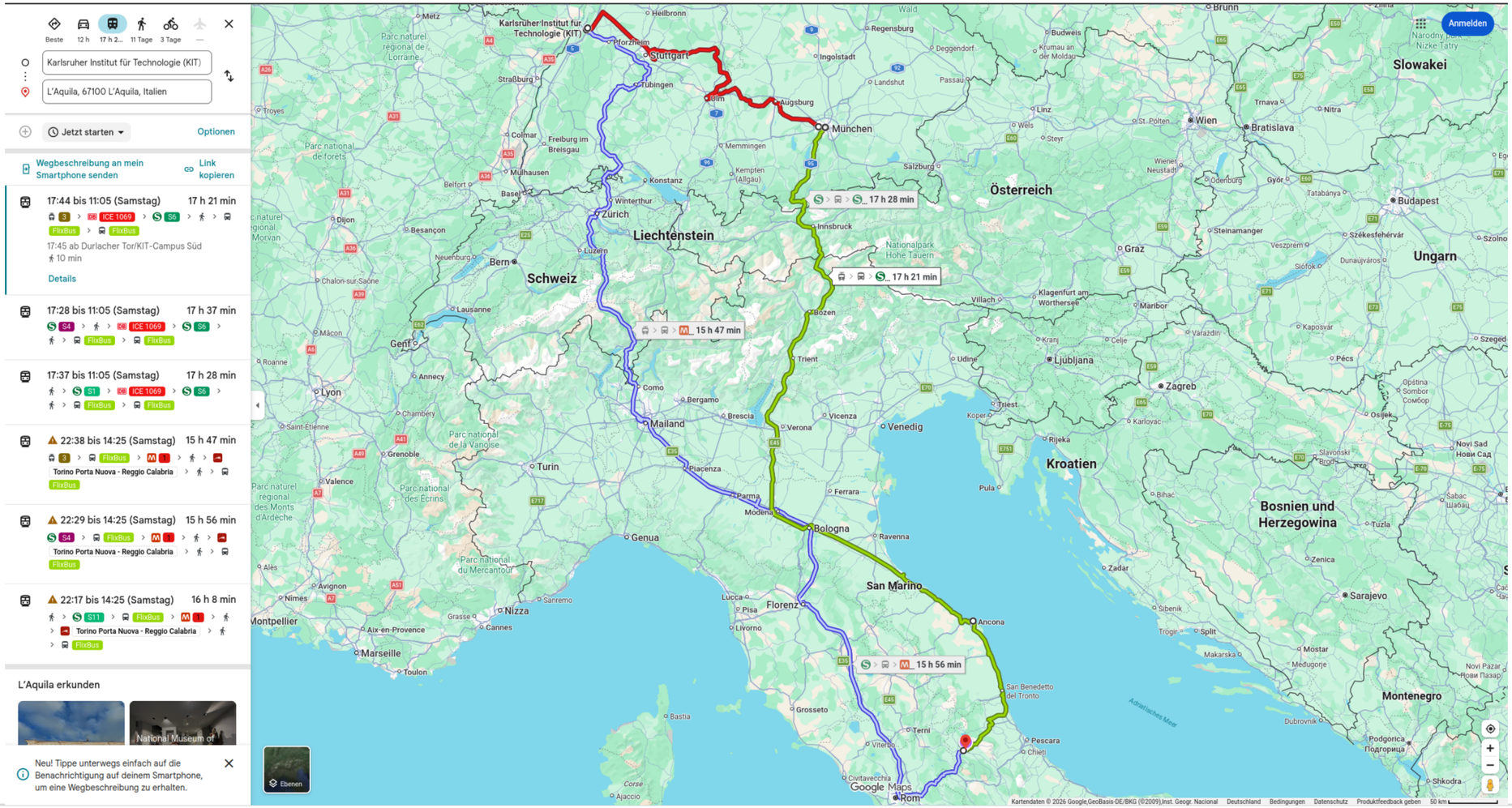
Karlsruhe Institute of Technology

T-REX: Transfer-Ranked EXploration

September 1, 2026

Jonas Sauer, Patrick Steil, Sascha Witt

Motivation


Karlsruher Institut für Technologie (KIT)
L'Aquila, 67100 L'Aquila, Italien
Jetzt starten
Optionen
Wegbeschreibung an mein Smartphone senden
Link kopieren
17:44 bis 11:05 (Samstag) 17 h 21 min
17:45 ab Durlacher Tor/KIT-Campus Süd
10 min
Details
17:28 bis 11:05 (Samstag) 17 h 37 min
17:37 bis 11:05 (Samstag) 17 h 28 min
22:38 bis 14:25 (Samstag) 15 h 47 min
22:29 bis 14:25 (Samstag) 15 h 56 min
22:17 bis 14:25 (Samstag) 16 h 8 min
L'Aquila erkunden
Neu! Tippe unterwegs einfach auf die Benachrichtigung auf deinem Smartphone, um eine Wegbeschreibung zu erhalten.
17 h 28 min
17 h 21 min
15 h 47 min
15 h 56 min
Österreich
Schweiz
Liechtenstein
Ungarn
Slowakei
Kroatien
Bosnien und Herzegowina
Montenegro
San Marino
Anmelden
Ebenen
Google Maps

Motivation

- routing algorithm should
  1. Pareto-optimal journeys
     - (min arrival time, min number of transfers)
  2. millisecond queries
  3. fast customization after updates

Motivation

- routing algorithm should
  1. Pareto-optimal journeys
     - (min arrival time, min number of transfers)
  2. millisecond queries
  3. fast customization after updates

$\rightarrow$ current state-of-the-art is not satisfactory

Motivation

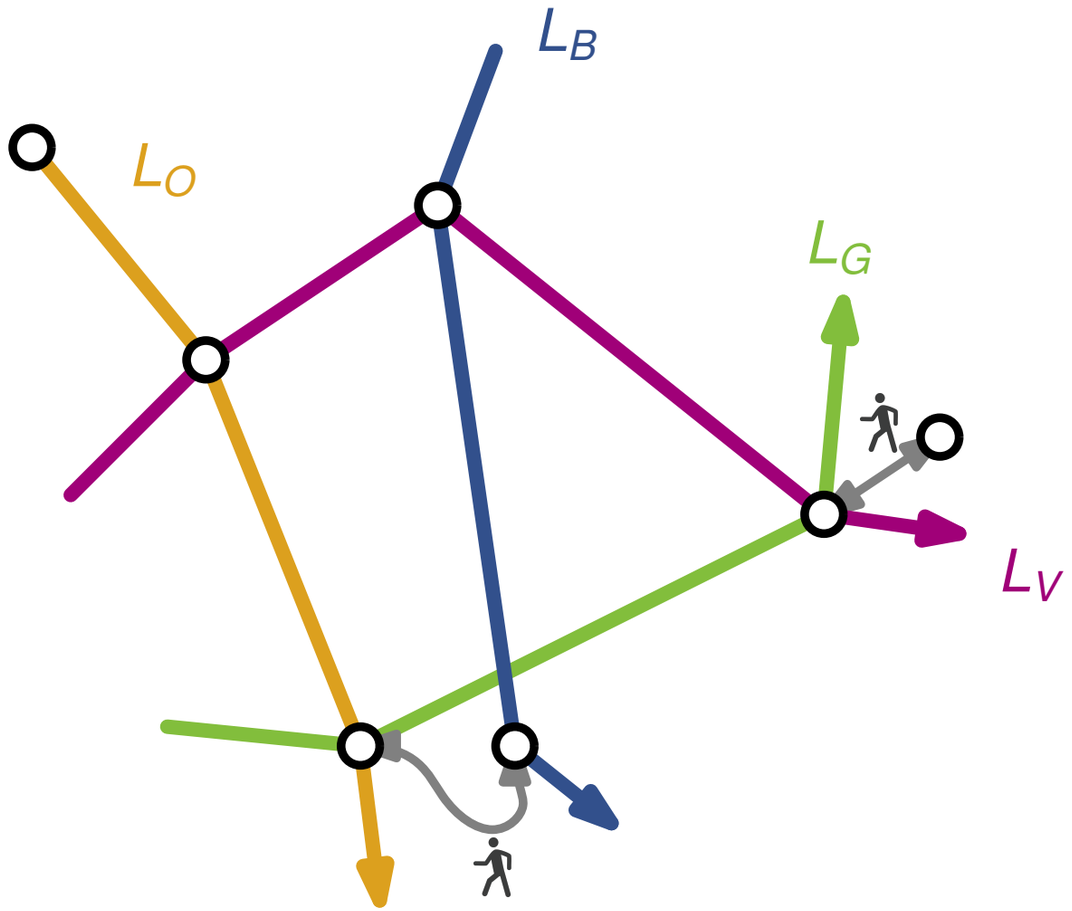
- routing algorithm should
  1. Pareto-optimal journeys
     - (min arrival time, min number of transfers)
  2. millisecond queries
  3. fast customization after updates

$\rightarrow$ current state-of-the-art is not satisfactory

tradeoff between *preprocessing overhead* and *query speed*

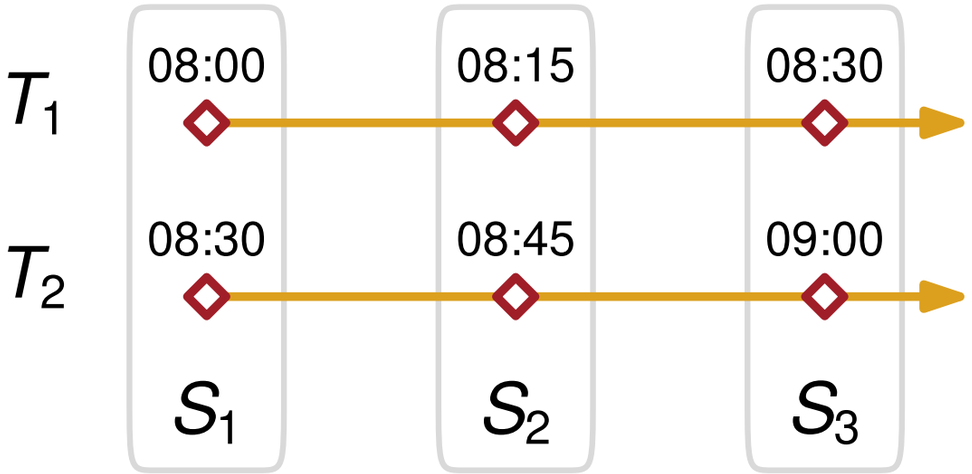
Timetable

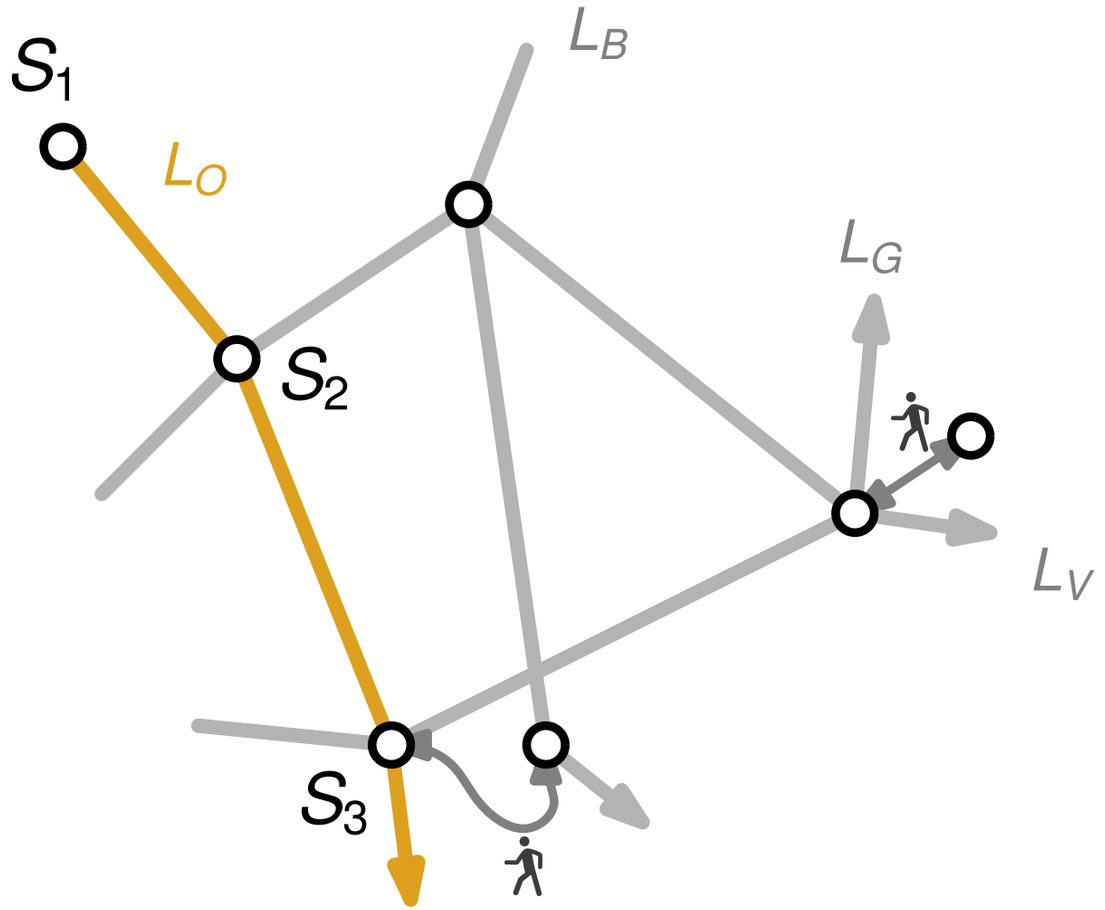
- stops / stations
- footpaths
- routes / lines



Timetable

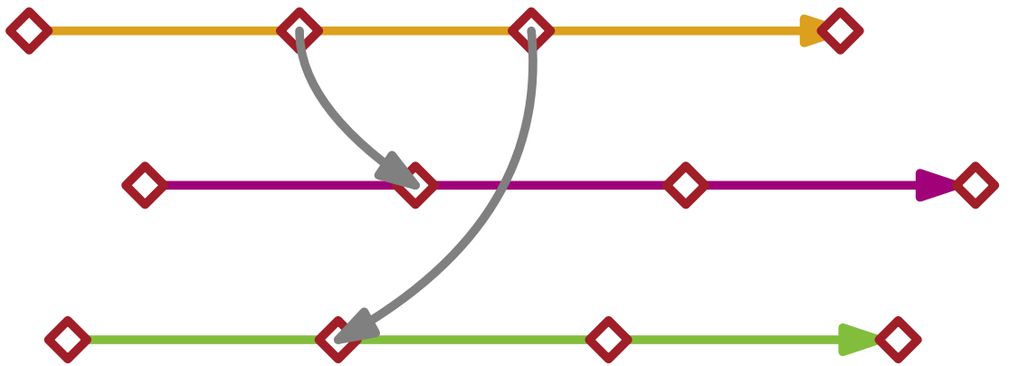
- stops / stations
- footpaths
- routes / lines
- trips





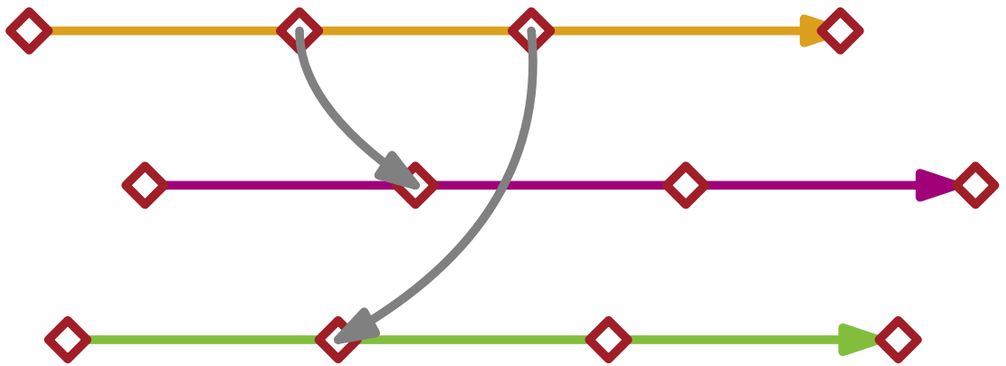
Trip-Based Routing [Witt16]

- process the network in *rounds*
- precompute *sensible* transfers
- BFS-like fashion, queue contains reached trips

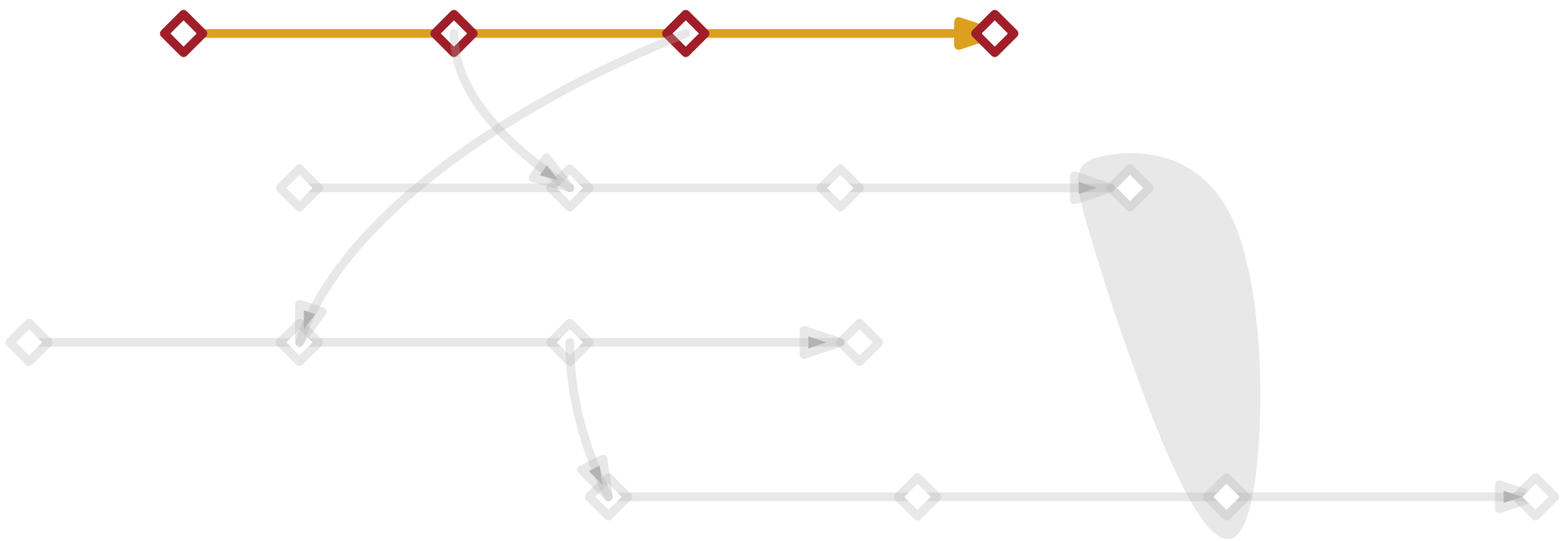


Trip-Based Routing [Witt16]

- process the network in *rounds*
- precompute *sensible* transfers
- BFS-like fashion, queue contains reached trips
  - scan trips
  - *relax* transfers
  - solutions in round i have $i - 1$ transfers

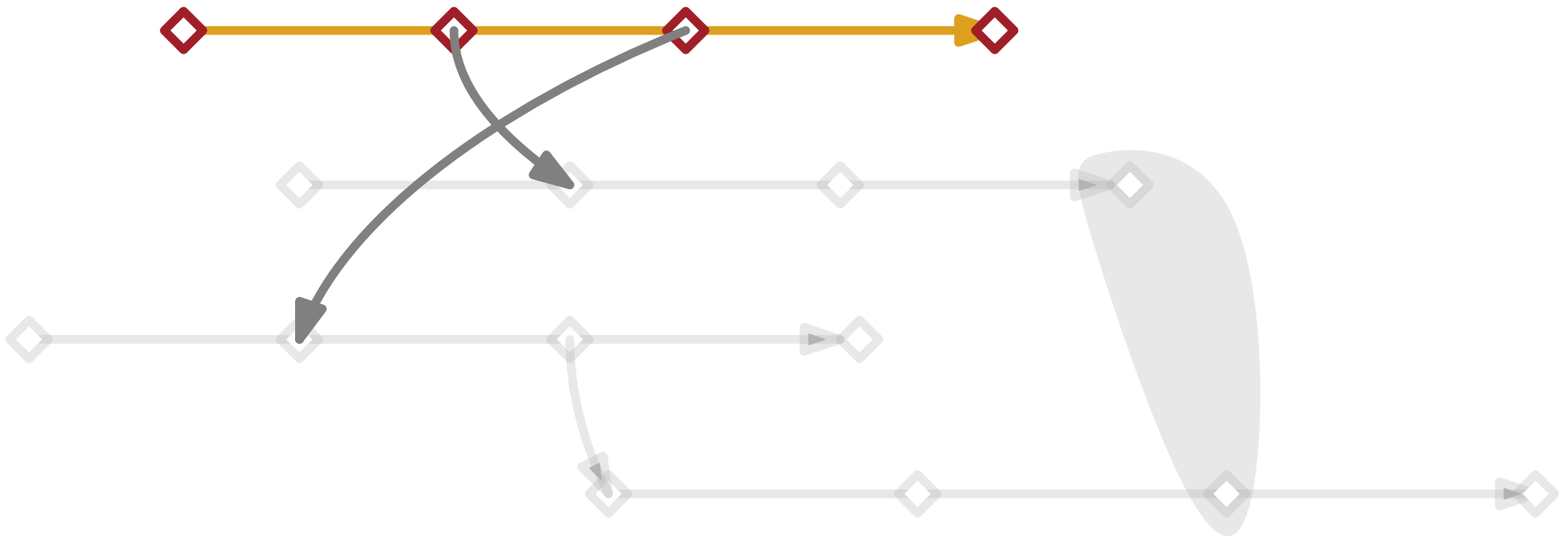


Trip-Based Routing [Witt16]



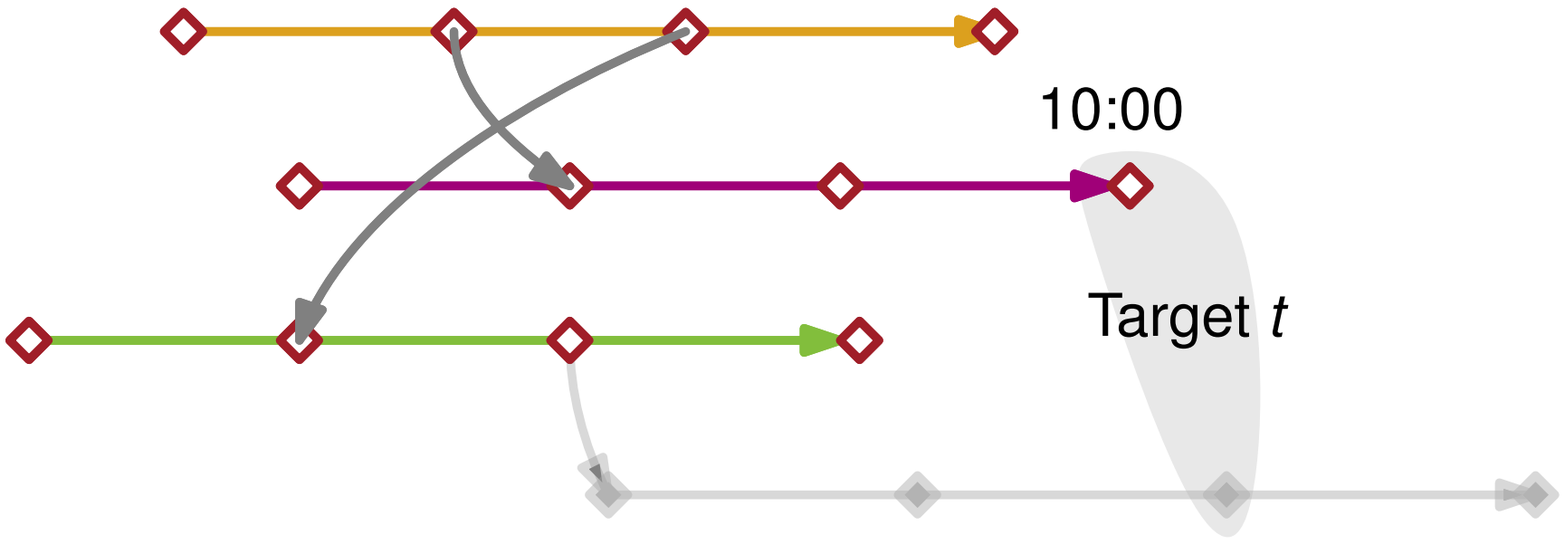
Optimal journeys:

Trip-Based Routing [Witt16]



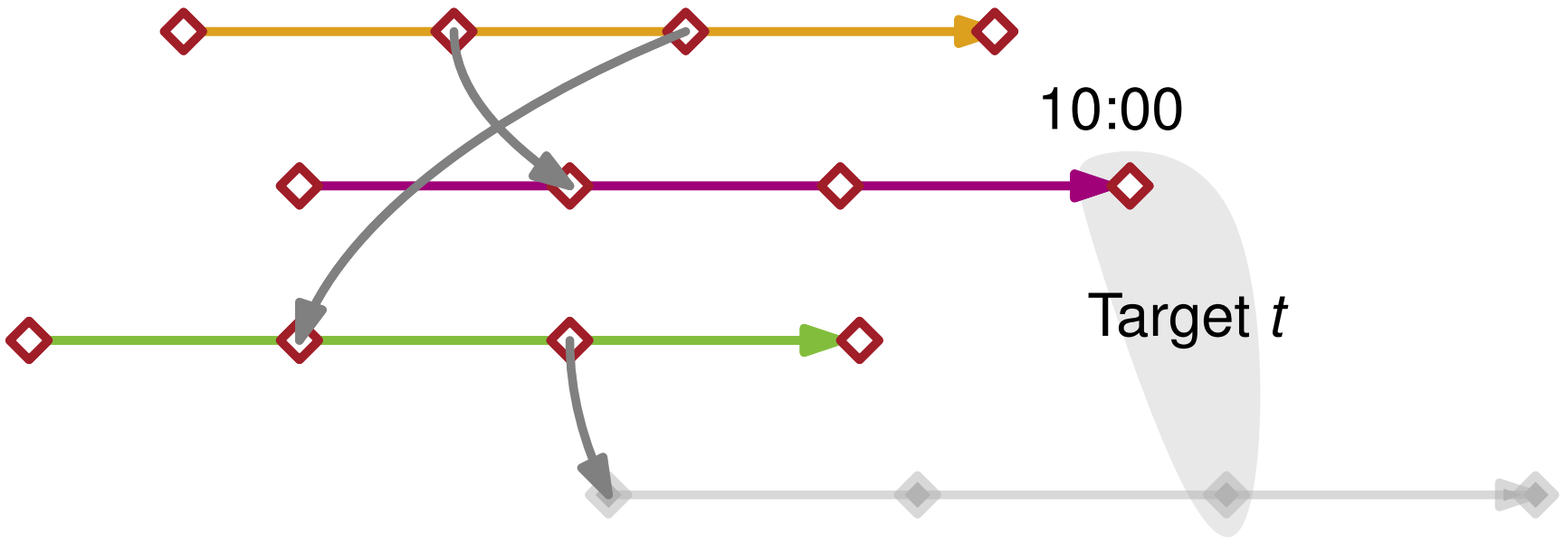
Optimal journeys:

Trip-Based Routing [Witt16]



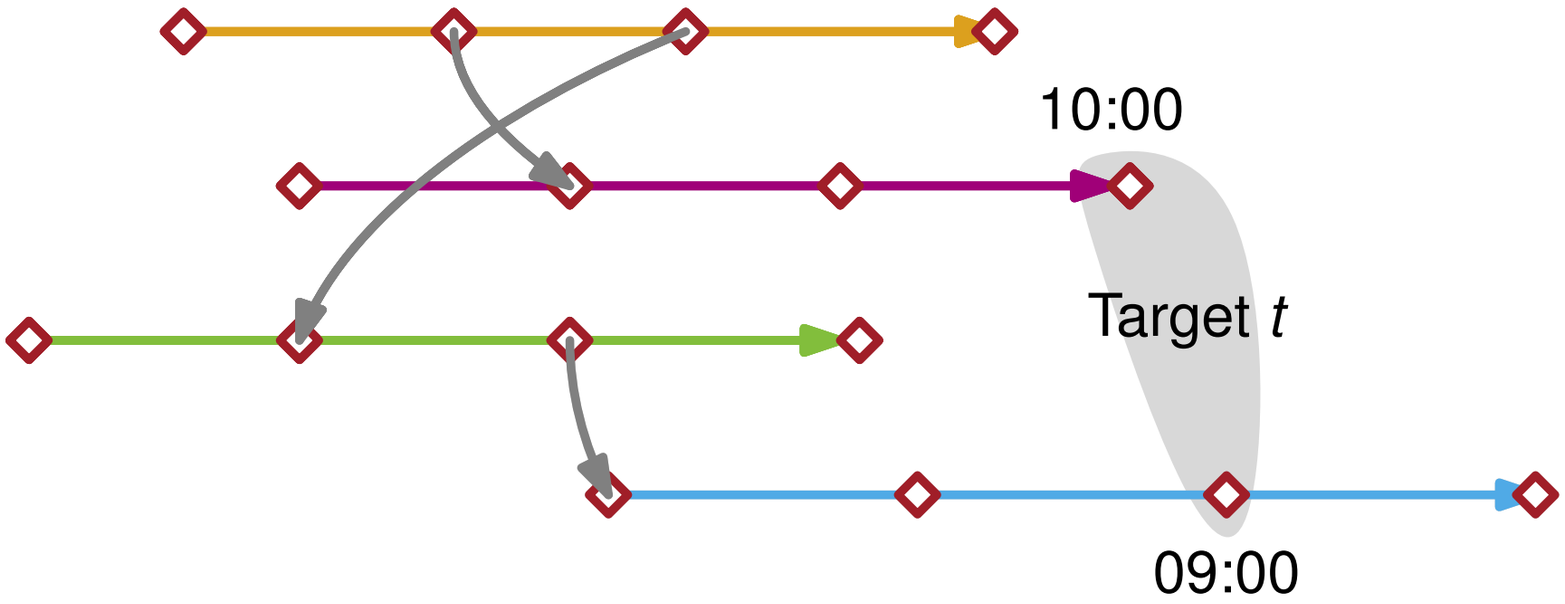
Optimal journeys: $\{(10{:}00, 1)\}$

Trip-Based Routing [Witt16]



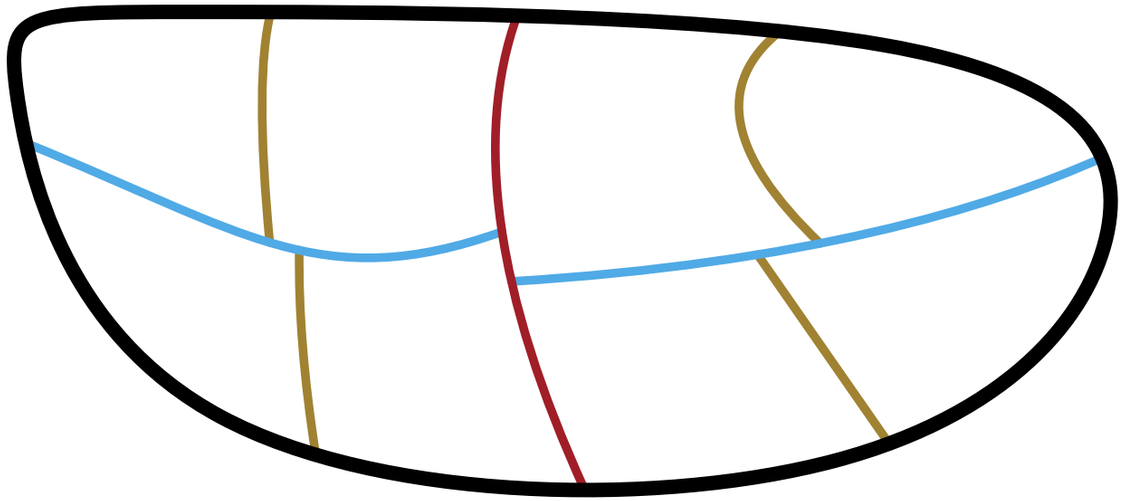
Optimal journeys: $\{(10{:}00, 1)\}$

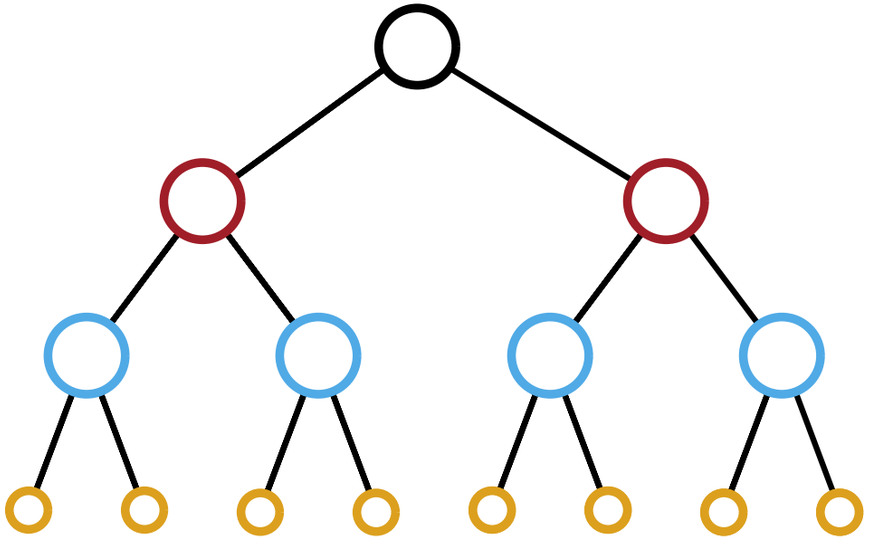
Trip-Based Routing [Witt16]



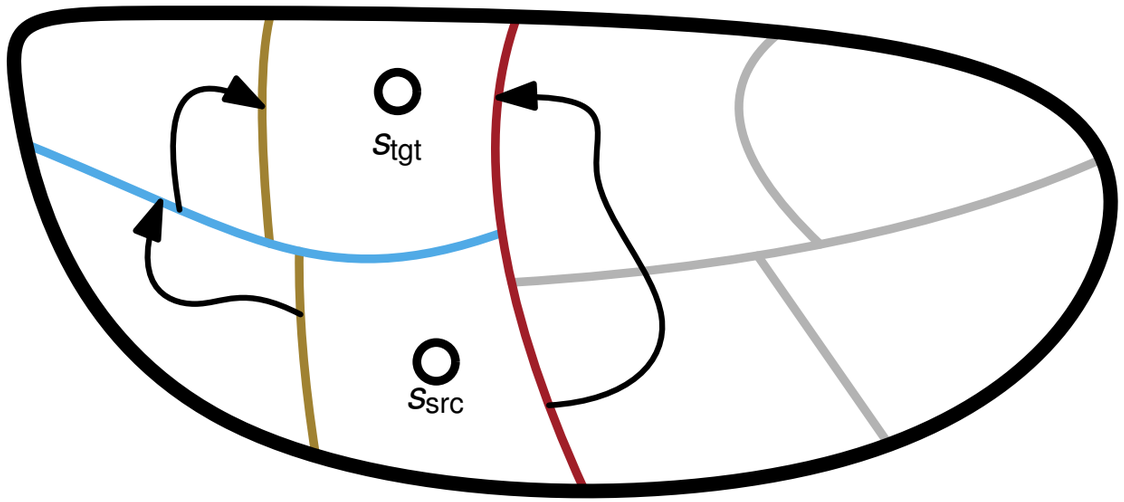
Optimal journeys: $\{(10{:}00, 1), (09{:}00, 2)\}$

Multilevel Overlays



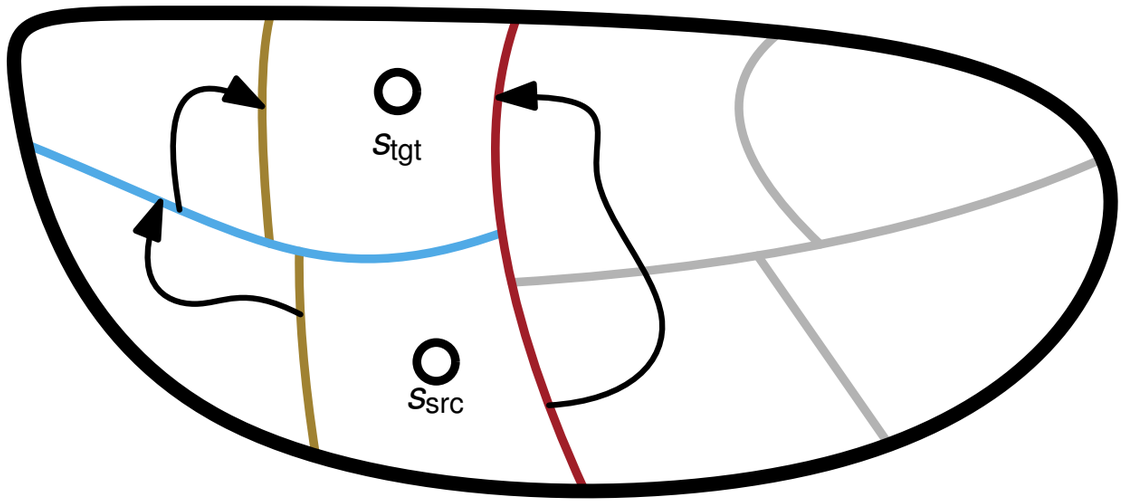


Multilevel Overlays



- Idea:
  1. precompute black *shortcuts*
  2. allows to “skip” unimportant cells during query

Multilevel Overlays



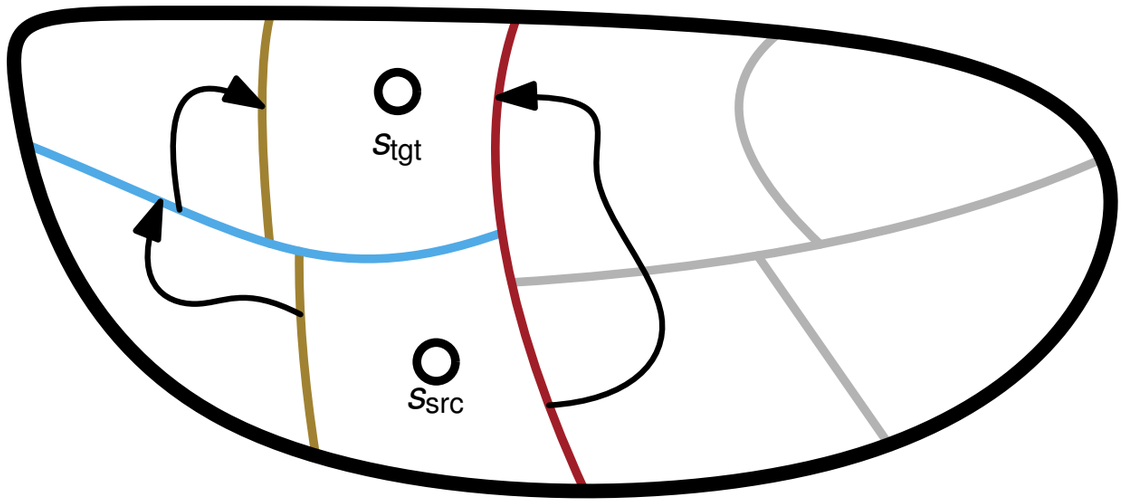
- Idea:
  1. precompute black *shortcuts*
  2. allows to “skip” unimportant cells during query
- Very successful on road-networks

  Preprocessing order of minutes

  Customization order of seconds

  Query order of milliseconds

Multilevel Overlays



- Idea:
  1. precompute black *shortcuts*
  2. allows to “skip” unimportant cells during query
- Very successful on road-networks

  Preprocessing order of minutes

  Customization order of seconds

  Query order of milliseconds

→ What about public transit networks?

T-REX: Transfer-Ranked EXploration

Karlsruhe Institute of Technology

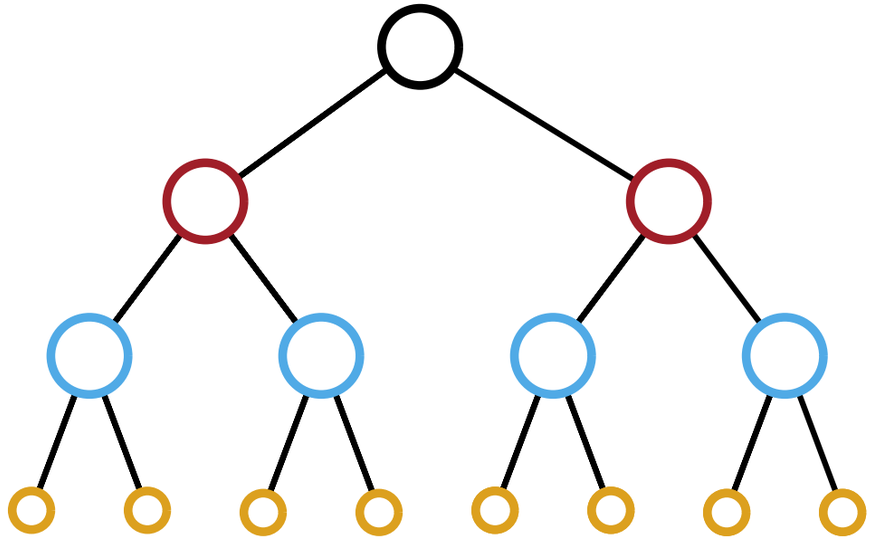
- combines multilevel partitioning with Trip-Based Routing

T-REX: Transfer-Ranked EXploration

Karlsruhe Institute of Technology

- combines multilevel partitioning with Trip-Based Routing
- assign *importance* to transfers

  Preprocessing compute partition and transfers



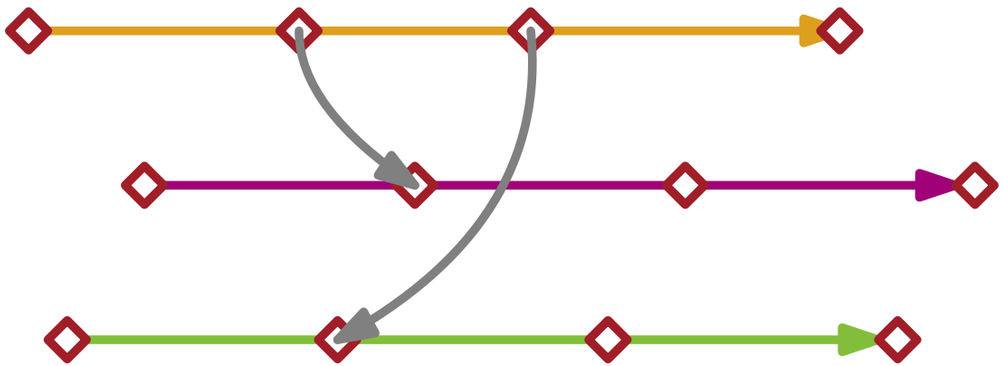
T-REX: Transfer-Ranked EXploration

Karlsruhe Institute of Technology

- combines multilevel partitioning with Trip-Based Routing
- assign *importance* to transfers

  Preprocessing compute partition and transfers

  Customization compute importance of transfers



T-REX: Transfer-Ranked EXploration

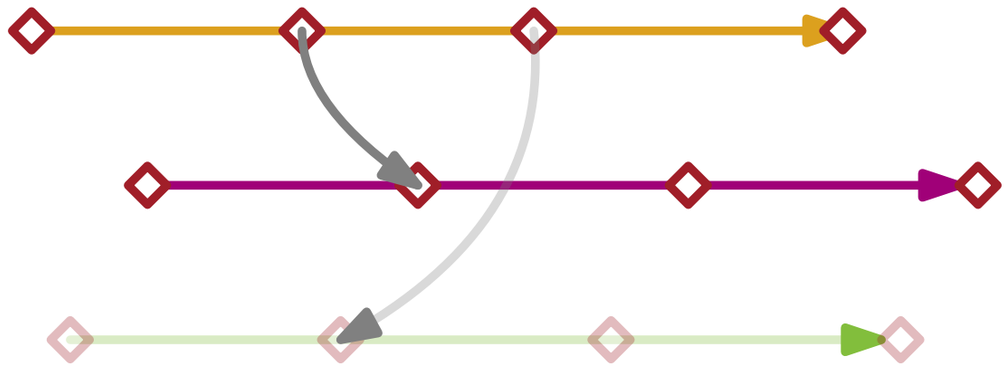
Karlsruhe Institute of Technology

- combines multilevel partitioning with Trip-Based Routing
- assign *importance* to transfers

  Preprocessing compute partition and transfers

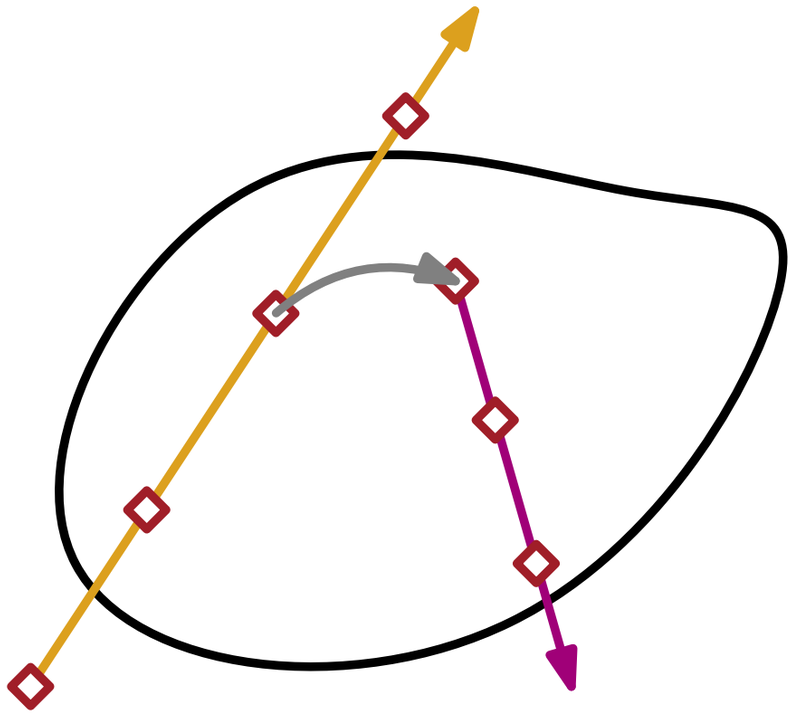
  Customization compute importance of transfers

  Query skip unimportant transfers



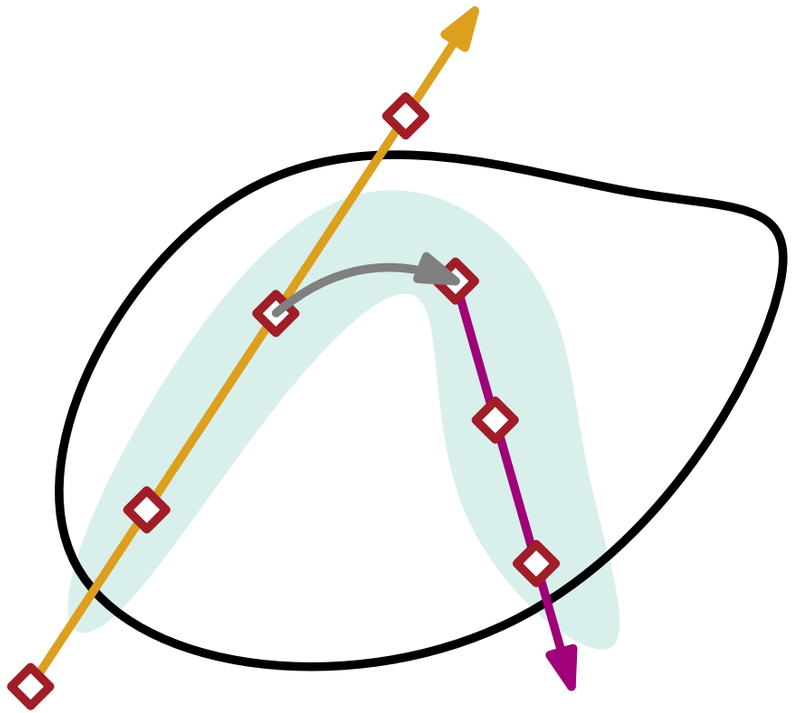
Importance

Customization



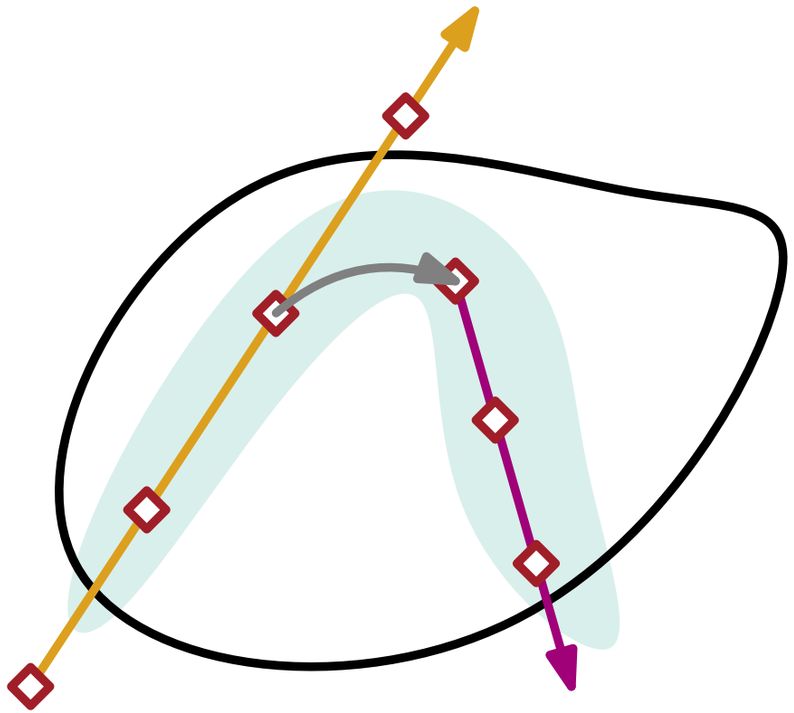
Importance

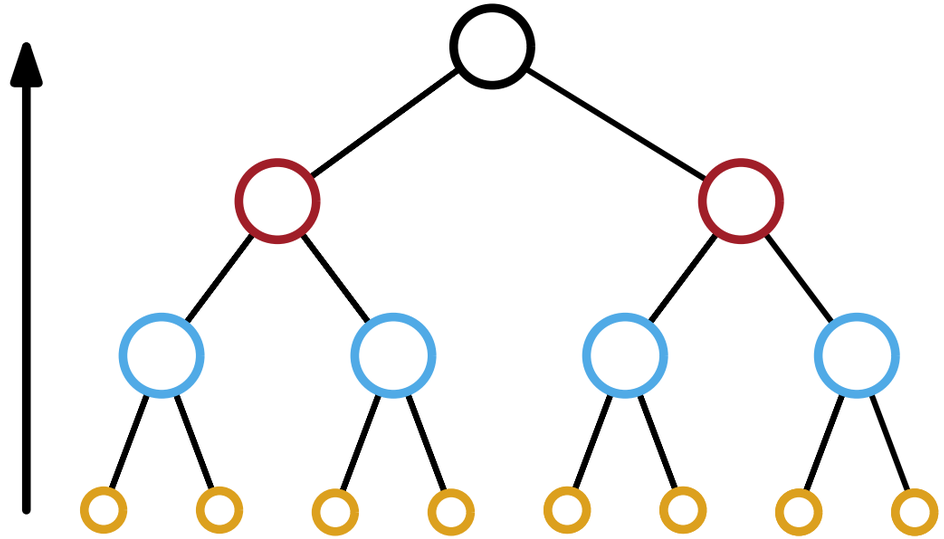
Customization



Importance

Customization

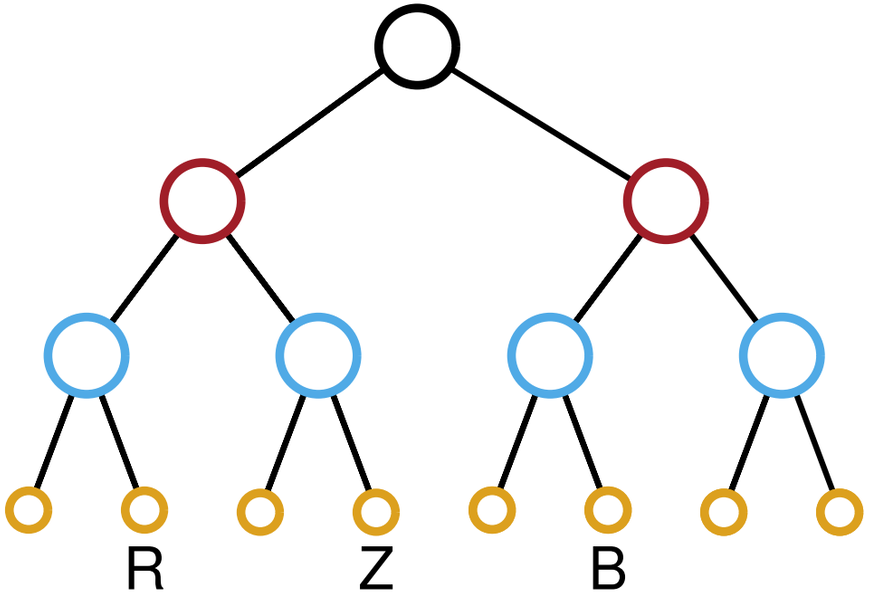




Importance

Query

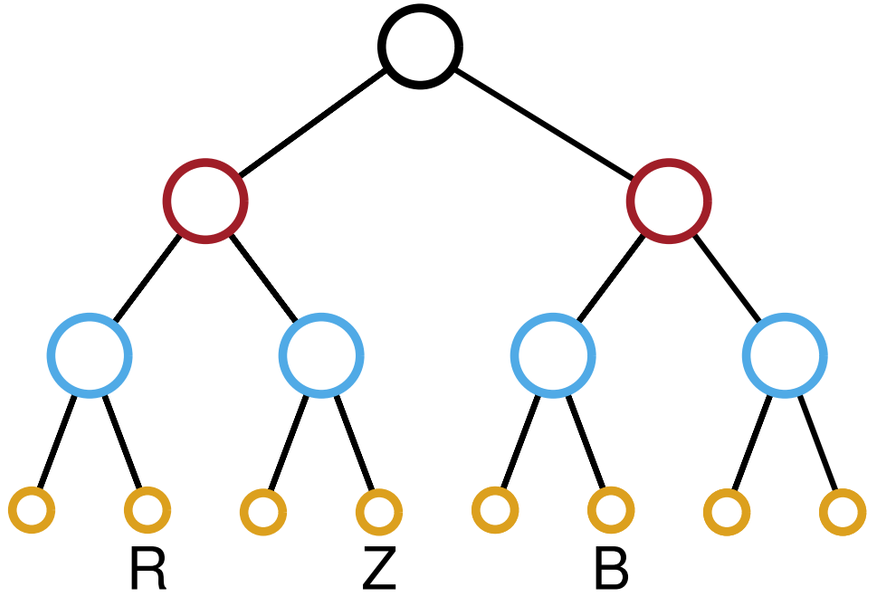
- consider query from Rome (R) to Berlin (B)
- currently at Zürich (Z) $\rightarrow$ which transfer?



Importance

Query

- consider query from Rome (R) to Berlin (B)
- currently at Zürich (Z) $\rightarrow$ which transfer?
- importance $\geq \min\{LCL(\mathrm{R},\mathrm{Z}), LCL(\mathrm{Z},\mathrm{B})\}$

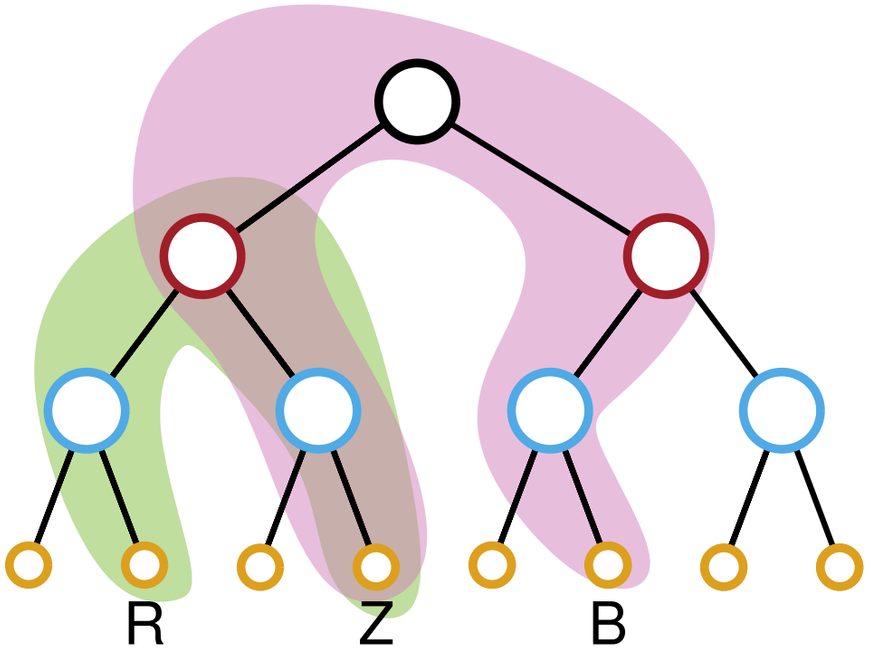


"don't transfer to local bus, rather long distance train"

Importance

Query

- consider query from Rome (R) to Berlin (B)
- currently at Zürich (Z) $\rightarrow$ which transfer?
- importance $\geq \min\{LCL(\mathrm{R},\mathrm{Z}), LCL(\mathrm{Z},\mathrm{B})\}$

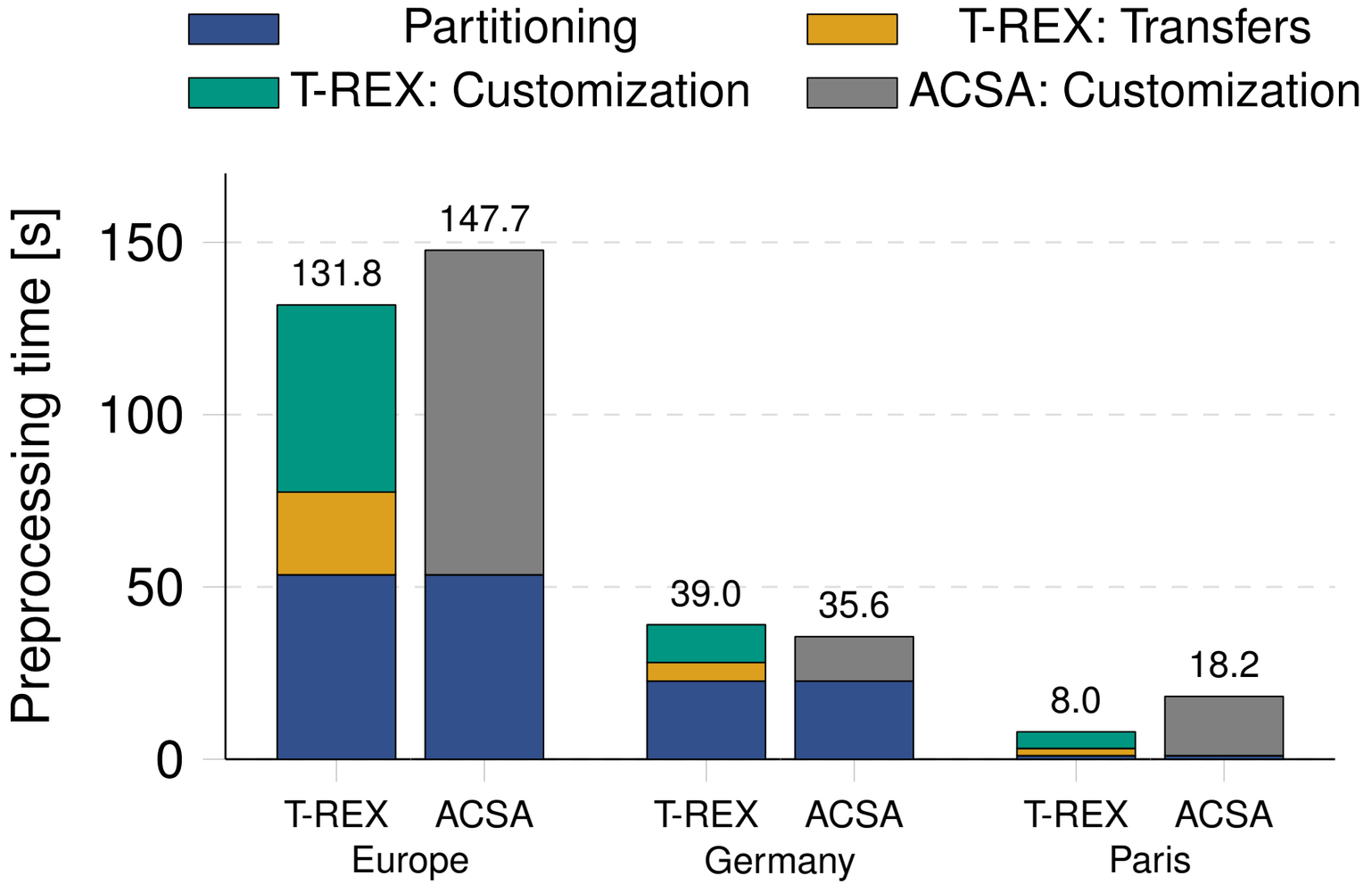


"don't transfer to local bus, rather long distance train"

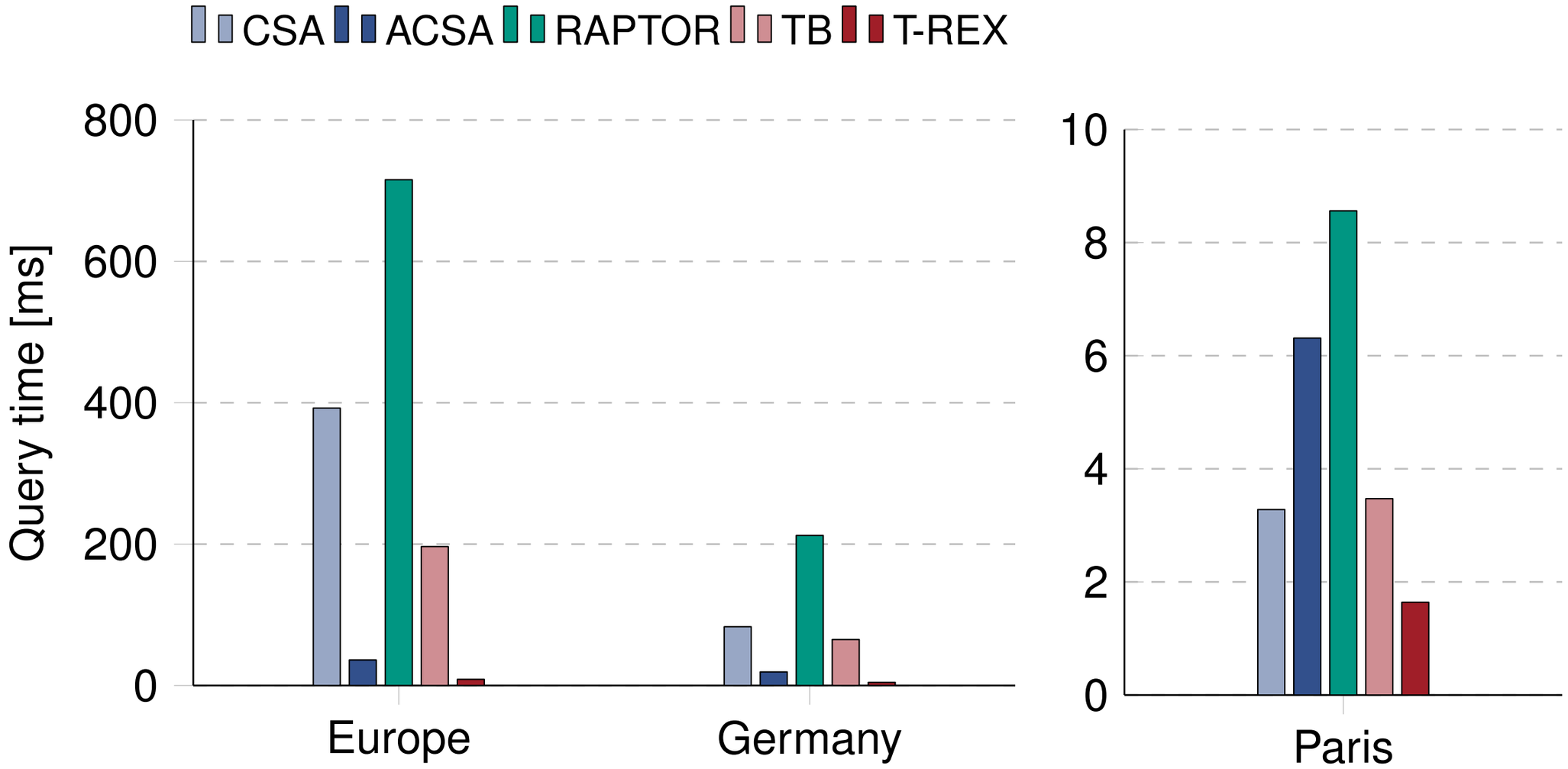
Experiments

	Europe	Germany	Paris
Stops	1 346 013	435 550	41 757
Stop events	107 252 199	30 680 868	4 636 238
Lines	384 075	202 238	9 558
Trips	4 848 876	1 547 126	215 526
Footpaths	1 752 418	1 116 976	445 912

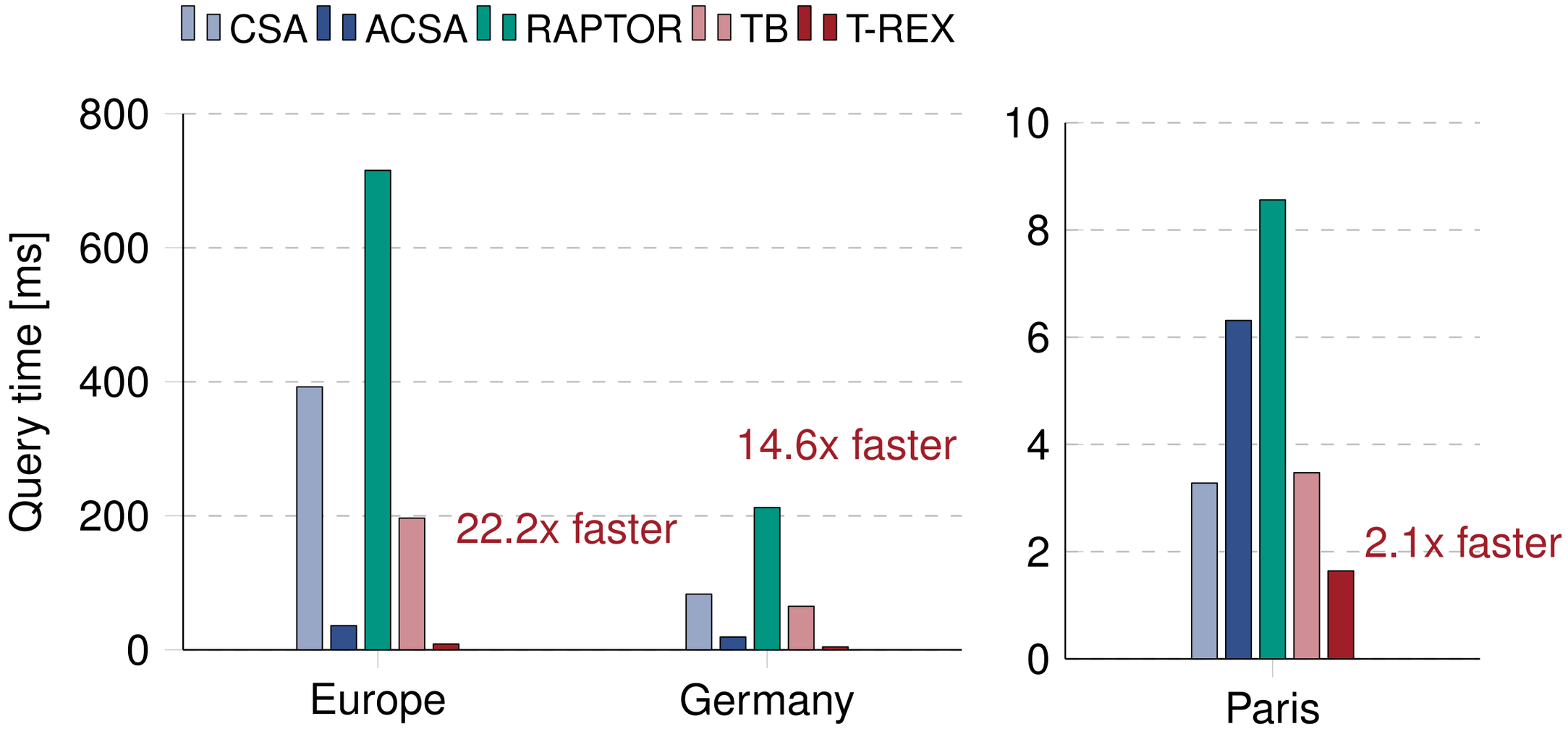
Preprocessing



Query


CSA
ACSA
RAPTOR
TB
T-REX
Query time [ms]
800
600
400
200
0
Europe
Germany
10
8
6
4
2
0
Paris

Query

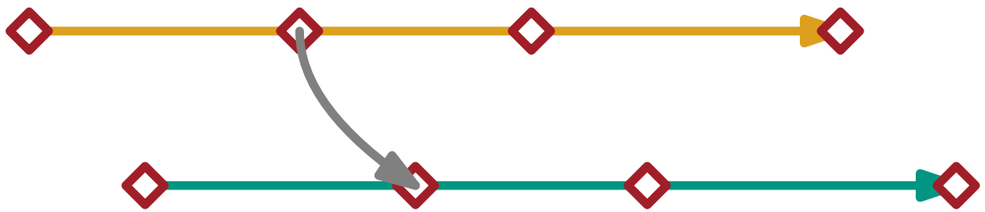

CSA
ACSA
RAPTOR
TB
T-REX
Query time [ms]
800
600
400
200
0
22.2x faster
14.6x faster
Europe
Germany
10
8
6
4
2
0
2.1x faster
Paris

“Why this works now”

1. Better partition: Mt-KaHyPar [GHMSS24]
   - allowing high imbalance
   - making use of social-graph settings

“Why this works now”

1. Better partition: Mt-KaHyPar [GHMSS24]
   - allowing high imbalance
   - making use of social-graph settings
2. Focus on transfers: finer-grained pruning information



“Why this works now”

1. Better partition: Mt-KaHyPar [GHMSS24]
   - allowing high imbalance
   - making use of social-graph settings
2. Focus on transfers: finer-grained pruning information
3. Redesigned query algorithm: cache and branch efficient

“Why this works now”

1. Better partition: Mt-KaHyPar [GHMSS24]
   - allowing high imbalance
   - making use of social-graph settings
2. Focus on transfers: finer-grained pruning information
3. Redesigned query algorithm: cache and branch efficient

$\rightarrow$ Public-transit networks do exhibit hierarchy

Conclusion

- T-REX enables interactive Pareto-optimal routing
  - metropolitain & continental scale
  - low memory overhead
  - fast preprocessing

Conclusion

- T-REX enables interactive Pareto-optimal routing
  - metropolitain & continental scale
  - low memory overhead
  - fast preprocessing
- Future Work
  - allowing unlimited walking
  - combining with goal-directed pruning
  - longer time horizons (timetables spanning a longer period)

Bibliography

[1] Sascha Witt. **Trip-Based Public Transit Routing**. ESA 2015.

[2] Daniel Delling, Thomas Pajor, Renato F. Werneck. **Round-bAsed Public Transit Touting**. *Transportation Science*.

[3] Julian Dibbelt, Thomas Pajor, Ben Strasser, Dorothea Wagner. **Connection Scan Algorithm**. *Journal of Experimental Algorithmics*.

[4] Lars Gottesbüren, Tobias Heuer, Nikolai Maas, Peter Sanders, and Sebastian Schlag. **Scalable high-quality hypergraph partitioning**. *ACM Transactions on Algorithms*

Previous Multilevel Partition Approaches

- RAPTOR [DPW15] and TB [Witt16] have been adapted to Multilevel **Hypergraph** Partitioning

  HypRAPTOR Speedup $\approx 2 - 3x$

  HypTB Speedup $\approx 0.95 - 1.3x$

$\rightarrow$ Not suffient for large networks

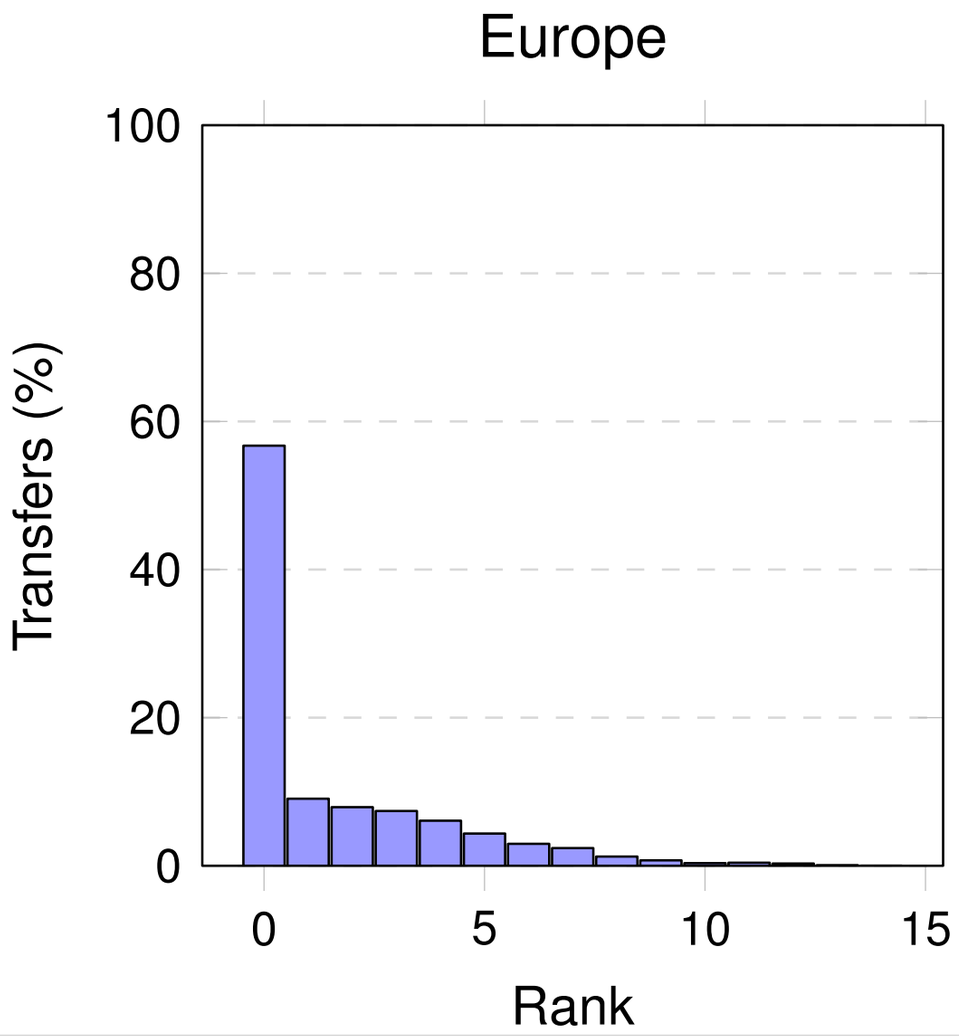
- The multilevel variant of CSA [DPSW18] (ACSA) achieves good speedups

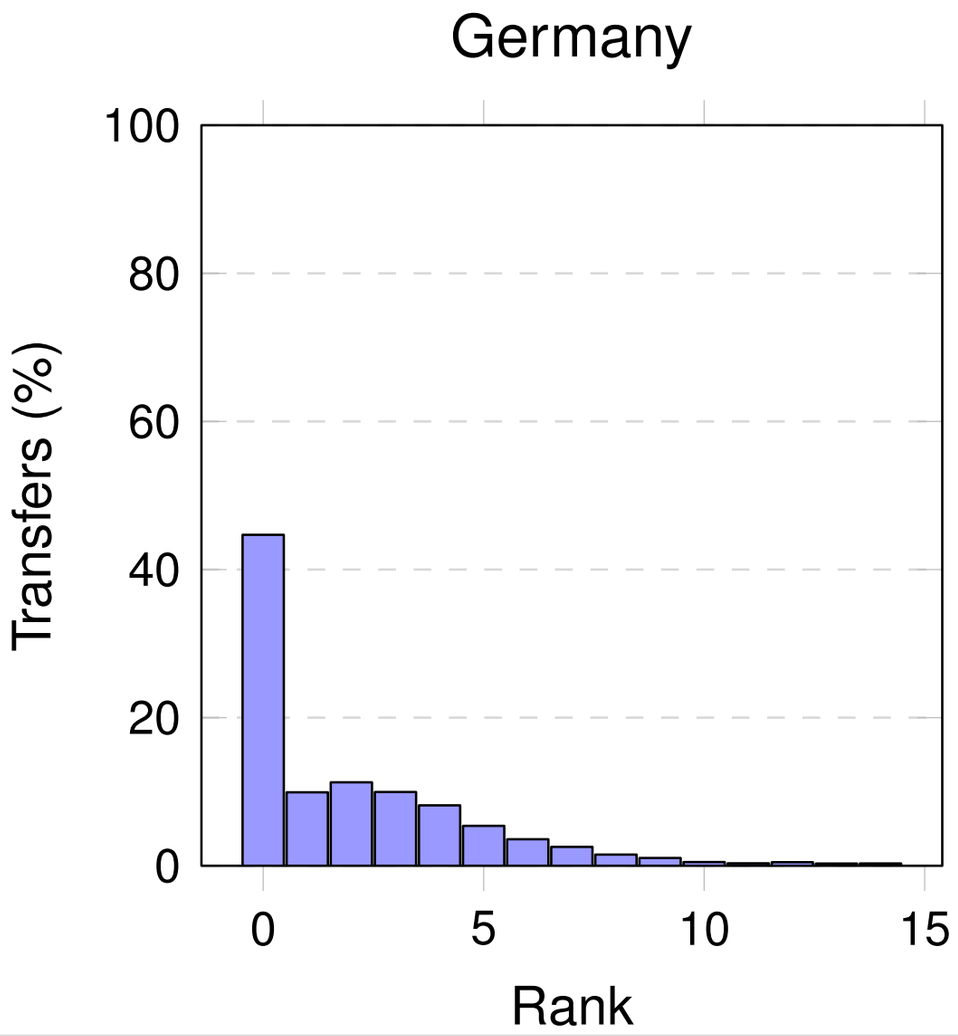
  Continental / Country $\approx 2 - 11x$

  Metropolitain $\approx 0.5 - 0.8x$

$\rightarrow$ Computes only the fastest journey

Rank Distribution





Rank Distribution