Engineering Highway Hierarchies

Peter Sanders **Dominik Schultes**

Institut für Theoretische Informatik – Algorithmik II

Universität Karlsruhe (TH)

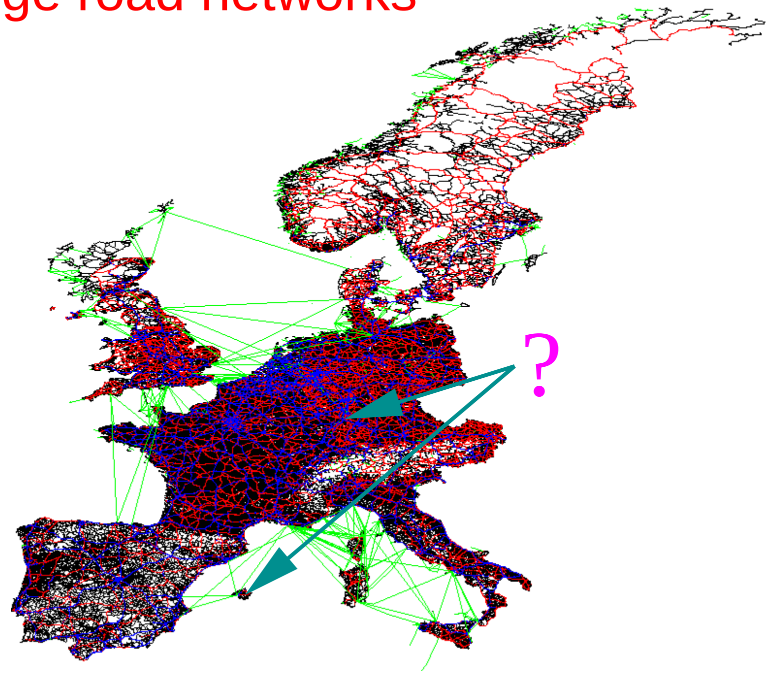
`http://algo2.iti.uka.de/schultes/hwy/`

Zürich, September 11, 2006

Route Planning

Goals:

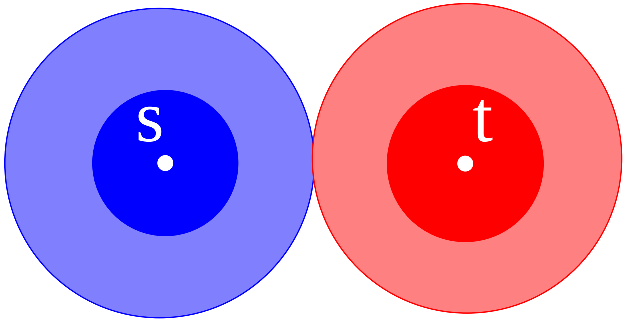
- exact shortest (i.e. fastest) paths in large road networks
- fast queries
- fast preprocessing
- low space consumption



Applications:

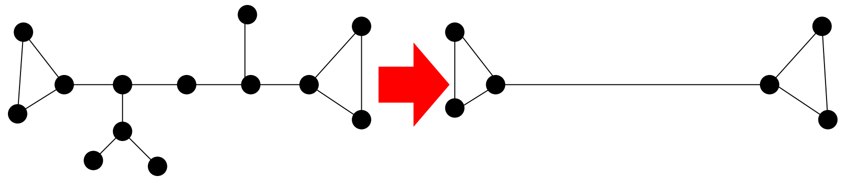
- route planning systems in the internet
- car navigation systems
- . . .

Our Approach: Highway Hierarchies[1]

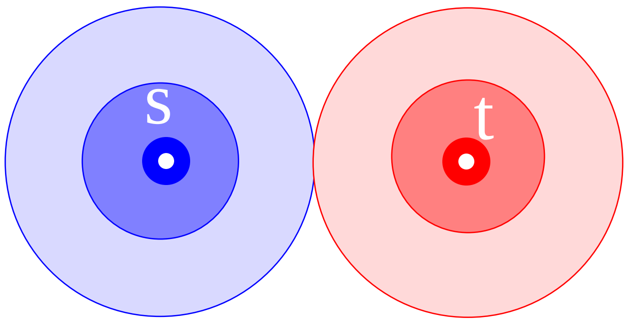


- complete search within a local area
- search in a (thinner) highway network

  = minimal graph that preserves all shortest paths



- contract network, e.g.,
- iterate ⇝ highway hierarchy



[1] presented at ESA 2005

Local Area

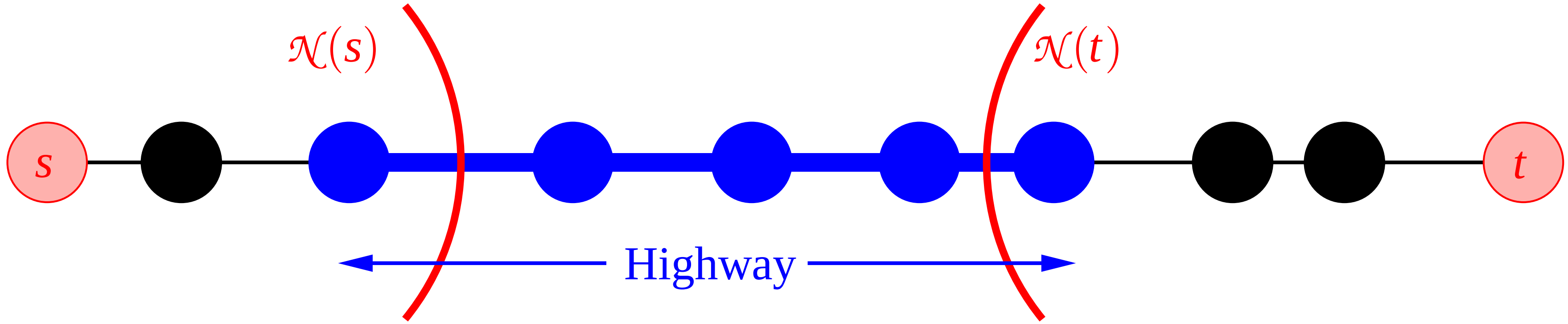
- choose neighbourhood radius $r(s)$

  (by a heuristic)

- define neighbourhood of s

$$\mathcal{N}(s) := \{v \in V \mid d(s,v) \leq r(s)\}$$

Highway Network



Edge (u,v) belongs to highway network *iff* there are nodes s and t s.t.

- (u,v) is on the "*canonical*" shortest path from s to t

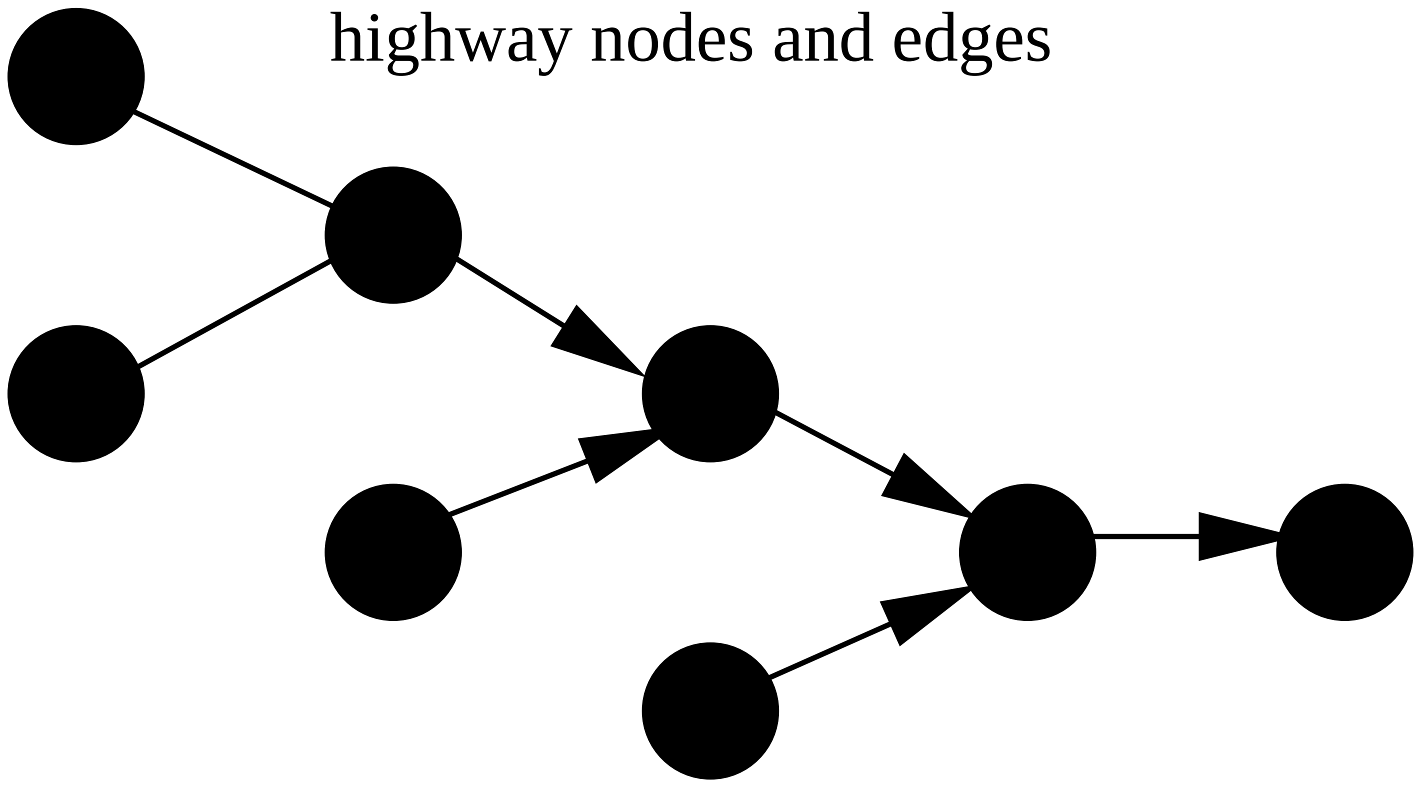
and

- (u,v) is not entirely within $\mathcal{N}(s)$ or $\mathcal{N}(t)$

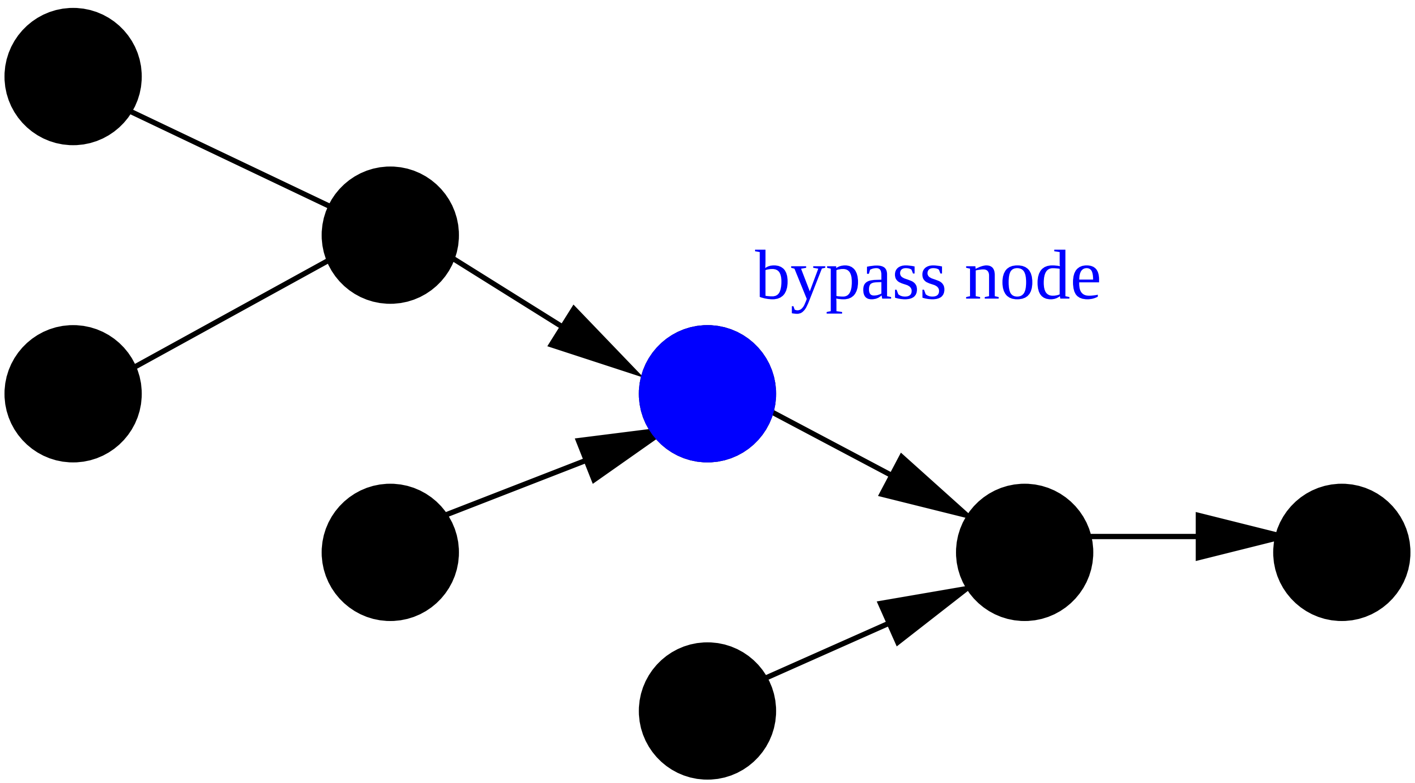
Improvements

- local area definition more flexible
- support of directed graphs
- simpler, yet more general and more effective contraction
- simpler query algorithm
- faster preprocessing, faster queries, less memory usage
- per-instance worst case performance guarantees

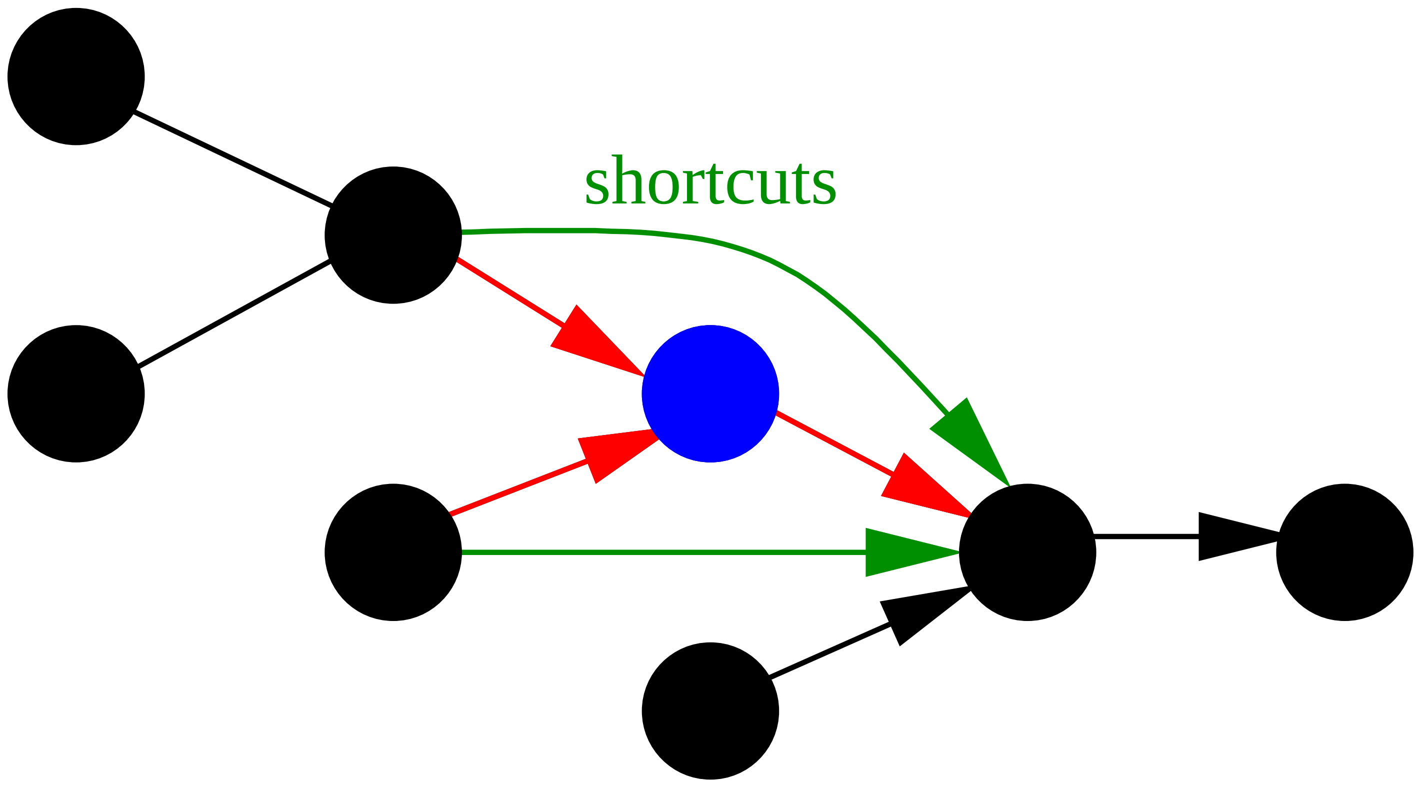
Contraction



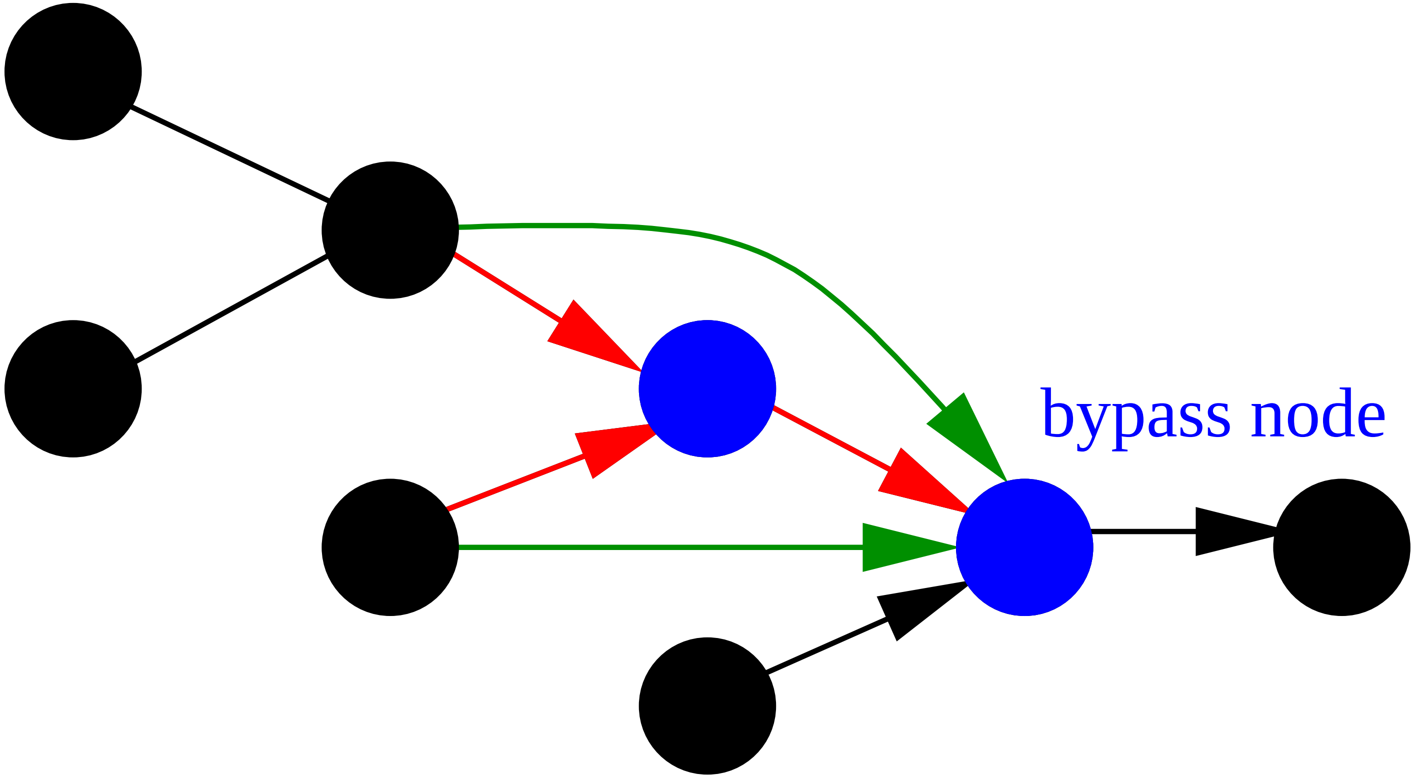
Contraction



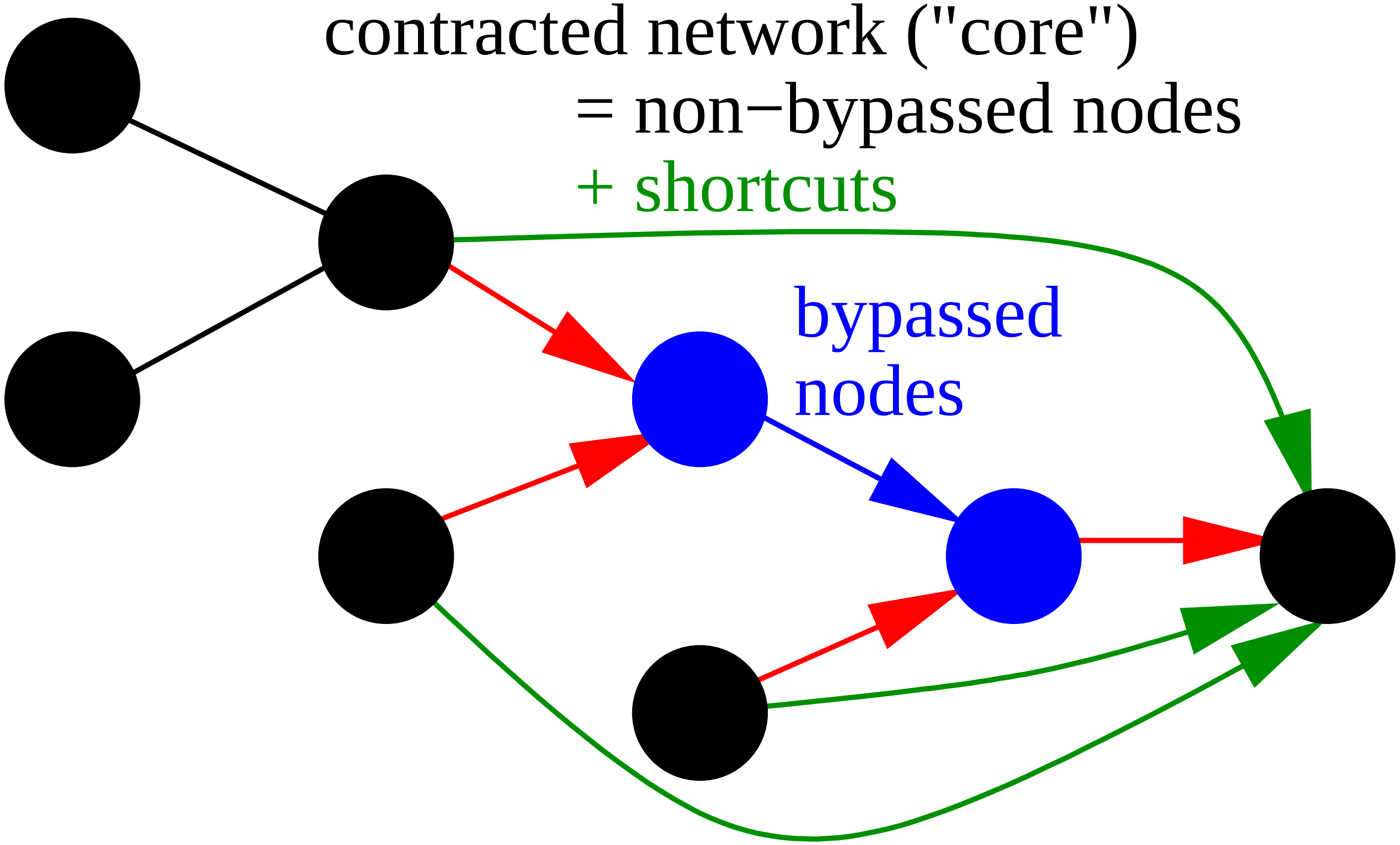
Contraction



Contraction



Contraction



Contraction

Which nodes should be bypassed?

Use some heuristic taking into account

- the number of shortcuts that would be created and
- the degree of the node.

Query

Bidirectional version of DIJKSTRA's algorithm

+ restrictions on the edges that are relaxed

+ a very simple abort criterion

$\searrow$ search space size increases[2] by $\approx$ 50%

running time decreases[2] by $\approx$ 50%

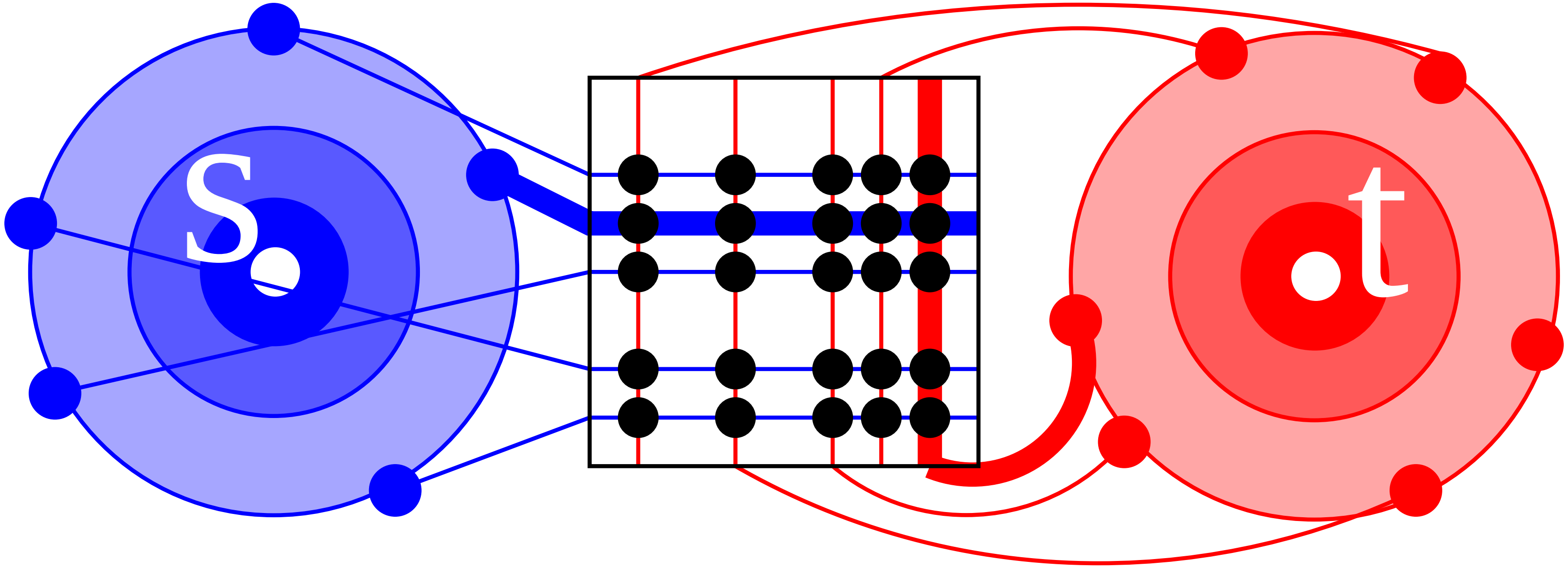
---

[2] compared to ESA 2005

Distance Table: Construction

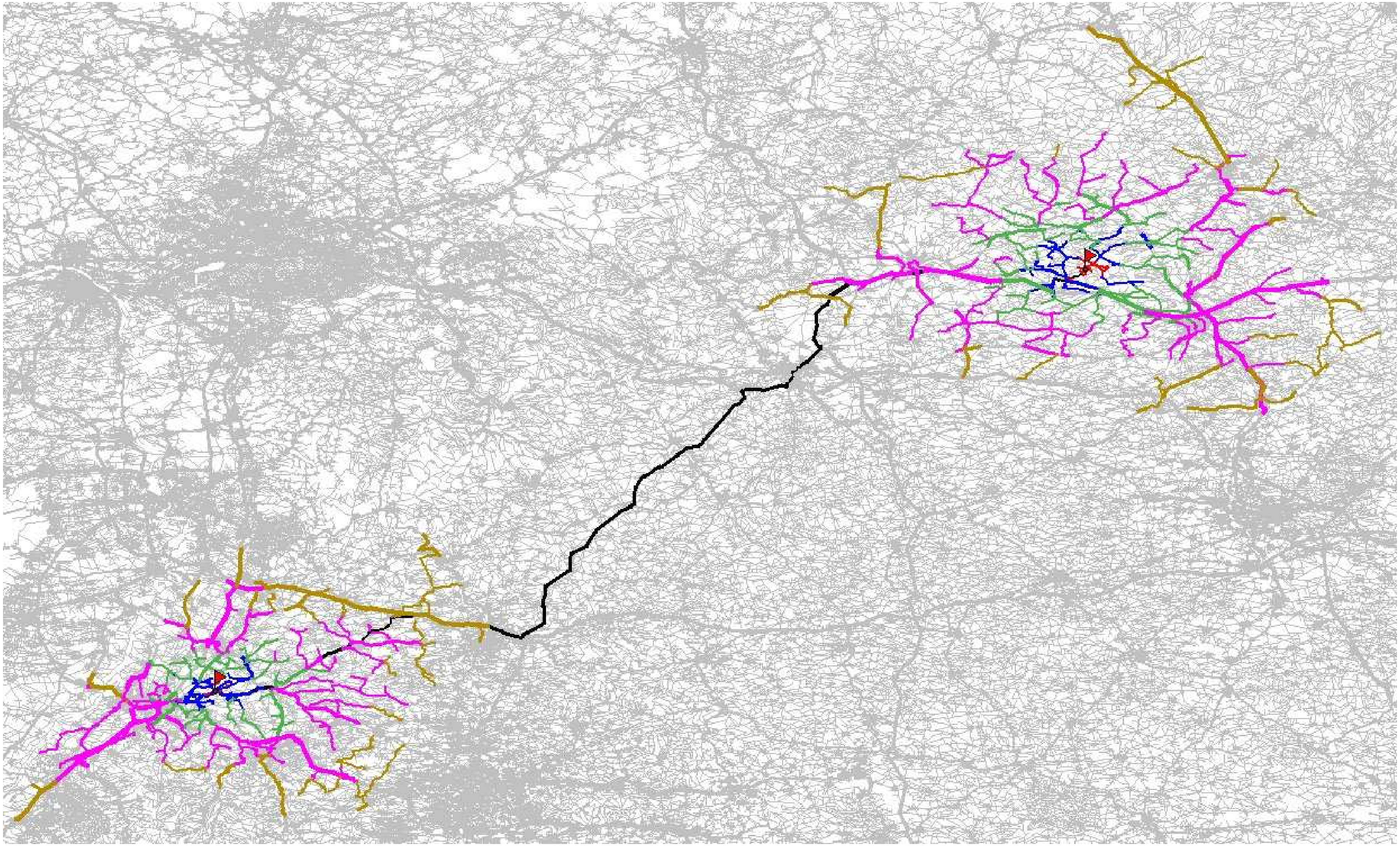
- Construct fewer levels. e.g. 4 instead of 9
- Compute an all-pairs distance table
  for the core of the topmost level L. 13 465 $\times$ 13 465 entries

Distance Table: Query



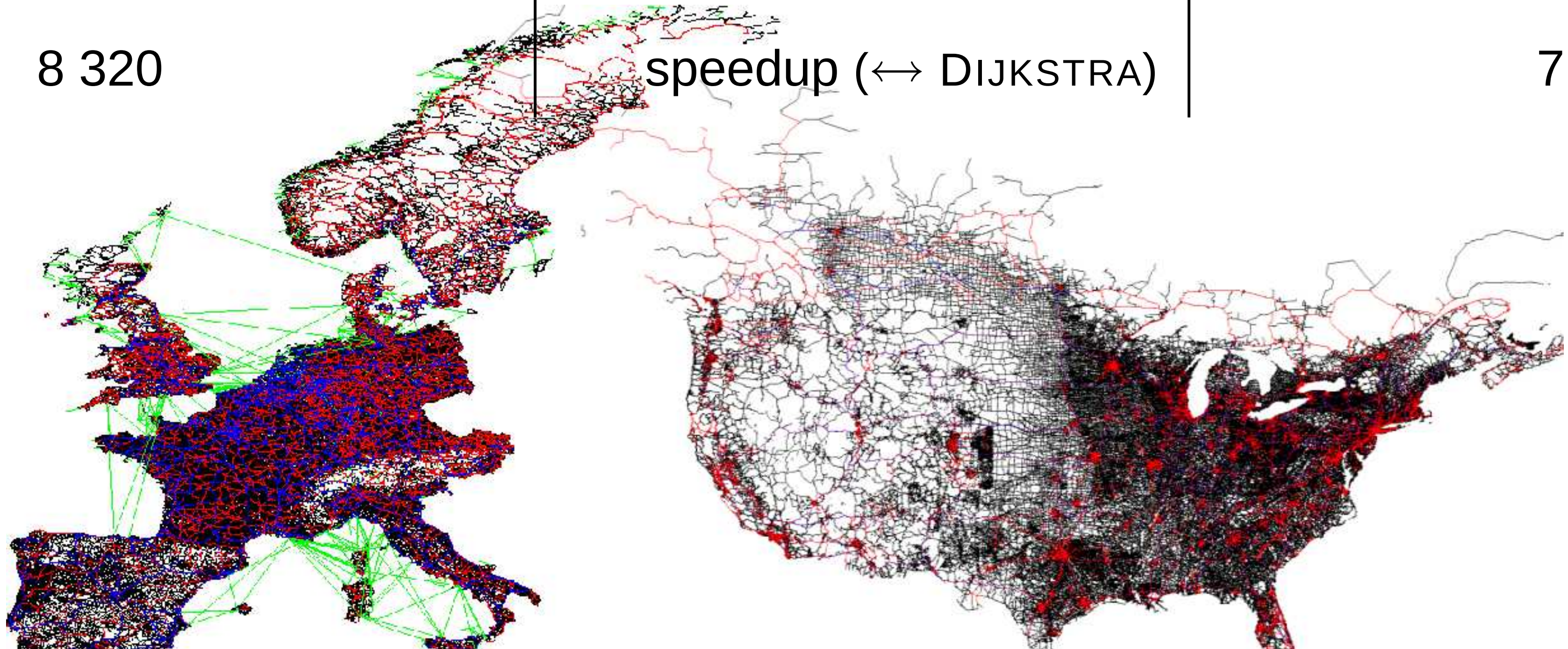
- Abort the search when all entrance points in the core of level L have been encountered. $\approx$ 55 for each direction
- Use the distance table to bridge the gap. $\approx 55 \times 55$ entries

Distance Table: Search Space Example



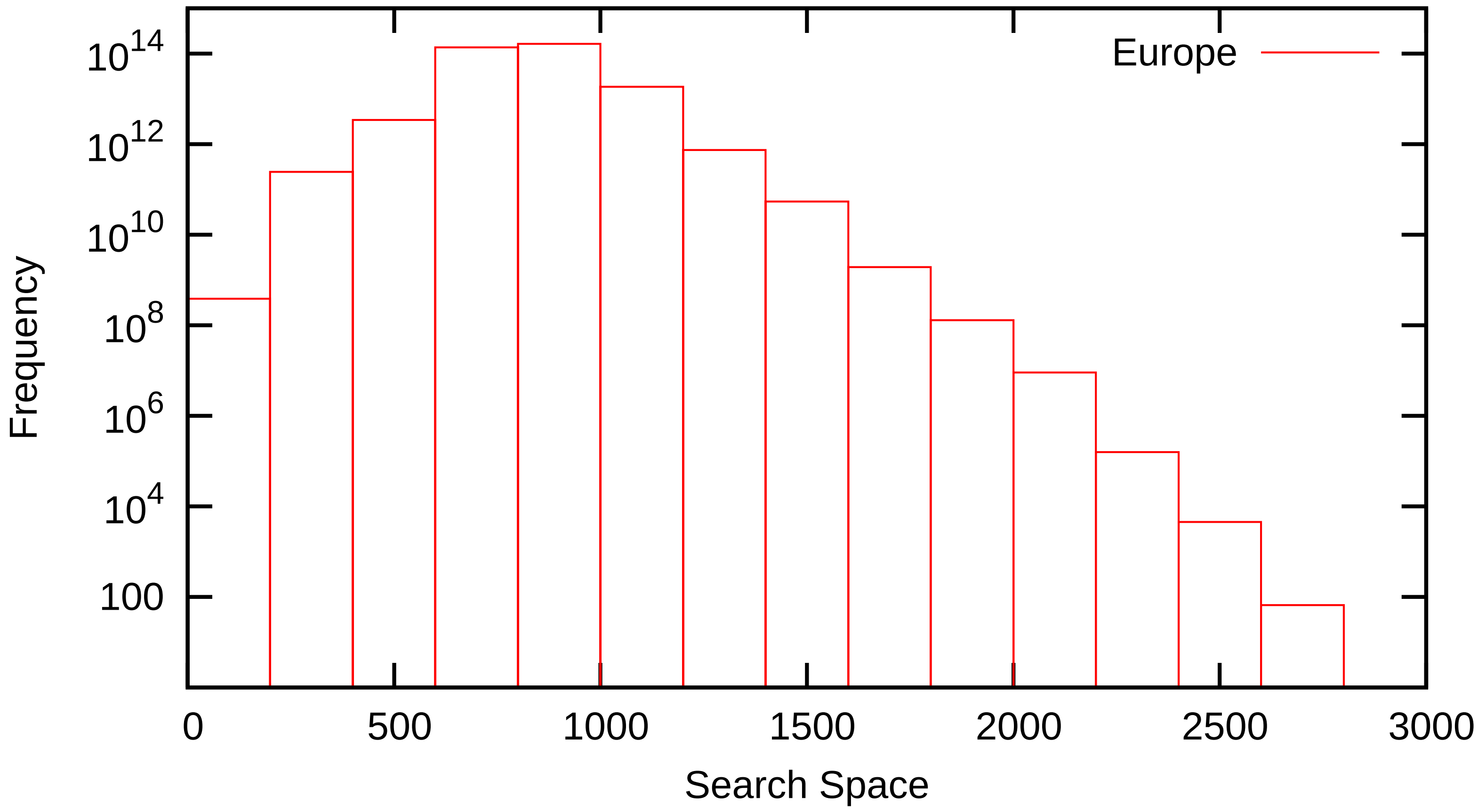
Experiments

W. Europe (PTV)			USA/CAN (PTV)
18 029 721		#nodes	18 741 705
42 199 587		#directed edges	47 244 849
15	[161][3]	construction [min]	20
0.76	[7.38][3]	search time [ms]	0.90
8 320		speedup (↔ DIJKSTRA)	7 232



[3] old values from ESA 2005

Worst Case Costs[4]



Worst Case for Europe: 2 737 settled nodes (< 0.016% of all nodes)

---

[4] using a new feature that limits the maximum shortcut length

Summary

- exact routes in large road networks (directed!) — 18 million nodes
- fast search — 0.76 ms
  - $\rightsquigarrow$ cheap, energy efficient processors in mobile devices
  - $\rightsquigarrow$ low server load
  - $\rightsquigarrow$ lots of room for additional functionality
- per-instance worst case guarantees — search space $\leq$ 2 737
- fast preprocessing — 15 min
- low space consumption — 17–68 bytes/node

Work in Progress

- combination with a goal directed approach (landmarks)

  joint work with [D. Delling, D. Wagner][5]

- computation of $M \times N$ distance tables

  (e.g. $10\,000 \times 10\,000$ table in one minute)

  joint work with [S. Knopp, F. Schulz, D. Wagner][5,6]

- storing all entrance points into the core of the topmost level

  $\rightsquigarrow$ *very* fast queries ($< 20\,\mu$s)

  joint work with [H. Bast, S. Funke, D. Matijevic][7]

---

[5] Universität Karlsruhe, Algorithmik I

[6] PTV AG, Karlsruhe

[7] Max-Planck-Institut für Informatik, Saarbrücken

Future Work

- fast, local updates on the highway network

  (e.g. for traffic jams)

- implementation for mobile devices

  (flash access, . . .)

- multi-criteria shortest paths

  joint work with [M. Müller-Hannemann, M. Schnee][8]

- . . .

---

[8]Technische Universität Darmstadt