Algorithm Engineering for Large Graphs

Fast Route Planning

Veit Batz, **Robert Geisberger**, Dennis Luxen,

Peter Sanders, Christian Vetter

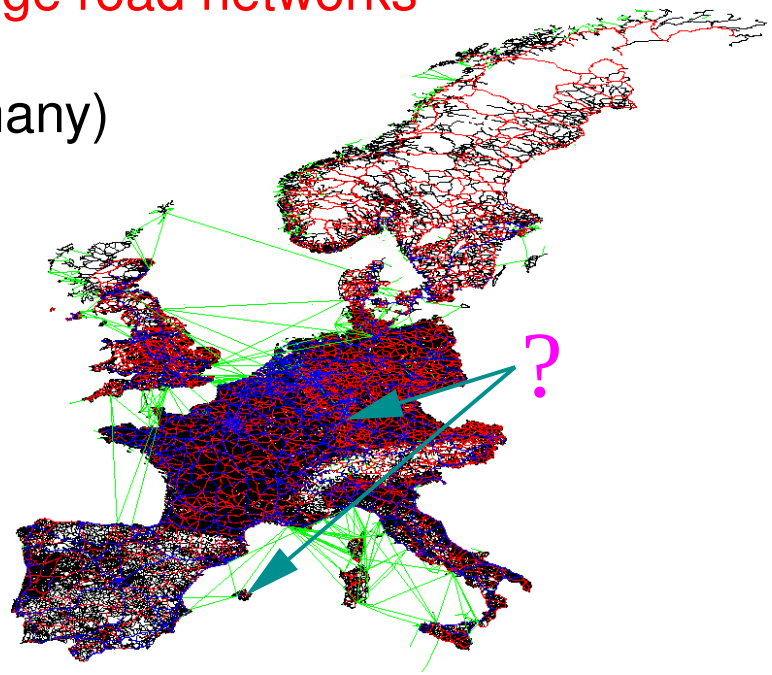
Universität Karlsruhe (TH)

Zuerich, September 2, 2008

Route Planning

Goals:

- exact shortest (i.e. fastest) paths in large road networks
- fast queries (point-to-point, many-to-many)
- fast preprocessing
- low space consumption
- fast update operations



Applications:

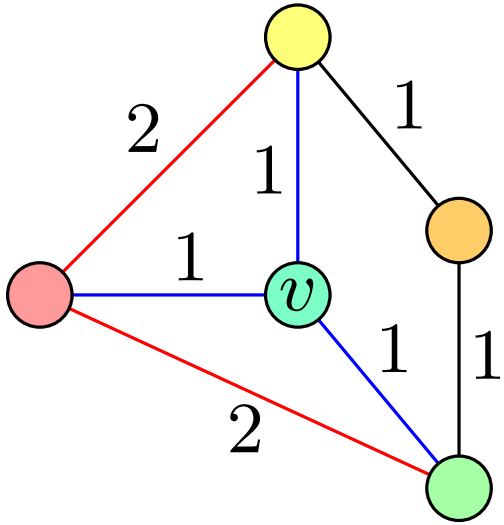
- route planning systems in the internet, car navigation systems,
- ride sharing, traffic simulation, logistics optimisation

Contraction Hierarchies

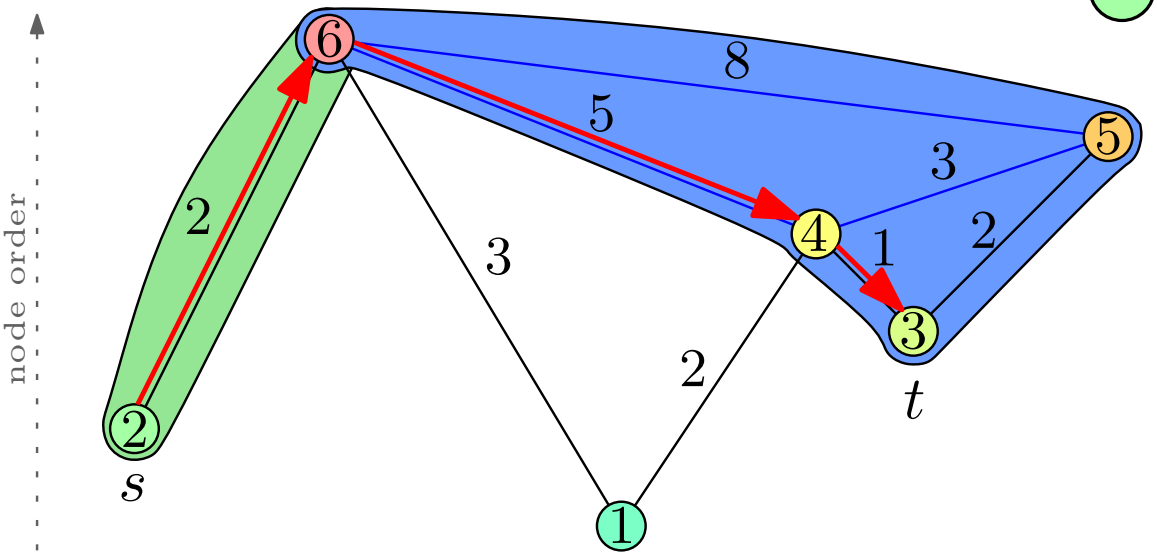
[WEA 08]

- order nodes by "importance", $V = \{1, 2, \ldots, n\}$
- contract nodes in this order, node v is contracted by

foreach *pair* (u, v) *and* (v, w) *of edges* **do**
  if $\langle u, v, w \rangle$ *is a unique shortest path* **then**
    add shortcut (u, w) with weight $w(\langle u, v, w \rangle)$

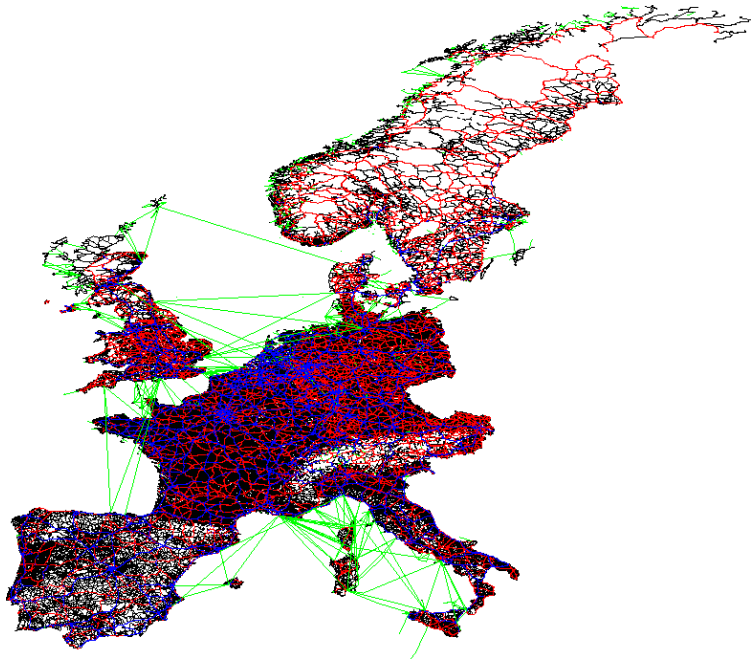


- query relaxes only edges
  to more "important" nodes
  $\Rightarrow$ valid due to shortcuts



Contraction Hierarchies

- foundation for our other methods
- conceptually very simple
- handles dynamic scenarios



Static scenario:

- 7.5 min preprocessing
- 0.21 ms to determine the path length
- 0.56 ms to determine a complete path description
- little space consumption (23 bytes/node)

Transit-Node Routing

[DIMACS Challenge 06, ALENEX 07, Science 07]

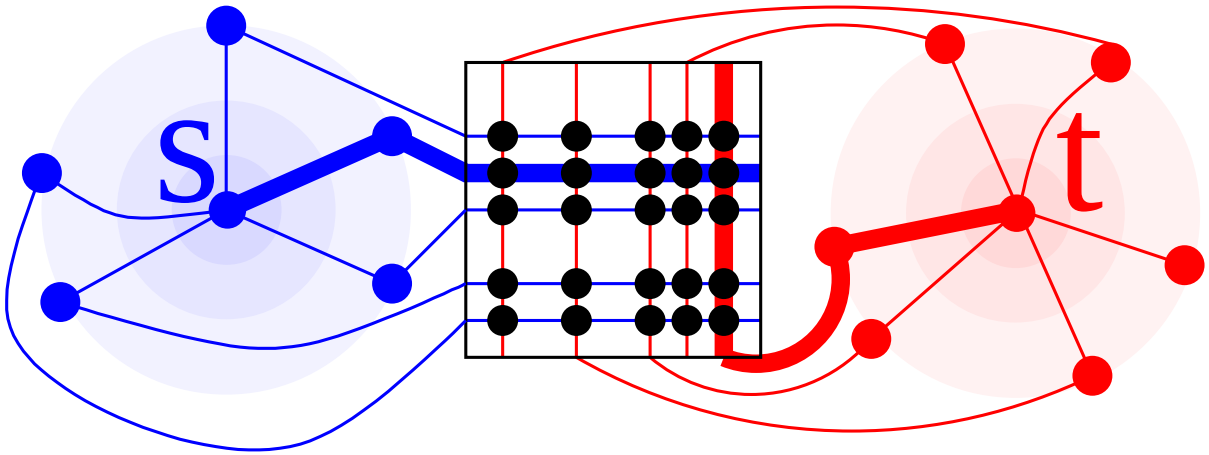
joint work with H. Bast, S. Funke, D. Matijevic

- *very* fast queries

  (down to 1.7 μs, 3 000 000 times faster than DIJKSTRA)

- winner of the 9th DIMACS Implementation Challenge

- more preprocessing time (2:37 h) and space (263 bytes/node) needed



SciAm50 Award

Mobile Contraction Hierarchies

[ESA 08]

- preprocess data on a personal computer
- highly compressed blocked graph representation — 8 bytes/node
- compact route reconstruction data structure — + 8 bytes/node

experiments on a Nokia N800 at 400 MHz

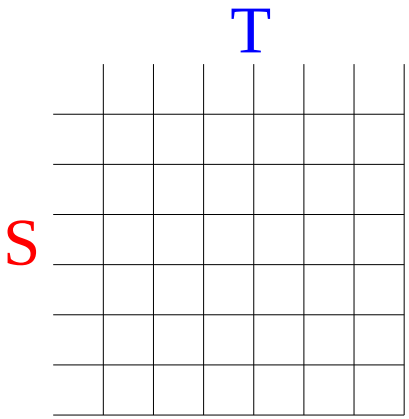
- cold query with empty block cache — 56 ms
- compute complete path — 73 ms
- recomputation, e.g. if driver took the wrong exit — 14 ms

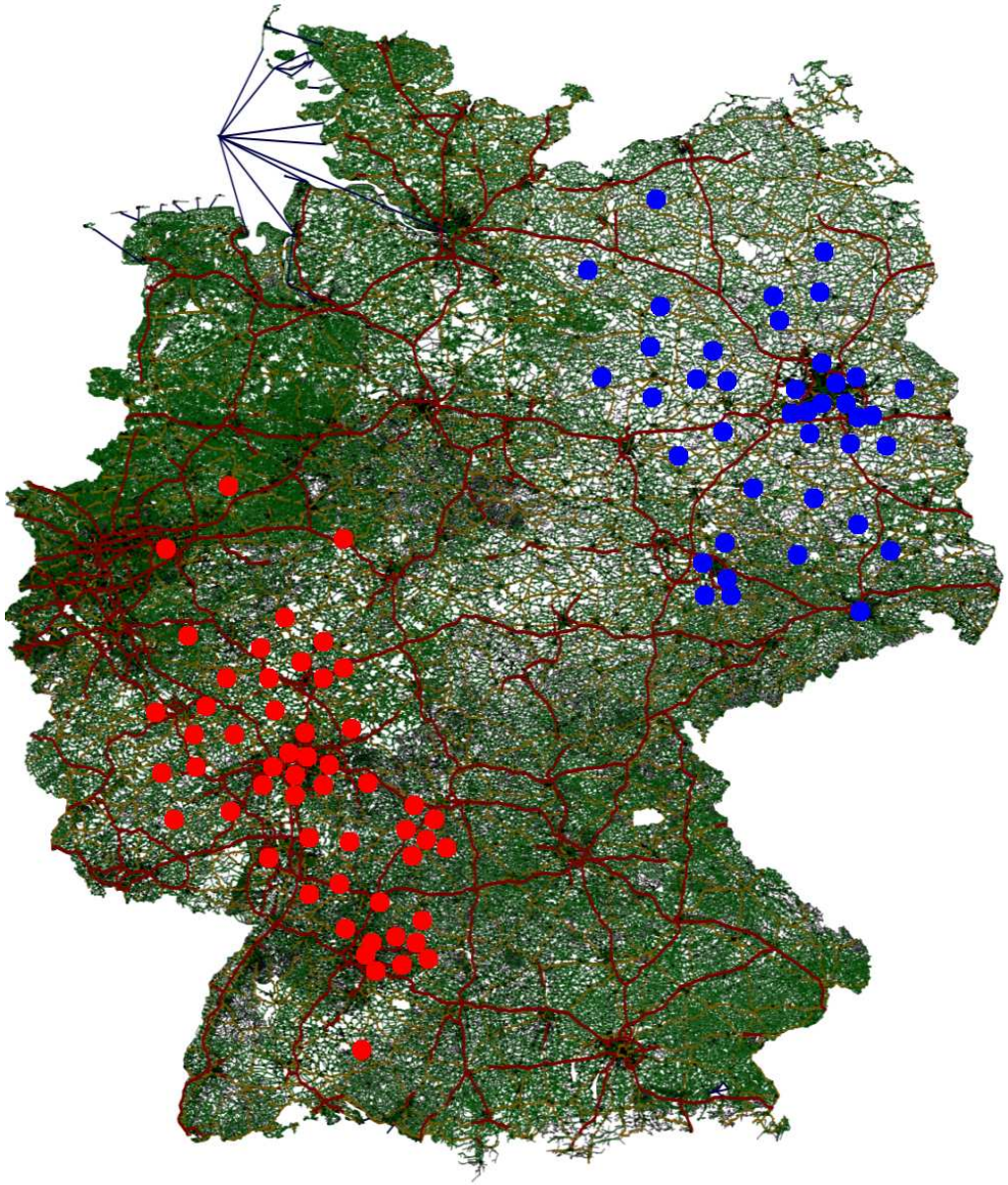
Many-to-Many Shortest Paths

joint work with S. Knopp, F. Schulz, D. Wagner

[ALENEX 07]

- efficient many-to-many variant of hierarchical bidirectional algorithms
- 10 000 × 10 000 table in 10s





Ride Sharing

Current approaches:

- match only ride offers with identical start/destination (perfect fit)
- sometimes radial search around start/destination

Our approach:

- driver picks passenger up and gives him a ride to his destination
- find the driver with the minimal detour (reasonable fit)

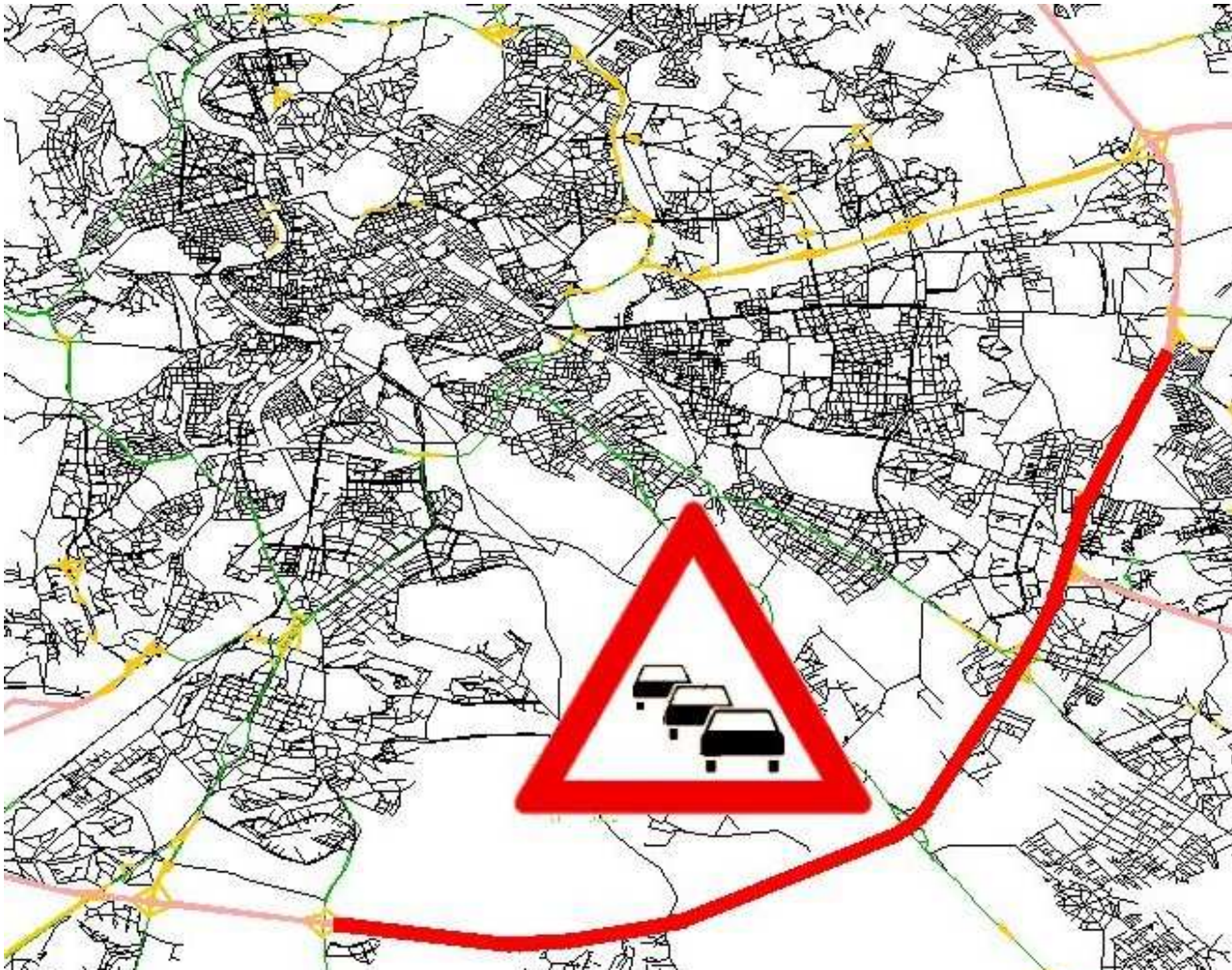
Efficient algorithm:

- adaption of the many-to-many algorithm

Dynamic Scenarios

- change entire cost function
  (e.g., use different speed profile)

- change a few edge weights
  (e.g., due to a traffic jam)



Summary

static routing in road networks is easy

$\rightsquigarrow$ applications that require massive amount or routing

$\rightsquigarrow$ instantaneous mobile routing

$\rightsquigarrow$ techniques for advanced models

$\rightsquigarrow$ updating a few edge weights is OK

Current / Future Work

- Time-dependent edge weights

  challenge: backward search impossible (?)
- Multiple objective functions and restrictions (bridge height,. . .)
- Multicriteria optimization (cost, time,. . .)
- Integrate individual and public transportation
- Other objectives for time-dependent travel
- Routing driven traffic simulation