graphen

Probleme

Die es zu lösen gilt...

- Zeitverhältnisse im Graphen
- **Takt**fahrpläne

16×9
(Original size: 320×180 bp)

Probleme

Die es zu lösen gilt...

- Zeitverhältnisse im Graphen
- **Takt**fahrpläne
- Umstiege bzw. Umsteigezeiten

Probleme

Die es zu lösen gilt...

- Zeitverhältnisse im Graphen
- **Takt**fahrpläne
- Umstiege bzw. Umsteigezeiten
- Fußwege zwischen / innerhalb von Stationen

Probleme

Die es zu lösen gilt...

- Zeitverhältnisse im Graphen
- **Takt**fahrpläne
- Umstiege bzw. Umsteigezeiten
- Fußwege zwischen / innerhalb von Stationen
- Intermodalität

Zwei gängige Modelle

Time-Expanded vs. Time-Dependent

16×9

Größenordnungen

- Ein paar Zahlen im Jahr 2008 (**nur** Züge):

Anzahl an	
Zügen	68073
Fußwegen	425

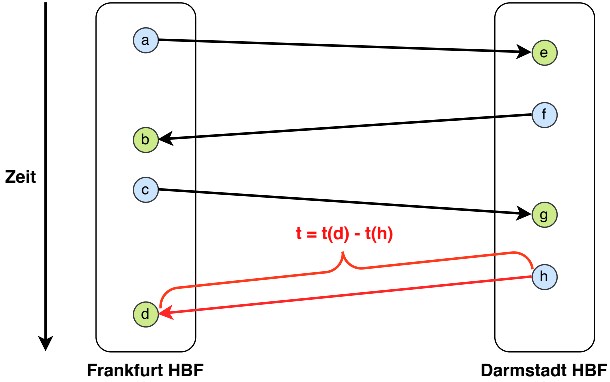
16×9
(Original size: 320×180 bp)

16×9

(Original size: 320×180 bp)

Das Time-Expanded Modell
Train-Edges

- Modelliere jedes "**Event**" innerhalb einer Station als eigenen Knoten.

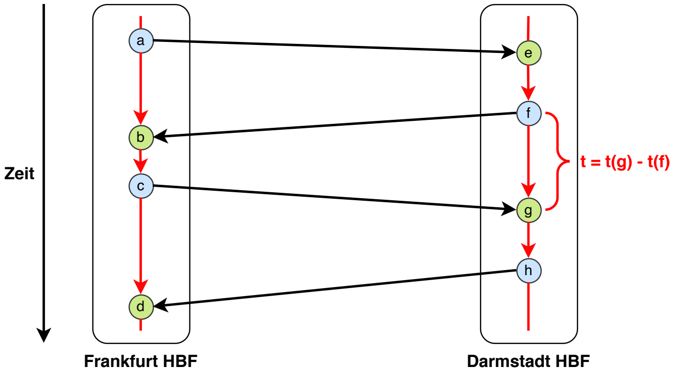


16×9

Das Time-Expanded Modell

Waiting-Edge

- Erstelle eine Stationsinterne Kante, die **Exchange-Edge**.



Das Time-Expanded Modell

Was bringt uns das Ganze jetzt?

Kanten innerhalb einer Station $\approx$ "Umstiege"

16×9

16:30 – 17:17 | [redacted] | 1 Umstieg

RE 60	RE 16

Darmstadt Hbf — Friedrichsdorf(Taunus)

Es liegen Meldungen vor.

Details

16:30 **Darmstadt Hbf** — Gl. 7

21min

RE 60

nach Frankfurt(Main)Hbf

Beförderer | Fahrradmitnahme | Weitere Informationen

Zuginformationen

1 Haltestelle

16:51 **Frankfurt(Main)Hbf** — Gl. 10

0min Umstieg

16:51 **Frankfurt(Main)Hbf** — Gl. 14

26min

RE 16

nach Friedberg(Hess)

Beförderer | Fahrradmitnahme | Weitere Informationen

Zuginformationen

2 Haltestellen

17:17 **Friedrichsdorf(Taunus)** — Gl. 5

16:30 – 17:17 | :) | 1 Umstieg

RE 60	RE 16

Darmstadt Hbf — Friedrichsdorf(Taunus)

Es liegen Meldungen vor.

Details

16:30 **Darmstadt Hbf** › Gl. 7

21min

RE 60

nach Frankfurt(Main)Hbf

Beförderer | Fahrradmitnahme | Weitere Informationen

Zuginformationen

1 Haltestelle

16:51 **Frankfurt(Main)Hbf** › Gl. 10

0min Umstieg

16:51 **Frankfurt(Main)Hbf** › Gl. 14

26min

RE 16

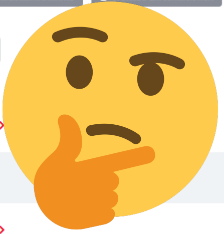
nach Friedberg(Hess)

Beförderer | Fahrradmitnahme | Weitere Informationen

Zuginformationen

2 Haltestellen

17:17 **Friedrichsdorf(Taunus)** › Gl. 5



Problem: Erhebliche Umsteigezeiten innerhalb von großen Stationen.

2. Griff in die Terminologie-Kiste

Realistische Umstiegsregeln

- Definiere Konstante und Variable Umstiegsregeln:
    1. Standard-Umstiegszeit für alle Züge

2. Griff in die Terminologie-Kiste

Realistische Umstiegsregeln

- Definiere Konstante und Variable Umstiegsregeln:
  1. Standard-Umstiegszeit für alle Züge
  2. Regeln basierend auf Transferklassen & Zuglinien

2. Griff in die Terminologie-Kiste

Realistische Umstiegsregeln

- Definiere Konstante und Variable Umstiegsregeln:
  1. Standard-Umstiegszeit für alle Züge
  2. Regeln basierend auf Transferklassen & Zuglinien
  3. Regeln zwischen einzelnen Zügen

16×9

Konstante Umstiegszeiten

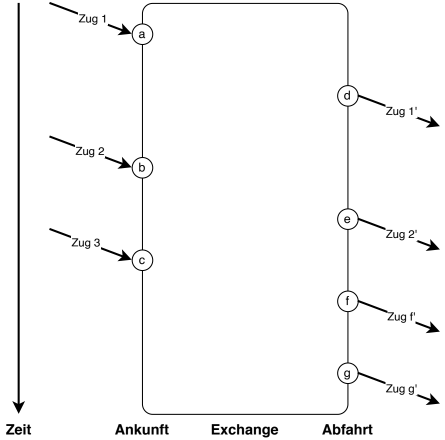
In order to describe the model formally, let S, S' be stations, z be a train, d_S^z the departure node at station S belonging to train z, and a_S^z the arrival node at station S for train z. Then $G = (V, E)$, with $V = D \cup A \cup C$ and $E = Z \cup W \cup L \cup E \cup Y$, where

- D is the set of departure nodes,
- A is the set of arrival nodes,
- C is the set of change nodes with $|C| = |D \cup A|$,
- $Z = \bigcup_{SS'} Z_{SS'}$ is the set of train edges, where for each pair of stations S, S': $Z_{SS'} = \{(d_S^z, a_{S'}^z) : d_S^z \in D, a_{S'}^z \in A, z \text{ is a train}\}$,
- $W = \bigcup_S W_S$ is the set of waiting edges connecting a change node to the next change node at the same station, where $W_S = \{(c_S, c'_S) : c_S, c'_S \in C\}$,
- $L = \{(a_S^z, c_S) : a_S^z \in A, c_S \in C\}$ is the set of edges for leaving a train,
- $E = \{(c_S, d_S^z) : c_S \in C, d_S^z \in D\}$ is the set of edges for entering a train, and
- $Y = \{(a_S^z, d_S^z) : a_S^z \in, d_S^z \in D\}$ is the set of edges for staying in a train, connecting the arrival of a train at a station to the departure of the same train at that station.

16×9

Konstante Umstiegszeiten

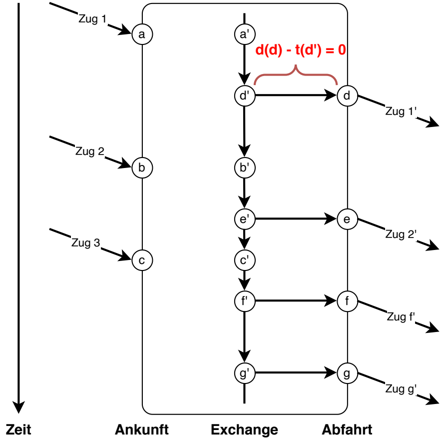
- Spalte Exchange-Edge in Ankunfts- und Abfahrtsknoten



16×9

Konstante Umstiegszeiten

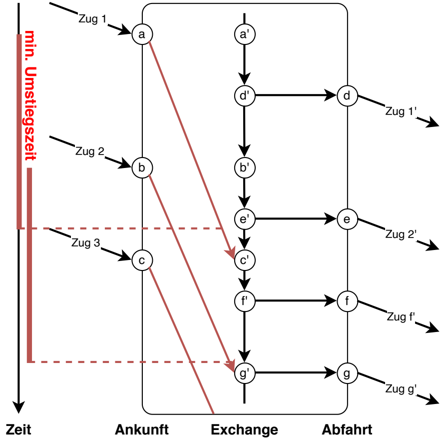
- Erstelle neue Exchange Edge



16×9

Konstante Umstiegszeiten

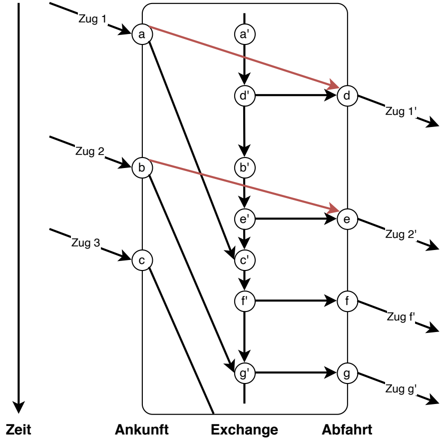
- Kanten von Ankunft $\rightarrow$ Exchange-Edge basierend auf min. Umstiegszeit



16×9

Variable Umstiegszeiten

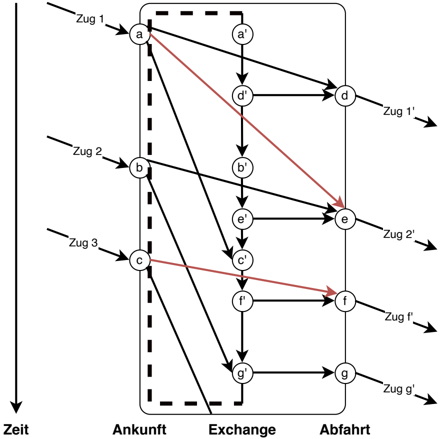
- Erstelle Kanten von Ankunft $\rightarrow$ Abfahrt des selben Zuges



16×9

Variable Umstiegszeiten

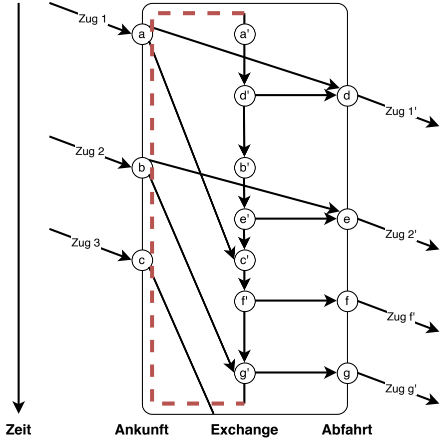
- Weitere variable Umstiegszeiten als zzgl. Kanten



16×9

Taktfahrplanmodellierung

- Taktfahrplanmodellierung via Exchange-Edge-Schleife



16×9

Verfeinerung 1: Verkehrstage

Problem: Der Takt unseres Fahrplans ist nicht nur ein Tag!

16×9

Verfeinerung 1: Verkehrstage

Problem: Der Takt unseres Fahrplans ist nicht nur ein Tag!

- N Tage ($1440 min/d$) im Taktfahrplan

Verfeinerung 1: Verkehrstage

Problem: Der Takt unseres Fahrplans ist nicht nur ein Tag!

- N Tage ($1440 min/d$) im Taktfahrplan
- Versehe Kanten mit einem Tupel aus Verkehstagen und Zeit: $\{[d], t\}$
- Beim SP-Durchlauf: Prüfe auf validen Tag, und bestimme Zeit für den Tag

Verfeinerung 1: Verkehrstage

Problem: Der Takt unseres Fahrplans ist nicht nur ein Tag!

- N Tage ($1440 min/d$) im Taktfahrplan
- Versehe Kanten mit einem Tupel aus Verkehstagen und Zeit: $\{[d], t\}$
- Beim SP-Durchlauf: Prüfe auf validen Tag, und bestimme Zeit für den Tag

$$Tag = floor(\frac{t_{Absolut}}{1440})$$

16×9

Verfeinerung 1: Verkehrstage

Problem: Der Takt unseres Fahrplans ist nicht nur ein Tag!

- N Tage ($1440 min/d$) im Taktfahrplan
- Versehe Kanten mit einem Tupel aus Verkehstagen und Zeit: $\{[d], t\}$
- Beim SP-Durchlauf: Prüfe auf validen Tag, und bestimme Zeit für den Tag

$$Tag = floor(\frac{t_{Absolut}}{1440})$$

$$length(E) = t_{Ankunft} - t_{Abfahrt} \mod 1440$$

16×9

(Original size: 320×180 bp)

Verfeinerung 1: Verkehrstage

Problem: Wie behandeln wir besuchte Knoten beim SP-Durchlauf, die invalide sind?

[1] Pyrga et. al.: Efficient Models for Timetable Information in Public Transportation Systems

Verfeinerung 1: Verkehrstage

Problem: Wie behandeln wir besuchte Knoten beim SP-Durchlauf, die invalide sind?

- Als Beispiel am Dijkstra:
  - $dist \to \infty$
  - Füge Knoten wieder am Ende des Sets ein ("Nächster" Tag).

Weitere Verfeinerungen sind möglich[1]...

[1] Pyrga et. al.: Efficient Models for Timetable Information in Public Transportation Systems

16×9

Verfeinerung 2: Fußwege

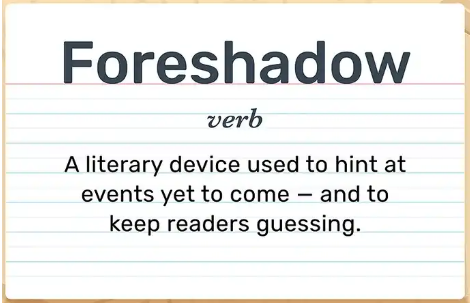
Die Rückkehr der Terminologie-Kiste

Was macht Fußwege so besonders?

Verfeinerung 2: Fußwege
Die Rückkehr der Terminologie-Kiste

Was macht Fußwege so besonders?

- Fußwege können zu jeder Zeit genutzt werden
- Man könnte sagen sie sind... *Zeitabhängig* (Time-Dependent)...



16×9

Verfeinerung 2: Fußwege

- Alle Fußwege im Graphen speichern ist nicht sinnvoll.

Verfeinerung 2: Fußwege

- Alle Fußwege im Graphen speichern ist nicht sinnvoll.
- Stattdessen: Speichere Liste an Fußwegen für jede Station mit konst. Zeit

Verfeinerung 2: Fußwege

- Nutze den Floyd-Warshall-Algorithmus

16×9

Verfeinerung 2: Fußwege

- Nutze den Floyd-Warshall-Algorithmus
- SP von allen Knoten zu allen anderen Knoten, negative Kantenwerte sind erlaubt

Verfeinerung 2: Fußwege

- Nutze den Floyd-Warshall-Algorithmus
- SP von allen Knoten zu allen anderen Knoten, negative Kantenwerte sind erlaubt
- Speichere Wege in jeder Station, maskiere Wege in Suchanfragen als Kanten zwischen Stationen
- Beachte Fußwege als Start einer Reise!

Größenordnungen

- Ein paar Zahlen im Jahr 2008 (**nur** Züge):

Anzahl an Knoten	
Ankunft	801.8 Tsd.
Abfahrt	801.8 Tsd.
Exchange	556.6 Tsd.
Gesamt	2160.2 Tsd.

Größenordnungen

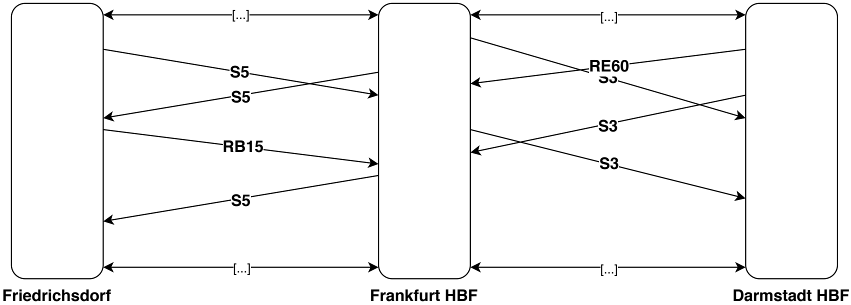
- Ein paar Zahlen im Jahr 2008 (**nur** Züge):

Anzahl an Kanten	
Zug	801.8 Tsd.
Weiterfahrt	733.7 Tsd.
Ankunft	796.7 Tsd.
Abfahrt	796.7 Tsd.
Warten (auf Exchange-Edge)	556.6 Tsd.
Besondere Umstiege	20.2 Tsd.
Gesamtanzahl	3704.7 Tsd.

16×9

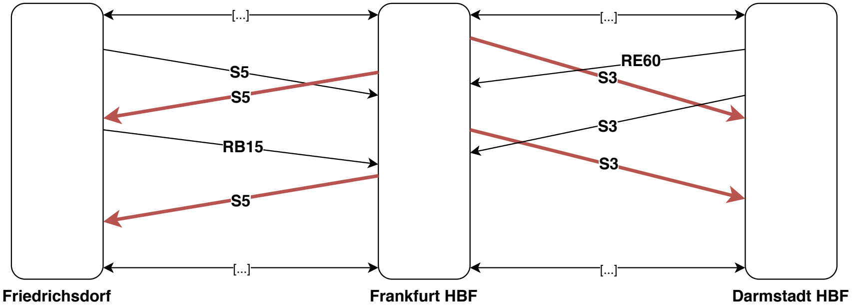
Das Time-Dependent Modell

- Denken wir zurück an das erste Beispiel...



Das Time-Dependent Modell

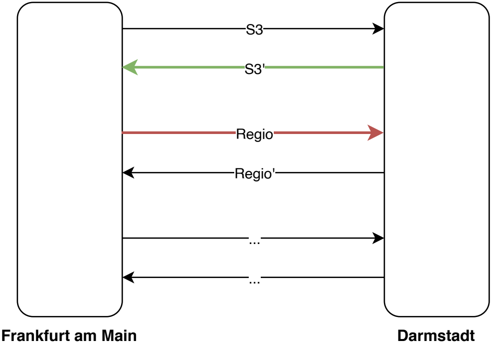
- Denken wir zurück an das erste Beispiel...



16×9

Das Time-Dependent Modell

- Jetzt bauen wir das ganze etwas um...

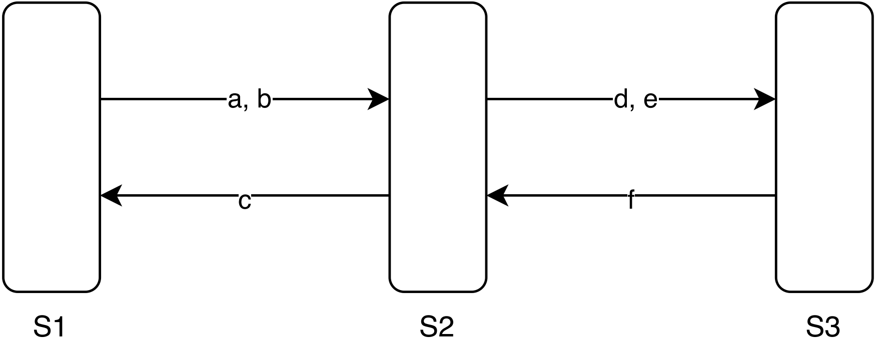


S3'
...
Abfahrt DA: 15:30
Ankunft FFM: 16:08
Abfahrt DA: 16:00
Ankunft FFM: 16:38
...

Regio
...
Abfahrt FFM: 15:20
Ankunft DA: 15:40
Abfahrt FFM: 15:50
Ankunft DA: 16:10
...

Das Time-Dependent Modell

- Oder etwas formeller:



Das Time-Dependent Modell

- Wir definieren für jede Kante (u, v) eine Funktion $f(t) : \mathcal{T} \to \mathcal{T}$.
  - $t \in \mathcal{T}$, $\mathcal{T} \triangleq$ Zeit.

Das Time-Dependent Modell

- Wir definieren für jede Kante (u, v) eine Funktion $f(t) : \mathcal{T} \rightarrow \mathcal{T}$.
  - $t \in \mathcal{T}$, $\mathcal{T} \triangleq$ Zeit.

- Dann ist unser Kantengewicht (*die Zeit die der Zug fährt*):

$$travel\_time(t) = f_{(u,v)}(t) - t$$

16×9

Das Time-Dependent Modell

- Wir definieren für jede Kante (u, v) eine Funktion $f(t) : \mathcal{T} \rightarrow \mathcal{T}$.
  - $t \in \mathcal{T}$, $\mathcal{T} \triangleq$ Zeit.

- Dann ist unser Kantengewicht (*die Zeit die der Zug fährt*):

$$travel\_time(t) = f_{(u,v)}(t) - t$$

Wie sieht so eine Funktion in der Realität aus?

16×9

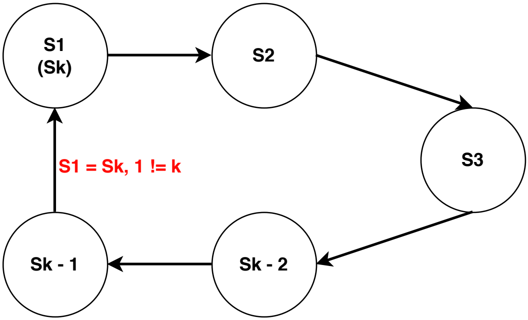
Das Time-Dependent Modell
Konstante Umstiegszeiten

- Für konstante Umstiegszeiten definieren wir Zugrouten:
  - Sei R eine Zugroute $S_0, S_1, ..., S_{k-1}, S_k$ für $k > 0$

16×9

Das Time-Dependent Modell
Konstante Umstiegszeiten

- Für konstante Umstiegszeiten definieren wir Zugrouten:
  - Sei R eine Zugroute $S_0, S_1, ..., S_{k-1}, S_k$ für $k > 0$
  - Erlaubt: $S_i = S_j$, $i, j \in k$, für $i \neq j$ (Schleifen)!

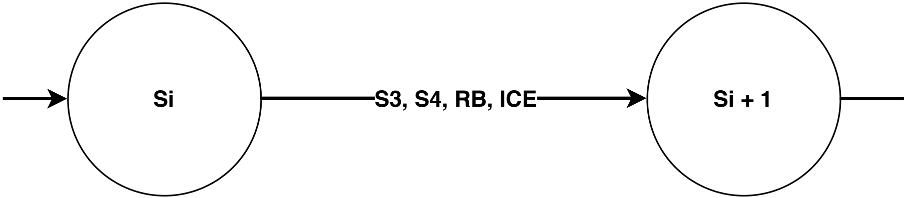


16×9

Das Time-Dependent Modell

Konstante Umstiegszeiten

- Jetze gruppieren wir alle Züge, die die gleiche Strecke fahren, in eine Zugroute
- **Was ist hier das Problem?**

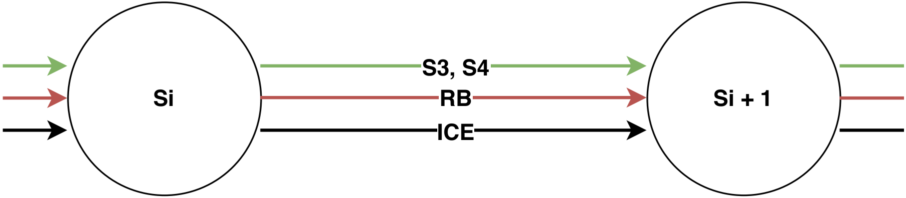


16×9

Das Time-Dependent Modell

Konstante Umstiegszeiten

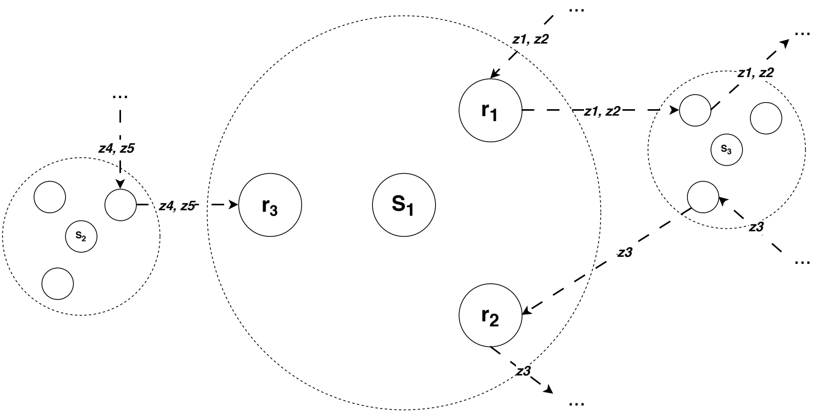
- Keine Züge z_1, z_2 dürfen S_i um t_1, t_2 ($t_1 \leq t_2$) verlassen, und z_2 vor z_1 S_{i+1} erreichen!
- In diesem Fall spalten wir in einzelne Routen:



16×9

Das Time-Dependent Modell
Konstante Umstiegszeiten

- Wir teilen eine Station in einen Stationsknoten S und Routenknoten r_i

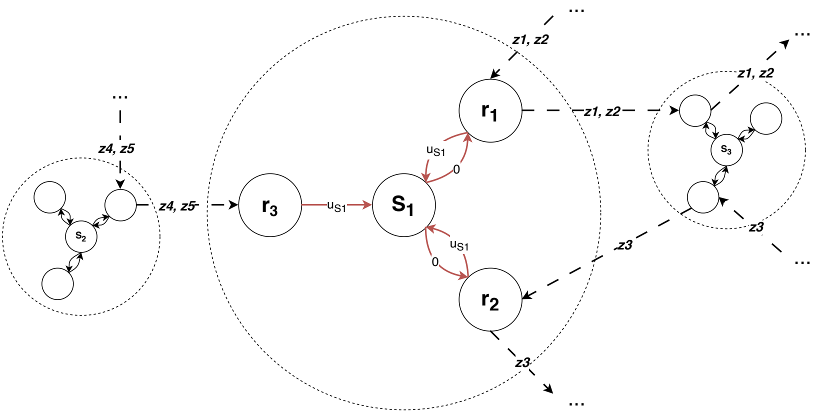


16×9

Das Time-Dependent Modell

Konstante Umstiegszeiten

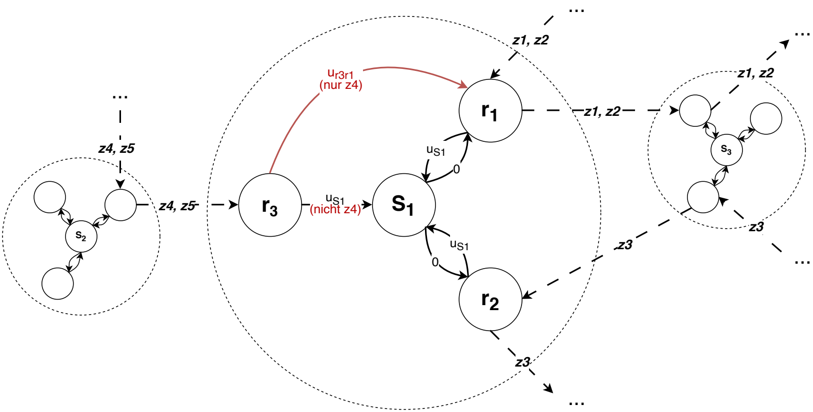
- Wir erstellen Transferkanten mit *ausgehenden* Transferkosten



16×9

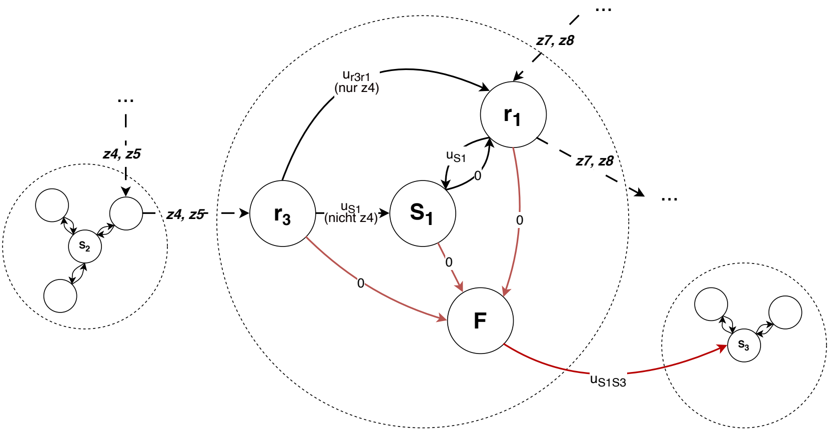
Das Time-Dependent Modell
Variable Umstiegszeiten

- Wir erstellen spezielle Kanten und Regeln zwischen Knoten



Fußwege

- Fußwege sind immer abhängig von der Zeit, also einfach zu modellieren



Größenordnungen

- Ein paar Zahlen im Jahr 2007 (**nur** Züge):
  - 56,994 Züge, 8916 Stationen

	Time Dependent	Time Expanded
Knoten	240 Tsd.	3479 Tsd.
Kanten	670 Tsd.	5633 Tsd.

Fahrpläne als Graphen

Das Beispiel

Das Problem

Naive Ansätze

Modell 1: Time-Expanded

Umstiegszeiten

Verfeinerungen

Modell 2: Time-Dependent

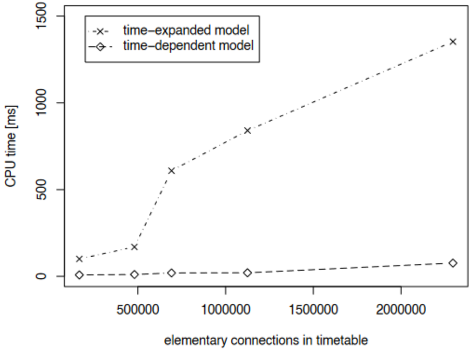
Vergleiche

Exkurs: Implementation in MOTIS

16×9

Performanz

- TD ist besser für Single-Criteria Suche[2].



[2] Pyrga et al.: Experimental Comparison of Shortest Path Approaches for Timetable Information, 2004

Performanz

- Aber: Komplexität von TD wächst schnell mit mehr Kriteria und Regeln
- Also in realistischen Szenarien!
  - TD nur 58% schneller (in CPU-Zeit) als TE

16×9

Implementation in MOTIS
Graphenmodell

- Time-Expanded mit Verfeinerungen
  - Konstante und Variable Umstiegszeiten
  - Verkehrstage
  - Fußwege

Implementation in MOTIS
Kantengewichtungen

- Reisezeit ($t_{Ankunft} - t_{Abfahrt} \mod 1440$)
- Anzahl der Umstiege (Ankunfts- und Spezialkanten)
- Ticketpreise

Implementation in MOTIS
Algorithmus

- Viele Algorithmusverfeinerungen und Spezialattribute
  - Thema des nächsten Vortrags!

Hinweis zur Ausrichtung (insbesondere columns)

Die Standardausrichtung wurde gegenüber den Beamer-Voreinstellungen von `c` zu `t` geändert. Dies bedeutet, dass Inhalt auf der Folie oben ausgerichtet wird. Dies entspricht den Vorgaben, hat allerdings den Nachteil, dass die `columns`-Umgebung in diesem Fall bei der Positionierung von Bildern ungewohnte Ergebnisse erzeugt.
Die Ausrichtung kann in diesem Fall entweder global mit der Option `c` wieder zum Standard geändert werden, oder aber das `c` wird direkt an die `columns`-Umgebung übergeben. Zum Beispiel:

- eins
- zwei