

Aufgabe 5.3

T. Adam, M. ben Ahmed

Universität Osnabrück

Æ

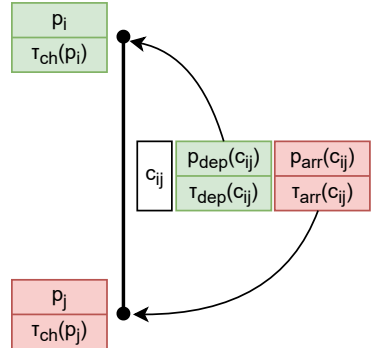
January 25, 2021

Intriguingly Simple and Fast Transit Routing (2013)

- J. Dibbelt, T. Pajor, B. Strasser, D. Wagner
- Berechnen von optimalen Reisen in öffentlichen Verkehrsnetzen
- Connection Scan Algorithmus (CSA)
 - earliest arrival
 - profile queries
 - multi-criteria optimization
 - minimum expected arrival time

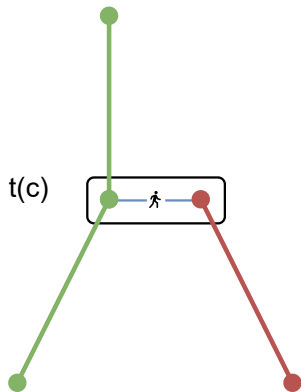
Stops und Connections

- Stops p
- Connections c
- $p_{dep}(c)$ Abfahrt-Stop
- $p_{arr}(c)$ Ankunft-Stop
- $\tau_{dep}(c)$ Abfahrtszeit von c
- $\tau_{arr}(c)$ Ankunftszeit von c
- $\tau_{ch}(p)$ Umstiegszeit an Stop p



Trips und Footpaths

- Trip als Reihenfolge von Connections des selben Fahrzeugs
- $t(c)$ Trip von Connection c
- Footpath zwischen zwei nahegelegenen Stops
- Footpaths dürfen nicht aufeinander folgen
- Routen als Mengen von Trips mit der selben Stop-Sequenz



Earliest Arrival Problem

- Gegeben:
 - Startort und Startzeit
 - Zielort
 - Fahrplan mit Abfahrts- und Ankunftszeiten
- Gesucht: Menge an Verbindungen (Route) mit frühester Ankunftszeit

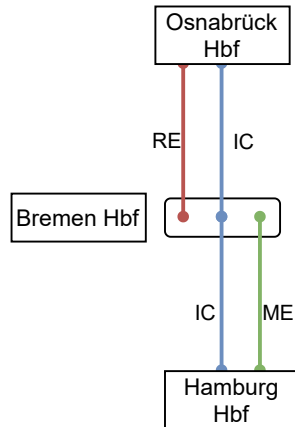


[deutschebahn.com]

Connection Scan Algorithmus

Connection Scan Algorithmus

- Array aller Connections nach Abfahrtszeit sortieren
- Für jeden Stop ein Label speichern (initial ∞)
- Über Connection Array iterieren und entsprechende Labels aktualisieren
- Beispiel mit 5 Stops und 4 Connections



Connection Scan Algorithmus - Beispiel

labels

Osnabrück Hbf	Bremen Hbf Gleis 5	Bremen Hbf Gleis 7	Bremen Hbf Gleis 9	Hamburg Hbf
14.00	∞	∞	∞	∞

$\tau_{ch}(\cdot) = 0.05$

connections

RE 14.30	RE 15.50	IC 15.00	IC 16.00	ME 16.00	ME 17.30	IC 16.05	IC 17.00
Osnabrück Hbf	Bremen Hbf Gleis 5	Osnabrück Hbf	Bremen Hbf Gleis 7	Bremen Hbf Gleis 9	Hamburg Hbf	Bremen Hbf Gleis 7	Hamburg Hbf

Connection Scan Algorithmus - Beispiel

labels

Osnabrück Hbf	Bremen Hbf Gleis 5	Bremen Hbf Gleis 7	Bremen Hbf Gleis 9	Hamburg Hbf
14.00	15.50	15.55	15.55	∞

$\tau_{ch}(\cdot) = 0.05$

connections

RE 14.30	RE 15.50	IC 15.00	IC 16.00	ME 16.00	ME 17.30	IC 16.05	IC 17.00
Osnabrück Hbf	Bremen Hbf Gleis 5	Osnabrück Hbf	Bremen Hbf Gleis 7	Bremen Hbf Gleis 9	Hamburg Hbf	Bremen Hbf Gleis 7	Hamburg Hbf

Connection Scan Algorithmus - Beispiel

labels

Osnabrück Hbf	Bremen Hbf Gleis 5	Bremen Hbf Gleis 7	Bremen Hbf Gleis 9	Hamburg Hbf
14.00	15.50	15.55	15.55	17.30

$\tau_{ch}(\cdot) = 0.05$

connections

RE 14.30	RE 15.50	IC 15.00	IC 16.00	ME 16.00	ME 17.30	IC 16.05	IC 17.00
Osnabrück Hbf	Bremen Hbf Gleis 5	Osnabrück Hbf	Bremen Hbf Gleis 7	Bremen Hbf Gleis 9	Hamburg Hbf	Bremen Hbf Gleis 7	Hamburg Hbf

Connection Scan Algorithmus - Beispiel

labels

Osnabrück Hbf	Bremen Hbf Gleis 5	Bremen Hbf Gleis 7	Bremen Hbf Gleis 9	Hamburg Hbf
14.00	15.50	15.55	15.55	17.00









$\tau_{ch}(\cdot) = 0.05$

connections

RE 14.30	RE 15.50	IC 15.00	IC 16.00	ME 16.00	ME 17.30	IC 16.05	IC 17.00
Osnabrück Hbf	Bremen Hbf Gleis 5	Osnabrück Hbf	Bremen Hbf Gleis 7	Bremen Hbf Gleis 9	Hamburg Hbf	Bremen Hbf Gleis 7	Hamburg Hbf

Profile Queries

- Abfahrts- und Ankunftszeiten häufig unbekannt
- Mehrere Reisen kommen in Frage
- Input: Fahrplan, Zielstop t
- Output: st-Profil für jeden Stop s
- Profile sind Mengen von (Ankunftszeit, Abfahrzeit) Paaren
- ggf. pareto-optimale Mengen verwalten

Osnabrück Hbf Hamburg Hbf	14:23 16:14	1:51	0	ICE	
 Details einblenden		Rückfahrt hinzufügen			
Osnabrück Hbf Hamburg Hbf	14:29 17:43	3:14	1	RE, ME	–
 Details einblenden		Rückfahrt hinzufügen			
Osnabrück Hbf Hamburg Hbf	15:23 17:14	1:51	0	IC	
 Details einblenden		Rückfahrt hinzufügen			
Osnabrück Hbf Hamburg Hbf	15:29 18:26	2:57	1	RE, ME	–
 Details einblenden		Rückfahrt hinzufügen			
Osnabrück Hbf Hamburg Hbf	16:23 18:14	1:51	0	ICE	
 Details einblenden		Rückfahrt hinzufügen			

[bahn.de]

Profile Connection Scan

- Sortiere Connections absteigend nach Abfahrtszeit
- Iteriere über alle Connections
- Füge zu $p_{dep}(c)$ Profil hinzu
- Entferne ggf. dominierte Paare

Osnabrück Hbf Hamburg Hbf	14:23 16:14	1:51	0	ICE	
Details einblenden		Rückfahrt hinzufügen			
Osnabrück Hbf Hamburg Hbf	14:29 17:43	3:14	1	RE, ME	–
Details einblenden		Rückfahrt hinzufügen			
Osnabrück Hbf Hamburg Hbf	15:23 17:14	1:51	0	IC	
Details einblenden		Rückfahrt hinzufügen			
Osnabrück Hbf Hamburg Hbf	15:29 18:26	2:57	1	RE, ME	–
Details einblenden		Rückfahrt hinzufügen			
Osnabrück Hbf Hamburg Hbf	16:23 18:14	1:51	0	ICE	
Details einblenden		Rückfahrt hinzufügen			

[bahn.de]

Multi Criteria

- Profile sind Triple aus (Ankunftszeit, Abfahrzeit, #Umstiege)
- Iteriere über alle Connections
- Füge zu $p_{dep}(c)$ Profil hinzu
- Entferne ggf. dominierte Paare

Minimum Expected Arrival Time

- Berechnet alternative Reisen

Osnabrück Hbf Potsdam Hbf	14:08 ⚠ 17:45	3:37 2	IC, ICE, RB	Ä 11
Details einblenden		> Rückfahrt hinzufügen		
Osnabrück Hbf Potsdam Hbf	14:08 ⚠ 17:58	3:50 1	IC	Ä 11
Details einblenden		> Rückfahrt hinzufügen		
Osnabrück Hbf Potsdam Hbf (S)	14:23 19:02	4:39 3	ICE, S	Ä 11
Details einblenden		> Rückfahrt hinzufügen		
Osnabrück Hbf Potsdam Hbf	14:23 19:06	4:43 2	ICE, RE	Ä 11
Details einblenden		> Rückfahrt hinzufügen		
Osnabrück Hbf Potsdam Hbf (S)	16:08 ⚠ 19:52	3:44 3	IC, ICE, S	Ä 11
Details einblenden		> Rückfahrt hinzufügen		

[bahn.de]

Algorithmen Übersicht

- Connection Scan
- Dijkstra
 - Time-Expanded: Graph mit Knoten = Zeitpunkte und Kanten = Connections
 - Time-Dependent: Graph mit Knoten = Stop und Kanten = elementar Verbindung
- Dynamische Programmierung auf Fahrplan
 - RAPTOR

London

- 20.843 Stops
- 25.537 Trips
- 4.850.431 Connections
- 2.135 Routen
- 45.652 Footpaths

Pareto Profile All-to-One

RAPTOR	1179 ms
Connection Scan	255 ms

Earliest Arrival One-to-All

Time-Expanded	876,2 ms
Time-Dependent	18,9 ms
Connection Scan	9,7 ms

Earliest Arrival One-to-One

Time-Expanded	64,4 ms
Time-Dependent	10,9 ms
Connection Scan	2,0 ms

Zusammenfassung

- CSA hat ein relativ einfaches Konzept
- CSA ist schneller als bestehende Ansätze
- Insbesondere MEAT wird besonders schnell gelöst
- Schnell genug für interaktive Anwendungen

Zusammenfassung

- CSA hat ein relativ einfaches Konzept
- CSA ist schneller als bestehende Ansätze
- Insbesondere MEAT wird besonders schnell gelöst
- Schnell genug für interaktive Anwendungen

Fragen?