

Arc-Flags & Trip-Based Routing

July 2023

Ernestine Großmann¹ Jonas Sauer² Christian Schulz¹ Patrick Steil^{1,2}

¹UNIVERSITY OF HEIDELBERG, ²KARLSRUHE INSTITUTE OF TECHNOLOGY



Best 32 min 28 min 1 hr 9 17 min

Universitat Politècnica de Catalunya · Bar

Cathedral of Barcelona, Pla de la Seu, s/r

Leave now Options

Send directions to your phone

<p>3:23 PM—3:51 PM 28 min</p> <p>3:29 PM from Palau Reial 13 min every 5 min</p> <p>Details</p>	
<p>3:21 PM—4:08 PM 47 min</p> <p>33 > V17</p>	
<p>3:23 PM—4:02 PM 39 min</p> <p>L95 > L1</p>	
<p>3:22 PM—4:09 PM 47 min</p> <p>V5 > L1</p>	

The map shows a route starting at Universitat Politècnica de Catalunya and ending at the Cathedral of Barcelona. The route is highlighted in green and includes a callout box stating "28 min every 5 min". The map also shows various landmarks and neighborhoods in Barcelona, including Sant Gervasi, Gràcia, Sants, and L'Hospitalet de Llobregat.

<https://www.google.de/maps>

Timetable

- Stops (train-, bus stations)
- Footpaths (walking)
- Lines (train-, bus lines)
- Trips (a vehicle of a line)
- Stop Events (arrival or departure)

●08:00 → ●08:05 → ●08:10 → ...

●08:30 → ●08:35 → ●08:40 → ...

●09:00 → ●09:05 → ●09:10 → ...



Timetable

- Stops (train-, bus stations)
- Footpaths (walking)
- Lines (train-, bus lines)
- Trips (a vehicle of a line)
- Stop Events (arrival or departure)

●08:00 → ●08:05 → ●08:10 → ...

●08:30 → ●08:35 → ●08:40 → ...

●09:00 → ●09:05 → ●09:10 → ...



Timetable

- Stops (train-, bus stations)
- Footpaths (walking)
- Lines (train-, bus lines)
- Trips (a vehicle of a line)
- Stop Events (arrival or departure)

●08:00 → ●08:05 → ●08:10 → ...

●08:30 → ●08:35 → ●08:40 → ...

●09:00 → ●09:05 → ●09:10 → ...



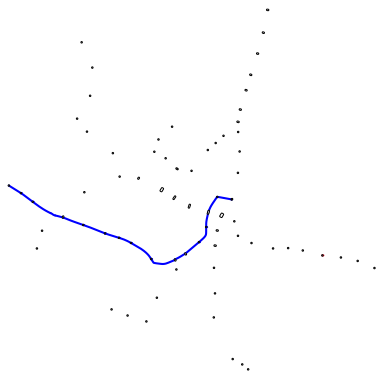
Timetable

- Stops (train-, bus stations)
- Footpaths (walking)
- Lines (train-, bus lines)
- Trips (a vehicle of a line)
- Stop Events (arrival or departure)

●08:00 → ●08:05 → ●08:10 → ...

●08:30 → ●08:35 → ●08:40 → ...

●09:00 → ●09:05 → ●09:10 → ...



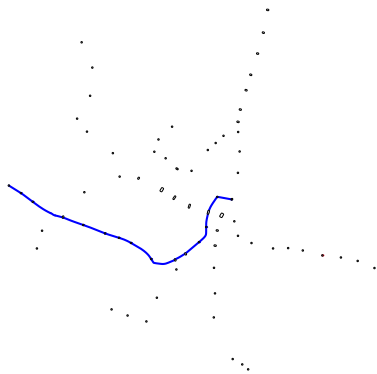
Timetable

- Stops (train-, bus stations)
- Footpaths (walking)
- Lines (train-, bus lines)
- Trips (a vehicle of a line)
- Stop Events (arrival or departure)

●08:00 → ●08:05 → ●08:10 → ...

●08:30 → ●08:35 → ●08:40 → ...

●09:00 → ●09:05 → ●09:10 → ...

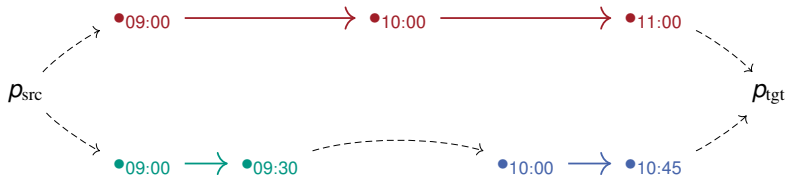


Problem Statement

Definition

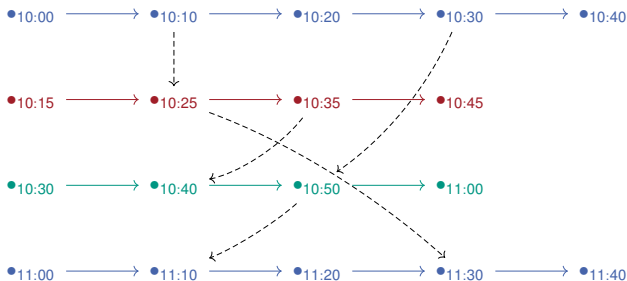
Given a source p_{src} , target p_{tgt} and fixed departure time π , find a set of journeys that minimizes the arrival time and number of trips.

$$\pi = 09:00$$



Time-Expanded Graph [5]

- nodes (*stop-events*) represent arrival (or departure) of a vehicle at a stop
- 2 types of edges: *trip- and transfer-edges*
- Dijkstra too slow



Time-Expanded Graph [5]

- nodes (*stop-events*) represent arrival (or departure) of a vehicle at a stop
- 2 types of edges: *trip*- and *transfer*-edges
- Dijkstra too slow

•10:00 → •10:10 → •10:20 → •10:30 → •10:40

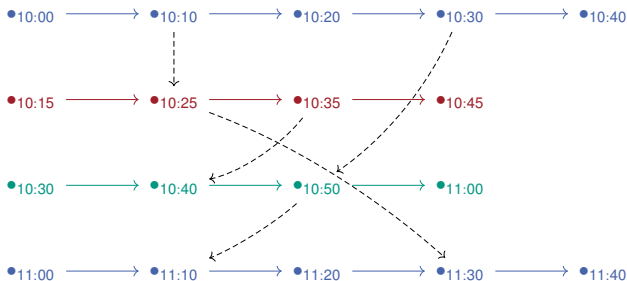
•10:15 → •10:25 → •10:35 → •10:45

•10:30 → •10:40 → •10:50 → •11:00

•11:00 → •11:10 → •11:20 → •11:30 → •11:40

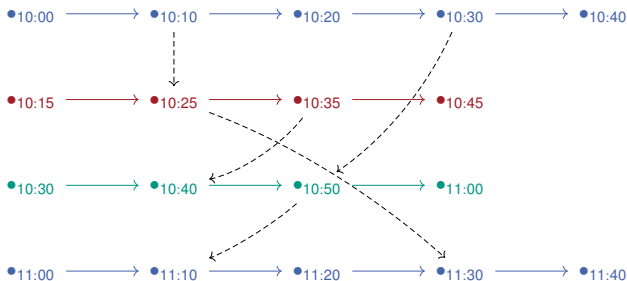
Time-Expanded Graph [5]

- nodes (*stop-events*) represent arrival (or departure) of a vehicle at a stop
- 2 types of edges: *trip*- and *transfer*-edges
- Dijkstra too slow

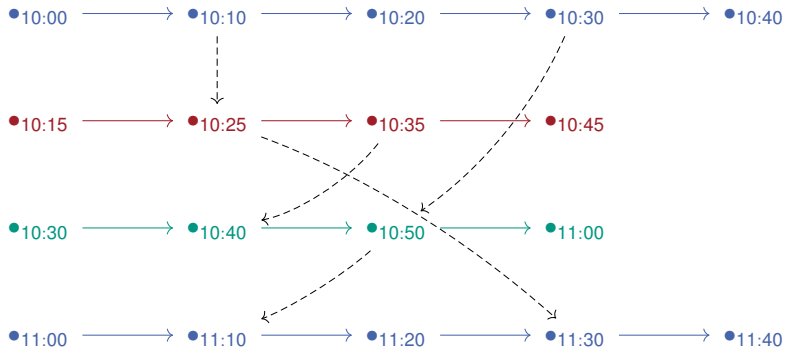


Time-Expanded Graph [5]

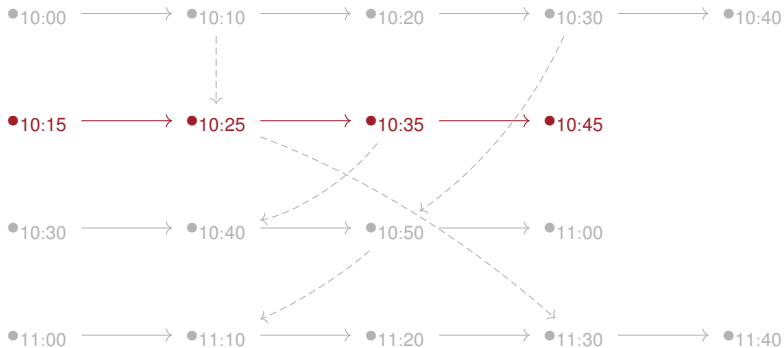
- nodes (*stop-events*) represent arrival (or departure) of a vehicle at a stop
- 2 types of edges: *trip*- and *transfer*-edges
- Dijkstra too slow



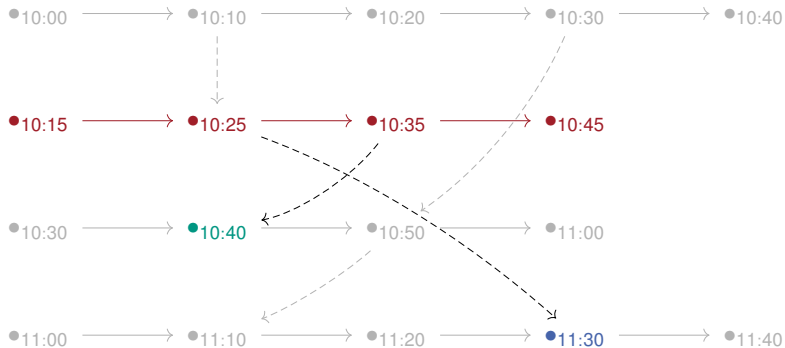
Trip-Based Routing [6]



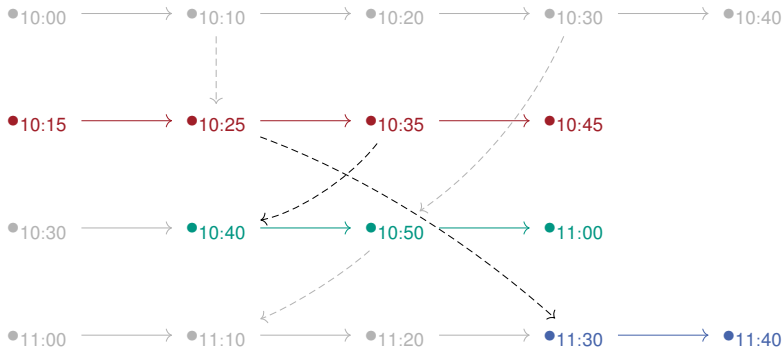
Trip-Based Routing [6]



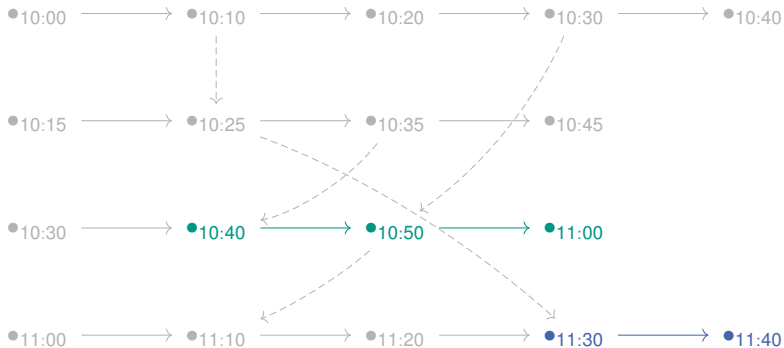
Trip-Based Routing [6]



Trip-Based Routing [6]



Trip-Based Routing [6]



Public Transit vs. Road

- TB on Germany: ≈ 40.8 ms [6]
 - Contraction Hierarchies on Europe: ≈ 70 μ s [3]
- \Rightarrow need for speedup techniques in public transit

Public Transit vs. Road

- TB on Germany: ≈ 40.8 ms [6]
- Contraction Hierarchies on Europe: ≈ 70 μ s [3]

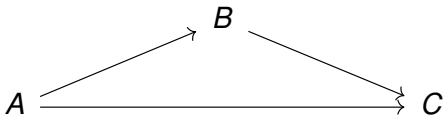
\Rightarrow need for speedup techniques in public transit

Public Transit vs. Road

- TB on Germany: ≈ 40.8 ms [6]
 - Contraction Hierarchies on Europe: ≈ 70 μ s [3]
- \Rightarrow need for speedup techniques in public transit

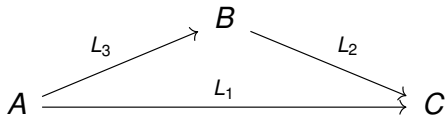
Condensed Search Trees [7]

- idea based on Transfer Patterns [1]
- very fast queries
- preprocessing & memory consumption $\Omega(|\mathcal{S}|^2)$



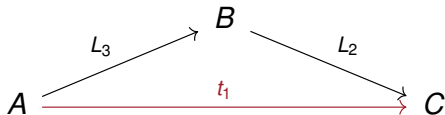
Condensed Search Trees [7]

- idea based on Transfer Patterns [1]
- very fast queries
- preprocessing & memory consumption $\Omega(|\mathcal{S}|^2)$



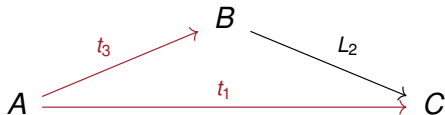
Condensed Search Trees [7]

- idea based on Transfer Patterns [1]
- very fast queries
- preprocessing & memory consumption $\Omega(|\mathcal{S}|^2)$



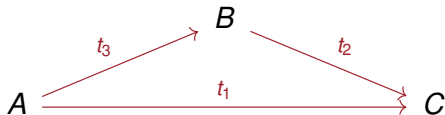
Condensed Search Trees [7]

- idea based on Transfer Patterns [1]
- very fast queries
- preprocessing & memory consumption $\Omega(|\mathcal{S}|^2)$



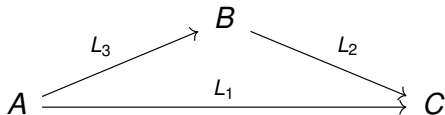
Condensed Search Trees [7]

- idea based on Transfer Patterns [1]
- very fast queries
- preprocessing & memory consumption $\Omega(|\mathcal{S}|^2)$



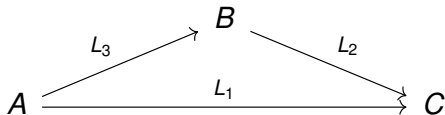
Condensed Search Trees [7]

- idea based on Transfer Patterns [1]
- very fast queries
- preprocessing & memory consumption $\Omega(|\mathcal{S}|^2)$



Condensed Search Trees [7]

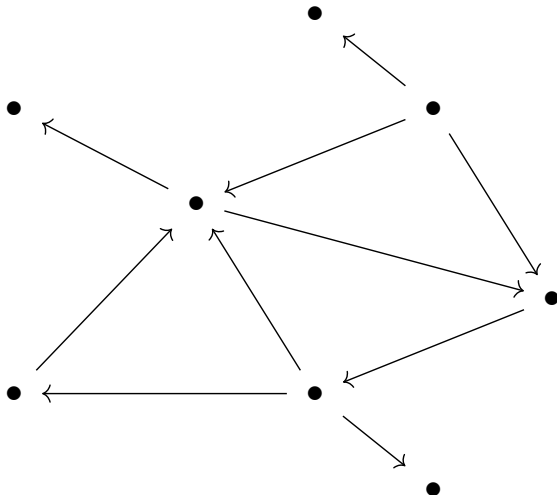
- idea based on Transfer Patterns [1]
- very fast queries
- preprocessing & memory consumption $\Omega(|\mathcal{S}|^2)$



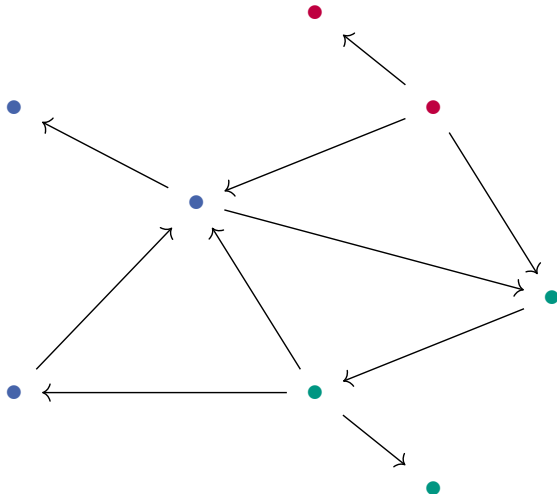
Arc-Flags [4]



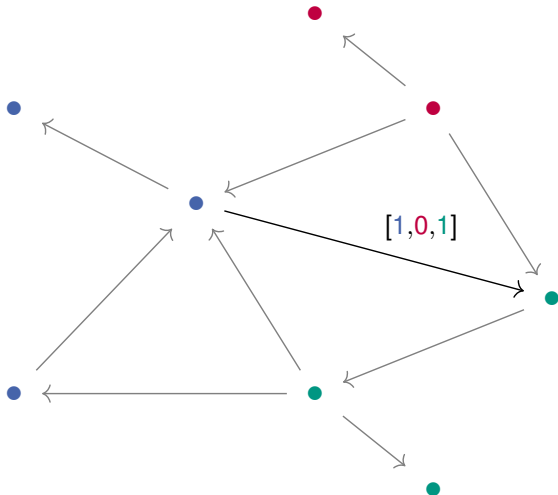
Arc-Flags [4]



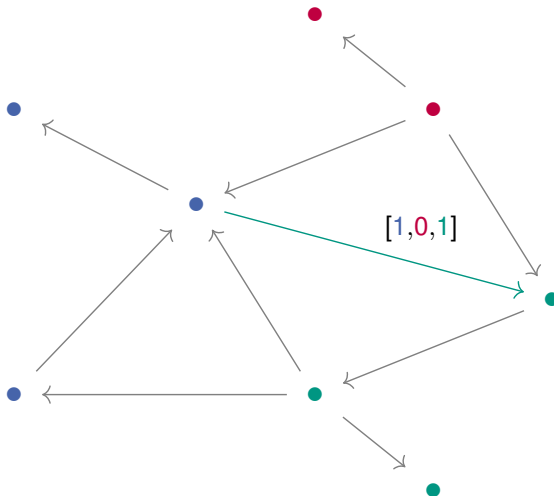
Arc-Flags [4]



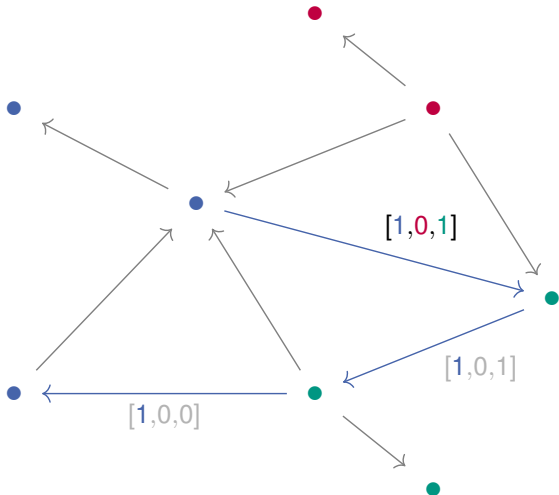
Arc-Flags [4]



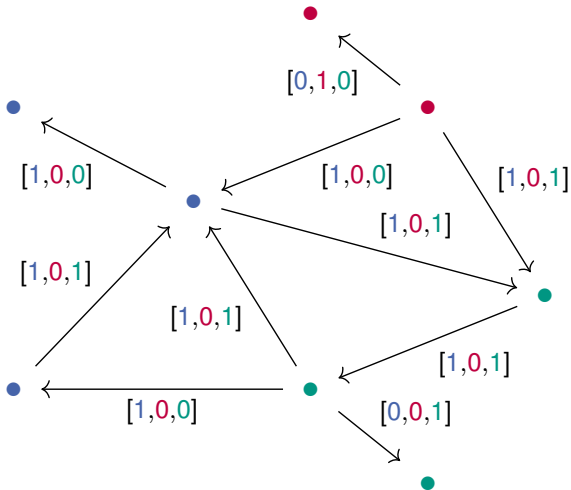
Arc-Flags [4]



Arc-Flags [4]



Arc-Flags [4]



- performed well on road networks ($\approx 5\,000$)
- manageable memory $\Theta(km)$, $k \ll n$
- Arc-Flags on Time-Expanded Graphs [2]
 - “flagged” trip- and transfer-edges
 - noticed conflicts between some pruning rules
 - speedup of ≈ 4

- performed well on road networks ($\approx 5\,000$)
- manageable memory $\Theta(km)$, $k \ll n$
- Arc-Flags on Time-Expanded Graphs [2]
 - “flagged” trip- and transfer-edges
 - noticed conflicts between some pruning rules
 - speedup of ≈ 4

- performed well on road networks ($\approx 5\,000$)
- manageable memory $\Theta(km)$, $k \ll n$
- Arc-Flags on Time-Expanded Graphs [2]
 - “flagged” trip- and transfer-edges
 - noticed conflicts between some pruning rules
 - speedup of ≈ 4

- performed well on road networks ($\approx 5\,000$)
- manageable memory $\Theta(km)$, $k \ll n$
- Arc-Flags on Time-Expanded Graphs [2]
 - “flagged” trip- and transfer-edges
 - noticed conflicts between some pruning rules
 - speedup of ≈ 4

- performed well on road networks ($\approx 5\,000$)
- manageable memory $\Theta(km)$, $k \ll n$
- Arc-Flags on Time-Expanded Graphs [2]
 - “flagged” trip- and transfer-edges
 - noticed conflicts between some pruning rules
 - speedup of ≈ 4

- performed well on road networks ($\approx 5\,000$)
- manageable memory $\Theta(km)$, $k \ll n$
- Arc-Flags on Time-Expanded Graphs [2]
 - “flagged” trip- and transfer-edges
 - noticed conflicts between some pruning rules
 - speedup of ≈ 4

Key idea: flag *transfer* edges

- 1 partition stops into k cells
- 2 compute flags by performing *profile queries*
- 3 remove transfer edges with no flags set to true

Query same as TB, but only relax “flagged” transfer edges

Key idea: flag *transfer* edges

- 1 partition stops into k cells
- 2 compute flags by performing *profile queries*
- 3 remove transfer edges with no flags set to true

Query same as TB, but only relax “flagged” transfer edges

Key idea: flag *transfer* edges

- 1 partition stops into k cells
- 2 compute flags by performing *profile queries*
- 3 remove transfer edges with no flags set to true

Query same as TB, but only relax “flagged” transfer edges

Key idea: flag *transfer* edges

- 1 partition stops into k cells
- 2 compute flags by performing *profile queries*
- 3 remove transfer edges with no flags set to true

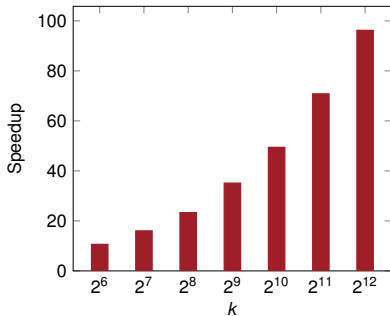
Query same as TB, but only relax “flagged” transfer edges

Key idea: flag *transfer* edges

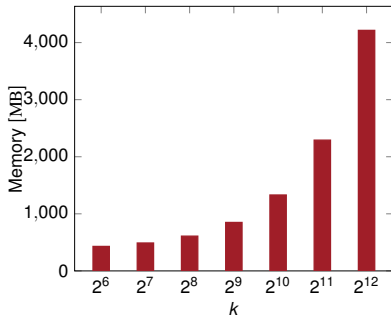
- 1 partition stops into k cells
- 2 compute flags by performing *profile queries*
- 3 remove transfer edges with no flags set to true

Query same as TB, but only relax “flagged” transfer edges

Speedup



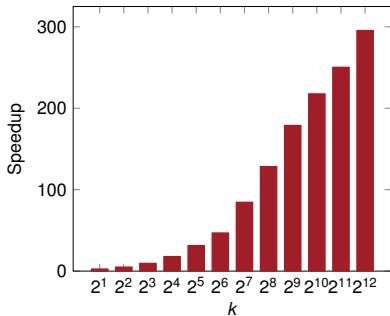
Memory



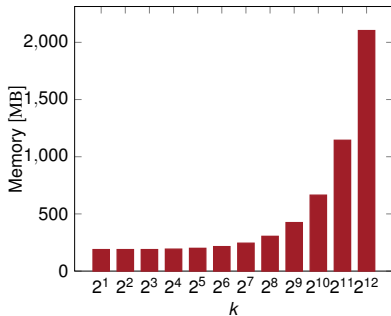
k	Prepro. [hh:mm:ss]	Query time [μ s]	Scanned trips	Memory [MB]
1	—	3 751	20 574	—
64	00:42:23	353	924	429
128	00:42:23	234	580	489
256	00:41:29	161	377	609
512	00:43:06	107	241	849
1 024	00:44:14	76	165	1 330
2 048	00:42:48	53	111	2 291
4 096	00:44:28	39	81	4 214

Results Switzerland

Speedup



Memory



Results Switzerland

k	Prepro. [hh:mm:ss]	Query time [μ s]	Scanned trips	Memory [MB]
1	–	5 005	35 951	–
2	00:16:13	2 001	12 142	188
4	00:16:17	1 023	6 229	188
8	00:15:53	529	3 082	188
16	00:16:34	281	1 601	192
32	00:15:47	160	846	199
64	00:14:29	107	491	214
128	00:13:24	59	289	244
256	00:12:06	39	172	304
512	00:12:03	28	116	424
1 024	00:11:48	23	87	664
2 048	00:11:46	20	69	1 144
4 096	00:11:52	17	57	2 103

Comparison to TB-CST

Network	Arc-Flag TB		TB-CST	
	Query [μ s]	Mem [MB]	Query [μ s]	Mem [MB]
Germany	140	18 923	(156)	114 080
Paris	39	4 214	507	6 992
Sweden	16	2 250	91	3 400
Switzerland	20	1 144	66	1 586

Why is it performing so well?

- conflicting pruning rules
 - Arc-Flags on Time-Expanded Graphs
 - TB-CST & Arc-Flag TB (initially)
- we resolve these issues and adapted Arc-Flag TB (and TB-CST)

Why is it performing so well?

- conflicting pruning rules
 - Arc-Flags on Time-Expanded Graphs
 - TB-CST & Arc-Flag TB (initially)
- we resolve these issues and adapted Arc-Flag TB (and TB-CST)

Profile- vs Fixed Departure Queries

- Profile-TB finds the latest trips possible
- TB always enters the earliest trip of a route
- some *necessary* transfers are not being “flagged”



Profile- vs Fixed Departure Queries

- Profile-TB finds the latest trips possible
- TB always enters the earliest trip of a route
- some *necessary* transfers are not being “flagged”



Profile- vs Fixed Departure Queries

- Profile-TB finds the latest trips possible
- TB always enters the earliest trip of a route
- some *necessary* transfers are not being “flagged”



Profile- vs Fixed Departure Queries

- Profile-TB finds the latest trips possible
- TB always enters the earliest trip of a route
- some *necessary* transfers are not being “flagged”



- Arc-Flags performs excellent in public transit context
- still $\Omega(|\mathcal{S}|^2)$ preprocessing, but manageable memory
- Arc-Flag TB new state-of-the-art
- Code: github.com/TransitRouting/Arc-FlagTB

Thank you!

- Arc-Flags performs excellent in public transit context
- still $\Omega(|\mathcal{S}|^2)$ preprocessing, but manageable memory
- Arc-Flag TB new state-of-the-art
- Code: github.com/TransitRouting/Arc-FlagTB

Thank you!

- Arc-Flags performs excellent in public transit context
- still $\Omega(|\mathcal{S}|^2)$ preprocessing, but manageable memory
- Arc-Flag TB new state-of-the-art
- Code: github.com/TransitRouting/Arc-FlagTB

Thank you!

- Arc-Flags performs excellent in public transit context
- still $\Omega(|\mathcal{S}|^2)$ preprocessing, but manageable memory
- Arc-Flag TB new state-of-the-art
- Code: github.com/TransitRouting/Arc-FlagTB

Thank you!

- [1] H. Bast, E. Carlsson, A. Eigenwillig, R. Geisberger, C. Harrelson, V. Raychev, and F. Viger. Fast routing in very large public transportation networks using transfer patterns. In M. de Berg and U. Meyer, editors, *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I*, volume 6346 of *Lecture Notes in Computer Science*, pages 290–301. Springer, 2010.
- [2] J. Dibbelt, T. Pajor, B. Strasser, and D. Wagner. Connection scan algorithm. *ACM J. Exp. Algorithmics*, 23, 2018.

- [3] R. Geisberger, P. Sanders, D. Schultes, and D. Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In C. C. McGeoch, editor, *Experimental Algorithms, 7th International Workshop, WEA 2008, Provincetown, MA, USA, May 30-June 1, 2008, Proceedings*, volume 5038 of *Lecture Notes in Computer Science*, pages 319–333. Springer, 2008.
- [4] R. H. Möhring II, H. Schilling, B. Schütz, D. Wagner, and T. Willhalm. Partitioning graphs to speed up dijkstra's algorithm. In S. E. Nikolettseas, editor, *Experimental and Efficient Algorithms, 4th International Workshop, WEA 2005, Santorini Island, Greece, May 10-13, 2005, Proceedings*, volume 3503 of *Lecture Notes in Computer Science*, pages 189–202. Springer, 2005.

- [5] E. Pyrga, F. Schulz, D. Wagner, and C. D. Zaroliagis. Efficient models for timetable information in public transportation systems. *ACM J. Exp. Algorithmics*, 12:2.4:1–2.4:39, 2007.
- [6] S. Witt. Trip-based public transit routing. In N. Bansal and I. Finocchi, editors, *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 1025–1036. Springer, 2015.
- [7] S. Witt. Trip-based public transit routing using condensed search trees. In M. Goerigk and R. F. Werneck, editors, *16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, ATMOS 2016, August 25, 2016, Aarhus, Denmark*, volume 54 of *OASICS*, pages 10:1–10:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.