Route Planning in Road Networks

Dominik Schultes

Institut für Theoretische Informatik – Algorithmik II

Universität Karlsruhe (TH)

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

Karlsruhe, 7. Februar 2008

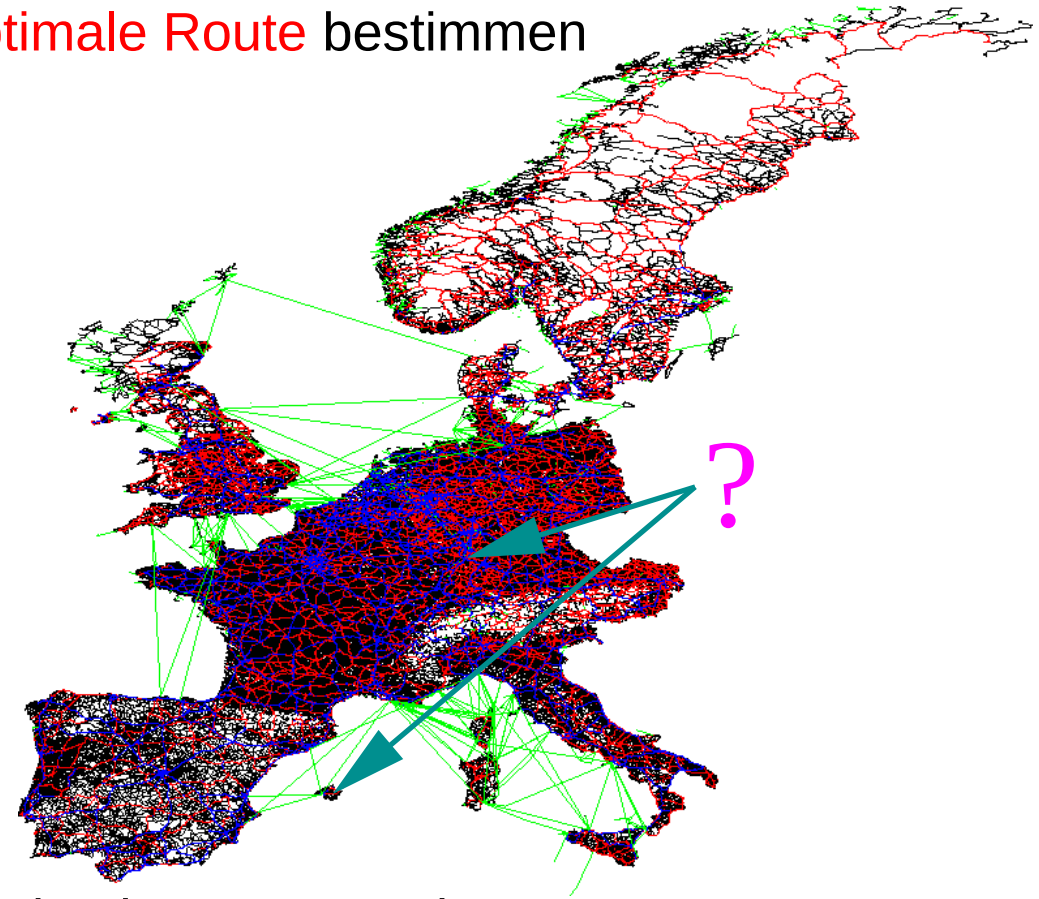
Routenplanung

Problemstellung:

In gegebenem Straßennetzwerk eine optimale Route bestimmen

von einem gegeben Startpunkt

zu einem gegebenen Zielpunkt



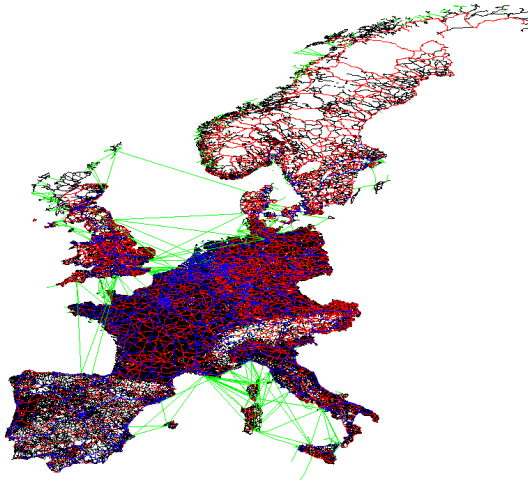
Anwendungen:

- Online-Routenplanungssysteme, Navigationssysteme im Auto
- Vekehrssimulationen, Logistik-Optimierungen

Dijkstras Algorithmus

die klassische Lösung [1959]

$O(n \log n + m)$ (mit Fibonacci Heaps)



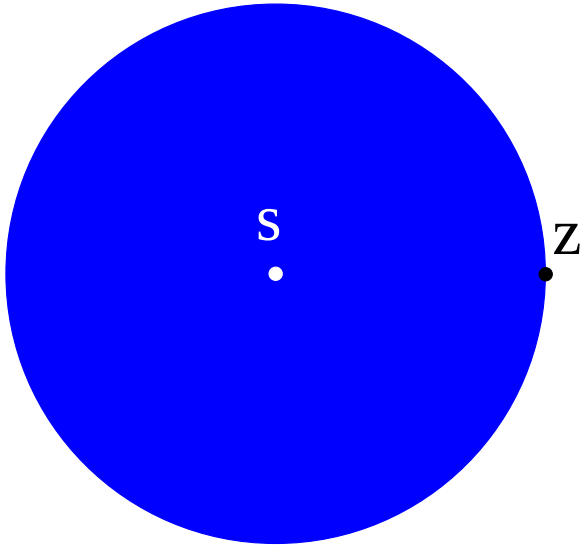
nicht praktikabel

für große Graphen

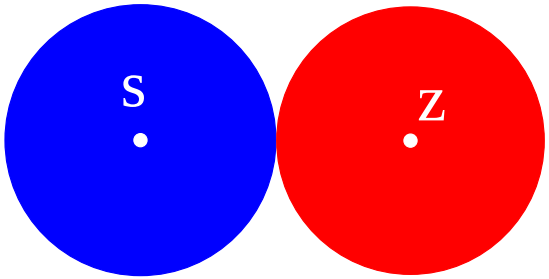
(z.B. Europäisches Straßennetz:

$\approx$ 18 000 000 Knotenpunkte)

Dijkstra



bidirektionaler Dijkstra



verbessert die Laufzeit,

aber immer noch zu langsam

Beschleunigungstechniken

$\rightsquigarrow$ allgemeine Lösung langsam — Dijkstra: $\Omega(n+m)$

aber:

für spezielle Fälle gibt es noch **Hoffnung** — z.B. bei Straßennetzen

- Zusatzdaten — z.B. Knotenkoordinaten
- Vorberechnungen $\rightsquigarrow$ Hilfsdaten — z.B. ‘Wegweiser’
- spezielle Eigenschaften des Graphen — z.B. planar, hierarchisch

Ziele

Primärziele:

- schnelle Suchzeiten
- beweisbar optimale Routen

Sekundärziele:

- schnelle Vorberechnung / Umgang mit großen Netzwerken
- niedriger Speicherverbrauch
- schnelle Aktualisierung
- Einfachheit

Highway Hierarchien

Transit−Node Routing
sehr schnelle Suchzeiten
[DIMACS 06, ALENEX 07, Science 07]

HH Star
zielgerichtet
[DIMACS 06]

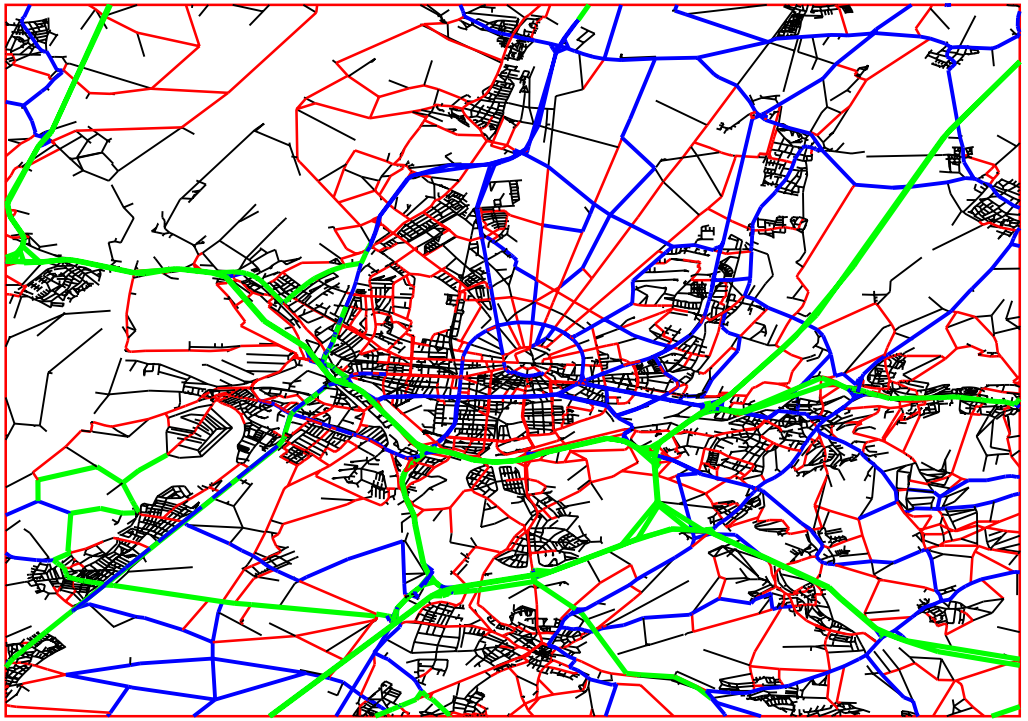
Highway Hierarchien
Grundlage
[ESA 05, ESA 06]

Hwy−Node Routing
Aktualisierungen möglich
[WEA 07]

Many−to−Many
Distanztabellen berechnen
[ALENEX 07]

Highway Hierarchien

- bestimme eine Hierarchie von Highway Netzwerken bzw.
- ordne Kanten nach ‘Wichtigkeit’



bidirektionaler Suchalgorithmus:

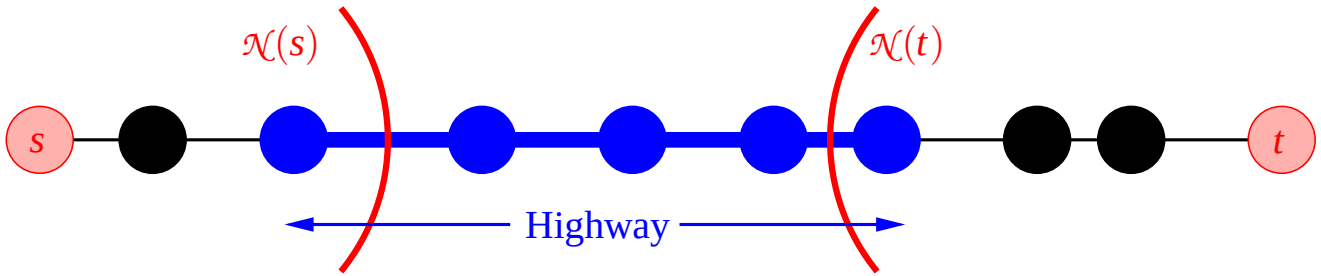
mit zunehmender Entfernung von Start/Ziel:

betrachte nur noch ‘wichtigere’ Kanten

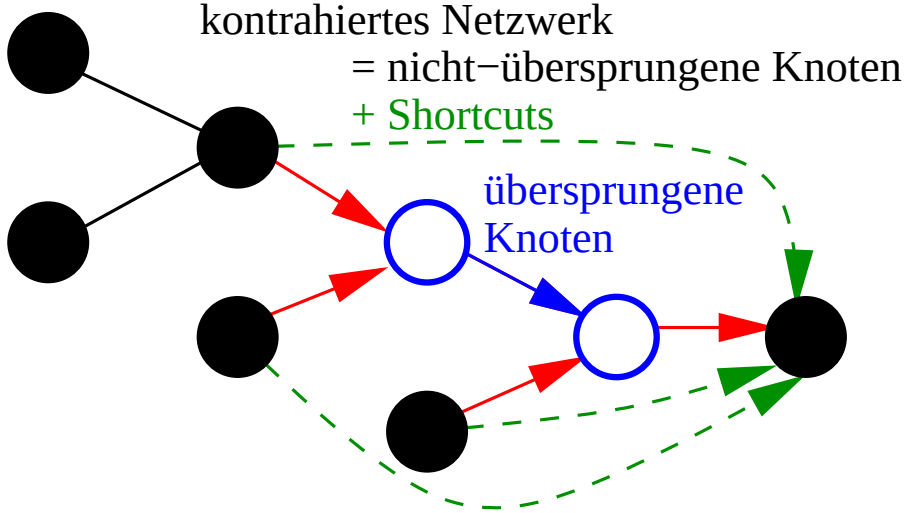
Highway Hierarchien

Konstruktion: abwechselnd

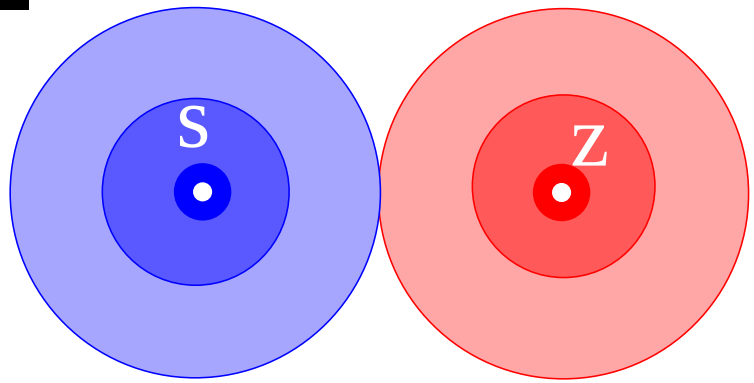
- Kantenreduktion: entferne Kanten, die ausschließlich in der Nähe von Start/Ziel gebraucht werden



- Knotenreduktion (‘Kontraktion’): überspringe Knoten mit niedrigem Grad



Highway Hierarchien



- Grundlage für andere Verfahren
- direkt anwendbar für Punkt-zu-Punkt Anfragen
- 13 min Vorberechnungszeit
- 0.61 ms um optimale Pfadlänge zu bestimmen
- (0.80 ms um kompletten Pfad zu bestimmen)

} Europa ≈ 18 000 000 Knoten AMD Opteron 2.0 GHz

- akzeptabler Speicherverbrauch (48 Byte/Knoten)
  kann reduziert werden auf 17 Byte/Knoten

Sanders, Schultes. ESA 2006.

Highway Hierarchies Star

Transit−Node Routing
sehr schnelle Suchzeiten
[DIMACS 06, ALENEX 07, Science 07]

HH Star
zielgerichtet
[DIMACS 06]

Highway Hierarchien
Grundlage
[ESA 05, ESA 06]

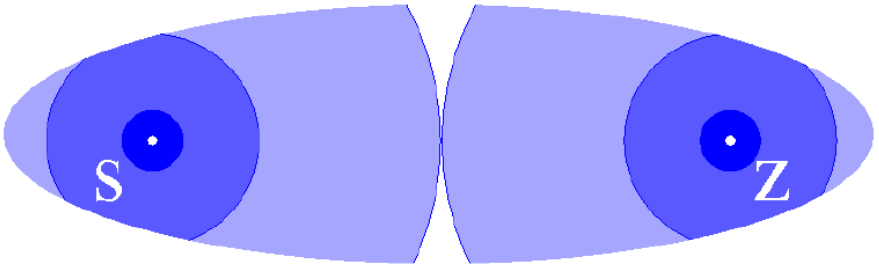
Hwy−Node Routing
Aktualisierungen möglich
[WEA 07]

Many−to−Many
Distanztabellen berechnen
[ALENEX 07]

Highway Hierarchies Star

- Kombination der Highway Hierarchien mit zielgerichteter Suche
- leichte Verbesserung (0.49 ms)
- größere Wirkung
  - für Näherungslösungen oder
  - bei Verwendung einer Distanzmetrik

    (anstatt einer Reisezeitmetrik)



Many-to-Many

HH Star
zielgerichtet
[DIMACS 06]

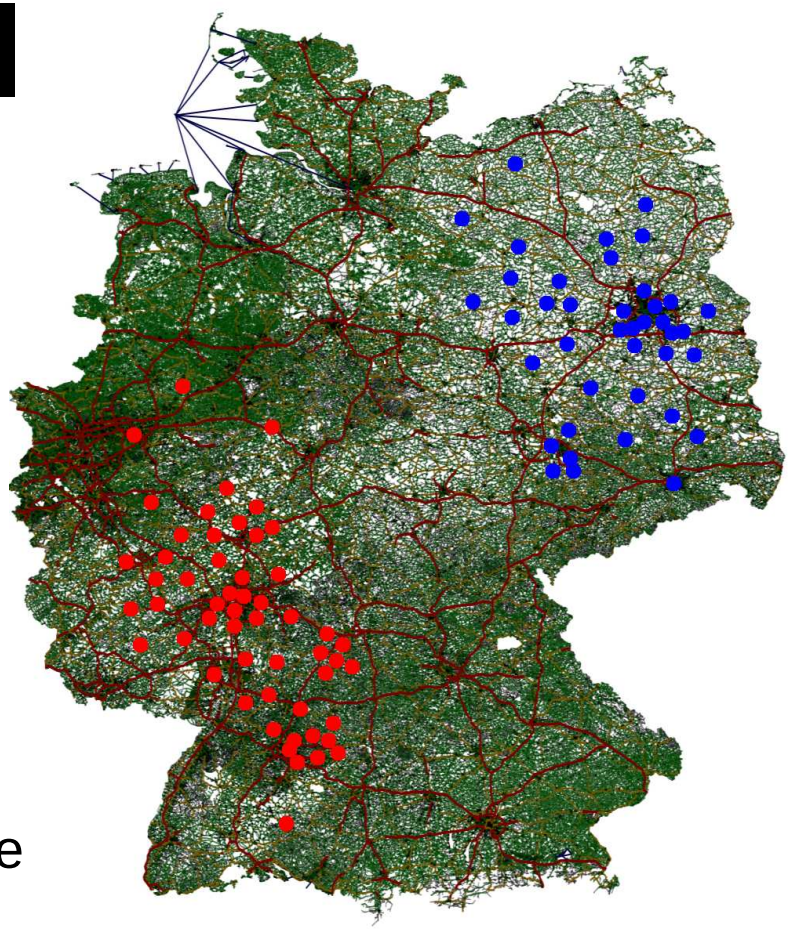
Transit−Node Routing
sehr schnelle Suchzeiten
[DIMACS 06, ALENEX 07, Science 07]

Highway Hierarchien
Grundlage
[ESA 05, ESA 06]

Hwy−Node Routing
Aktualisierungen möglich
[WEA 07]

Many−to−Many
Distanztabellen berechnen
[ALENEX 07]

Distanztabellen



Gegeben:

- Graph $G = (V, E)$
- Menge von Startknoten $S \subseteq V$
- Menge von Zielknoten $T \subseteq V$

Gesucht: $|S| \times |T|$ Distanztabelle

mit den Längen aller kürzester Wege

- z.B. 10 000 $\times$ 10 000 Tabelle in 23 Sekunden

Knopp, Sanders, Schultes, Schulz, Wagner. ALENEX 2007.

Transit-Node Routing

Transit−Node Routing
sehr schnelle Suchzeiten
[DIMACS 06, ALENEX 07, Science 07]

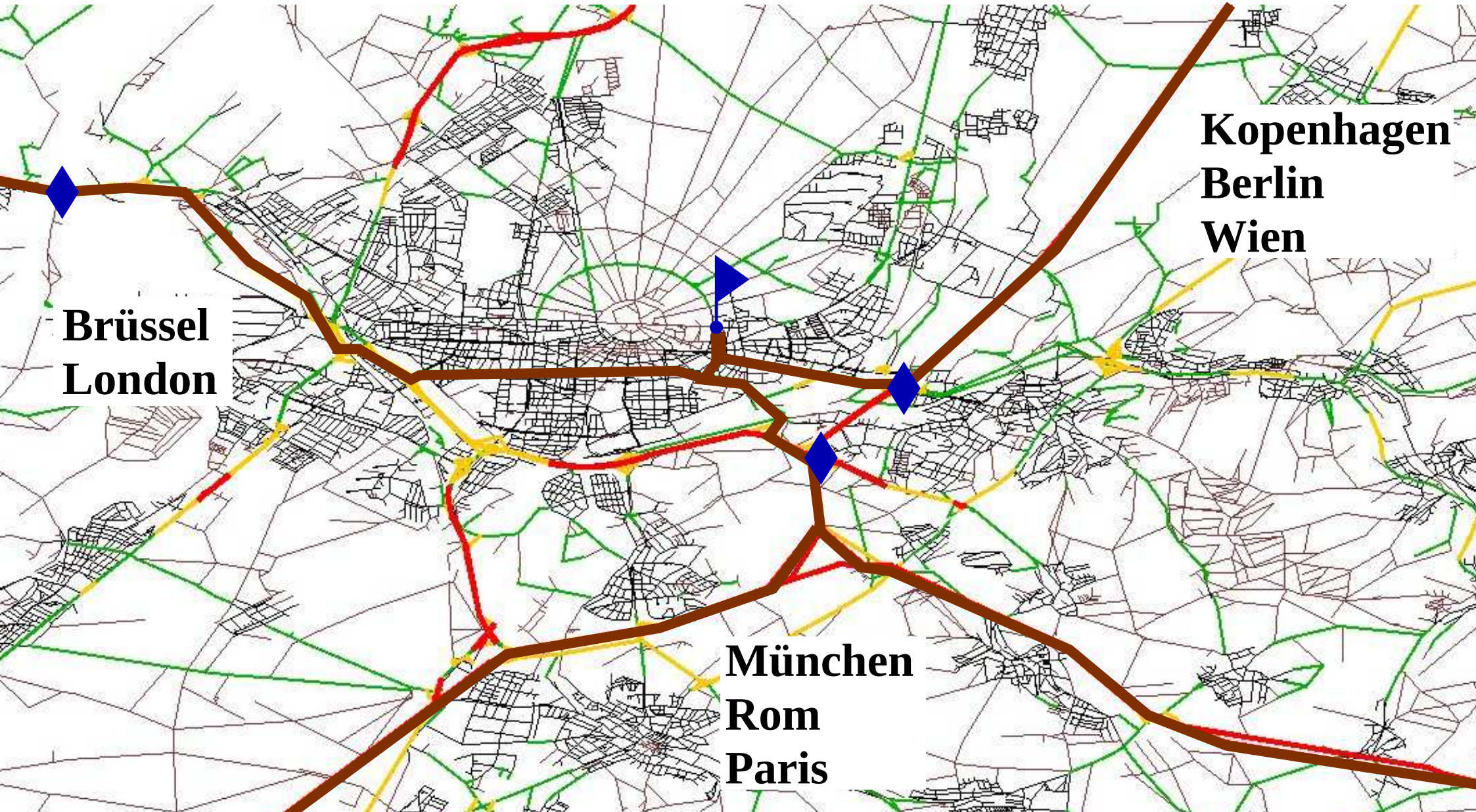
HH Star
zielgerichtet
[DIMACS 06]

Highway Hierarchien
Grundlage
[ESA 05, ESA 06]

Many−to−Many
Distanztabellen berechnen
[ALENEX 07]

Hwy−Node Routing
Aktualisierungen möglich
[WEA 07]

Motivation



Beobachtungen

1. Bei langen Fahrten verlässt man den Startpunkt

über einen von wenigen ‘wichtigen’ Verkehrsknotenpunkten

(“Zugangspunkte”)

($\rightsquigarrow$ für jeden Knoten alle Zugangspunkte speichern) [$\approx$ 10 pro Knoten]

---

2. Jeder Zugangspunkt ist für mehrere Knoten relevant. $\rightsquigarrow$

Vereinigungsmenge aller Zugangspunkte ist klein

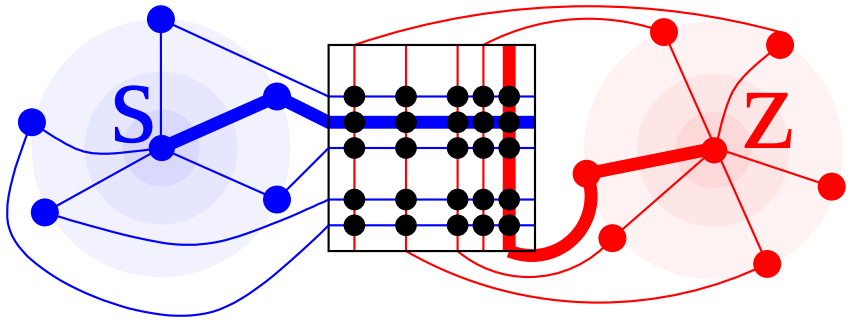
(“Transitknotenmenge”)

($\rightsquigarrow$ die Abstände zwischen allen Transitknoten speichern) [$\approx 10\,000^2$ Abstände]

Transit-Node Routing

Vorberechnung:

- bestimme Transitknotenmenge $\mathcal{T} \subseteq V$
- berechne $|\mathcal{T}| \times |\mathcal{T}|$ Distanztabelle
- für jeden Knoten: bestimme die Zugangsknoten ($A : V \rightarrow 2^{\mathcal{T}}$),
  speichere die Abstände



Suche (Start s und Ziel t gegeben): berechne

$$d_{\mathcal{T}}(s,t) := \min\{d(s,u)+d(u,v)+d(v,t) : u \in A(s), v \in A(t)\}$$

Transit-Node Routing

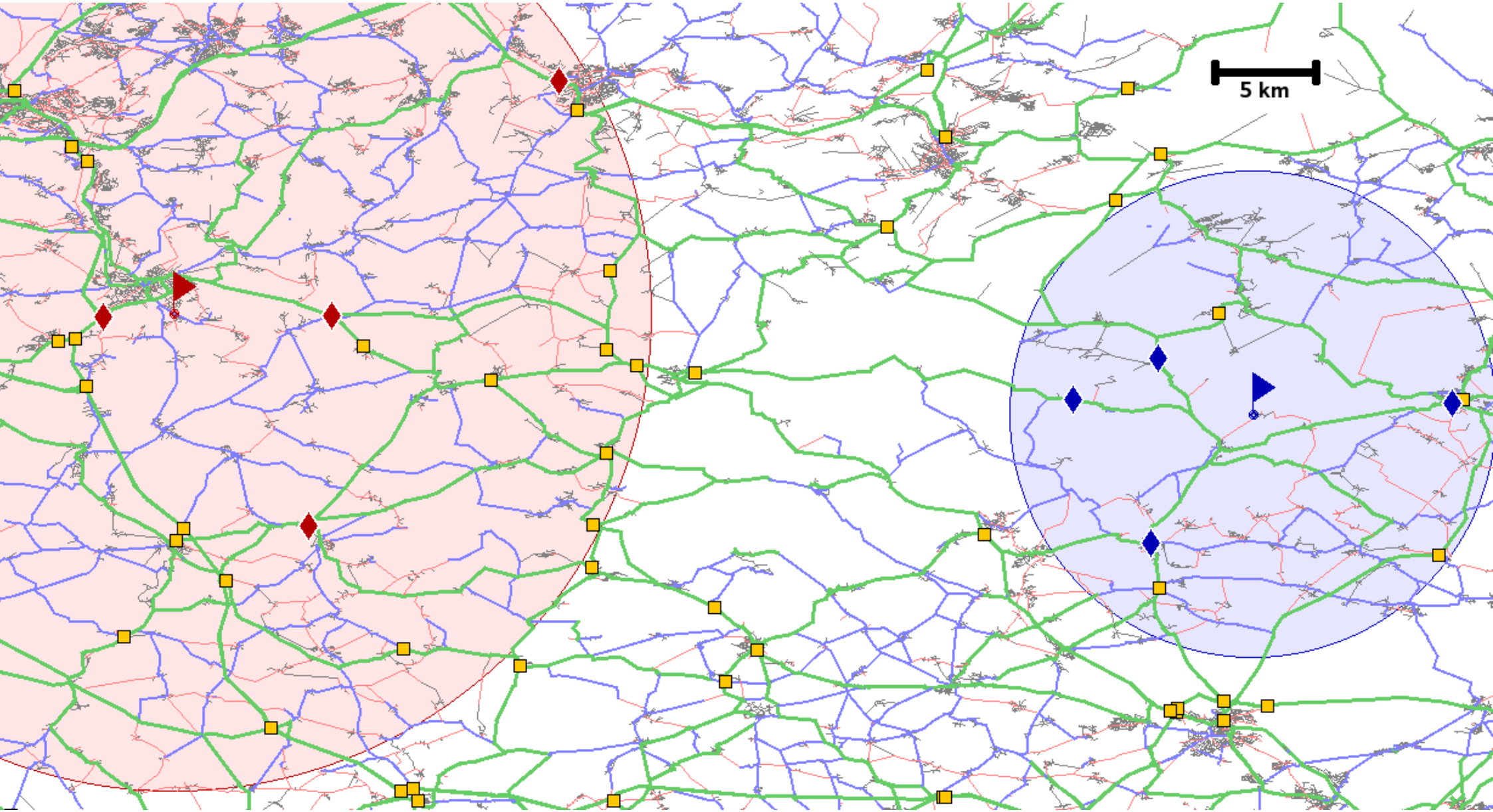
Lokalitätsfilter:

lokale Fälle müssen erkannt werden ($\rightsquigarrow$ Sonderbehandlung)

$L : V \times V \rightarrow \{\text{true}, \text{false}\}$

$\neg L(s,t)$ impliziert $d(s,t) = d_{\mathcal{T}}(s,t)$

Beispiel



Experimentelle Ergebnisse

- *extrem* schnelle Suchzeiten
  (4 μs, > 1 000 000 mal schneller als DIJKSTRA)
- größere Vorberechnungszeit (1:15 h) und
  mehr Speicher (247 Byte/Knoten) benötigt
- Sieger beim 9th DIMACS Implementation Challenge 2006
- Scientific American 50 Award 2007

Sanders, Schultes. DIMACS Challenge 2006.

Bast, Funke, Sanders, Schultes. Science, 2007.

Bast, Funke, Matijevic, Sanders, Schultes. ALENEX 2007.

c't

DIE WELT

Süddeutsche Zeitung

COMPUTER ZEITUNG

Offene Fragen

- Wie bestimmt man die Transitknoten?
- Wie kann man die Zugangspunkte effizient bestimmen?
- Wie bestimmt man einen effektiven Lokalitätsfilter?
- Wie geht man mit den lokalen Fällen um?

Offene Fragen

- Wie bestimmt man die Transitknoten?
- Wie kann man die Zugangspunkte effizient bestimmen?
- Wie bestimmt man einen effektiven Lokalitätsfilter?
- Wie geht man mit den lokalen Fällen um?

Antwort:

- Verwende andere Routenplanungstechniken!

Highway-Node Routing

Transit−Node Routing
sehr schnelle Suchzeiten
[DIMACS 06, ALENEX 07, Science 07]

HH Star
zielgerichtet
[DIMACS 06]

Highway Hierarchien
Grundlage
[ESA 05, ESA 06]

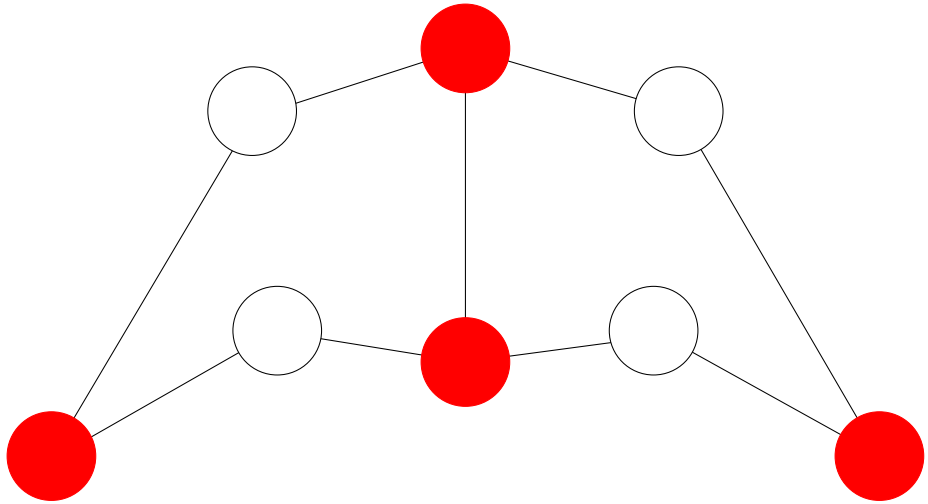
Hwy−Node Routing
Aktualisierungen möglich
[WEA 07]

Many−to−Many
Distanztabellen berechnen
[ALENEX 07]

Overlaygraph: Definition

[Holzer, Schulz, Wagner, Weihe, Zaroliagis 2000–2007]

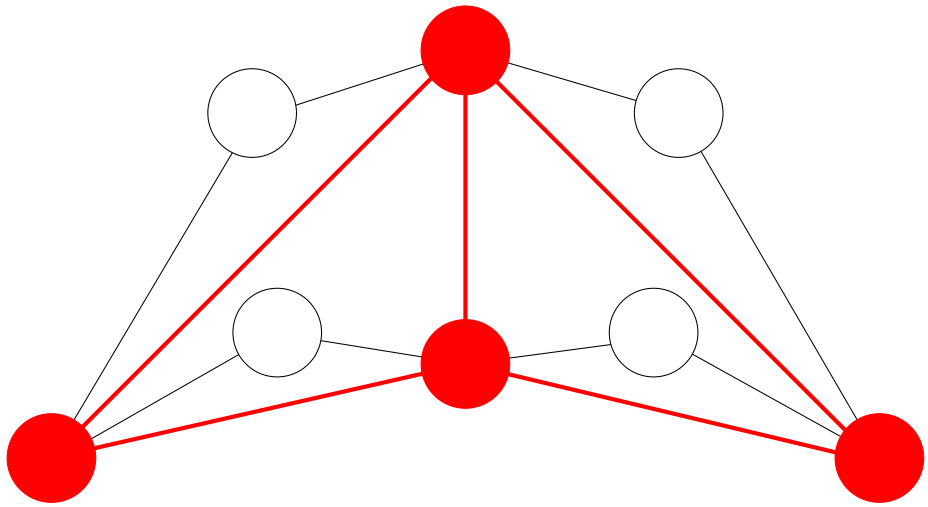
- gegeben: Graph $G = (V, E)$
- wähle Knotenteilmenge $S \subseteq V$



Overlaygraph: Definition

[Holzer, Schulz, Wagner, Weihe, Zaroliagis 2000–2007]

- gegeben: Graph $G = (V, E)$
- wähle Knotenteilmenge $S \subseteq V$

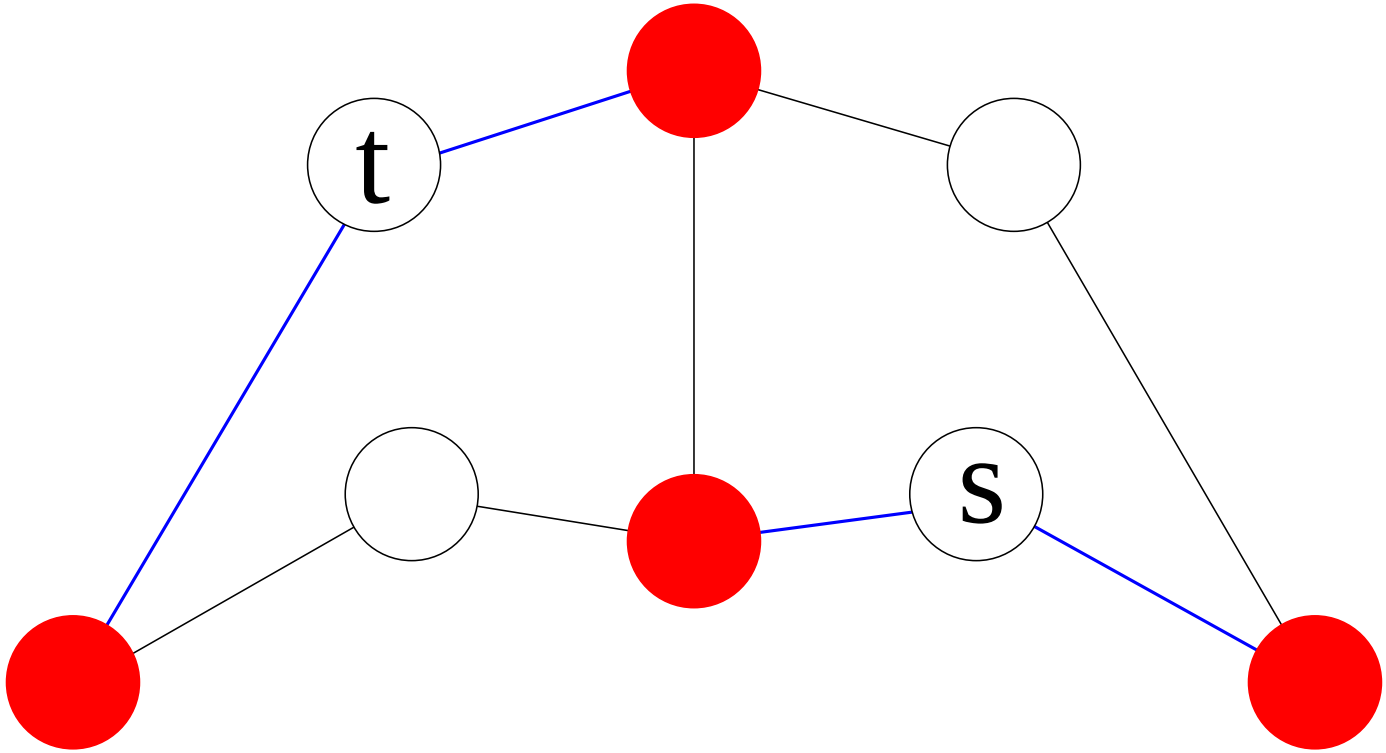


- Overlaygraph $G' := (S, E')$

  wähle Kantenmenge E', die kürzeste-Wege-Distanzen erhält

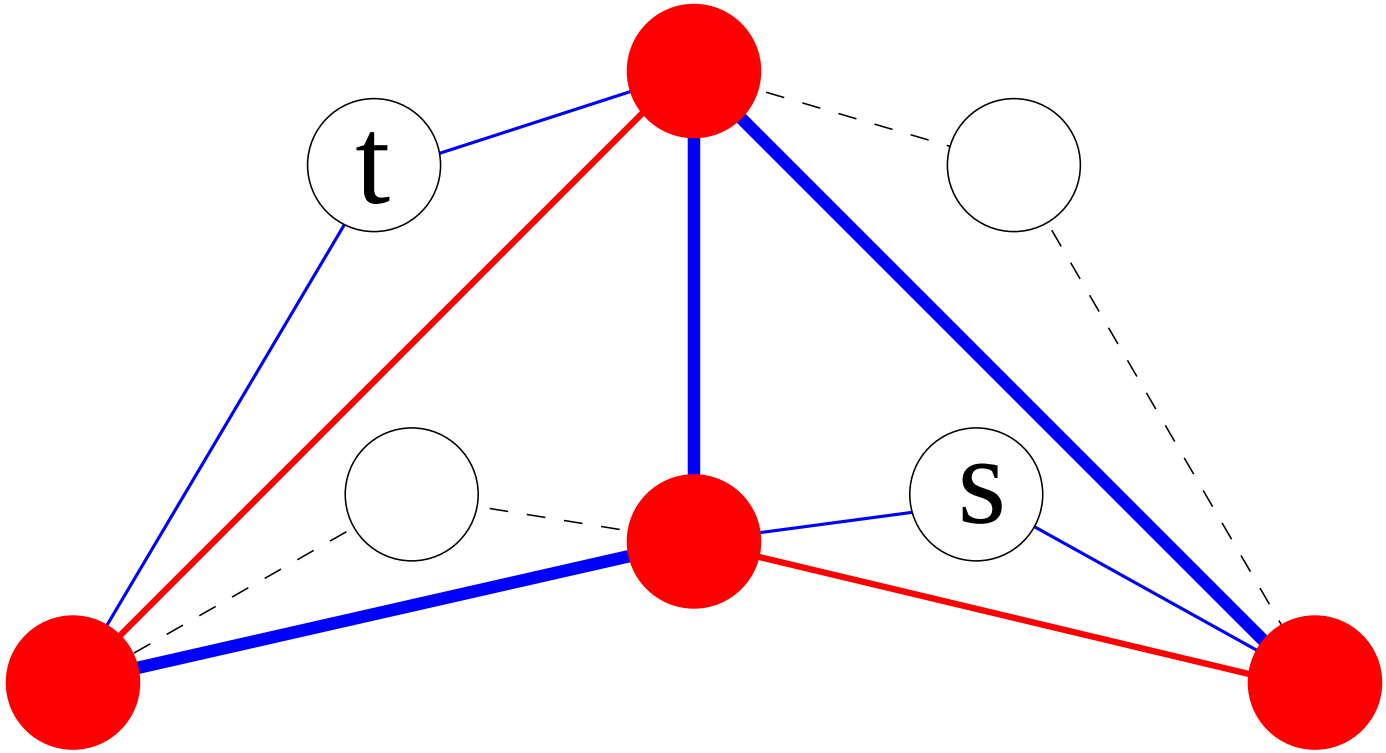
Suche: Intuition

- bidirektional
- suche in G bis Suchbäume durch Knoten in S ‘abgedeckt’



Suche: Intuition

- bidirektional
- suche in G bis Suchbäume durch Knoten in S ‘abgedeckt’
- setze Suche ausschließlich in G' fort



Highway-Node Routing

- verwende Overlaygraph-Konzept iterativ
- klassifiziere Knoten nach 'Wichtigkeit' mit Hilfe der Highway Hierarchien

  d.h. bestimme Knotenmengen $V =: S_0 \supseteq S_1 \supseteq S_2 \supseteq S_3 \ldots \supseteq S_L$ 13 min

  (entscheidender Unterschied zu [Holzer, Schulz, Wagner, Weihe, Zaroliagis])
- konstruiere mehrstufigen Overlaygraphen 2 min

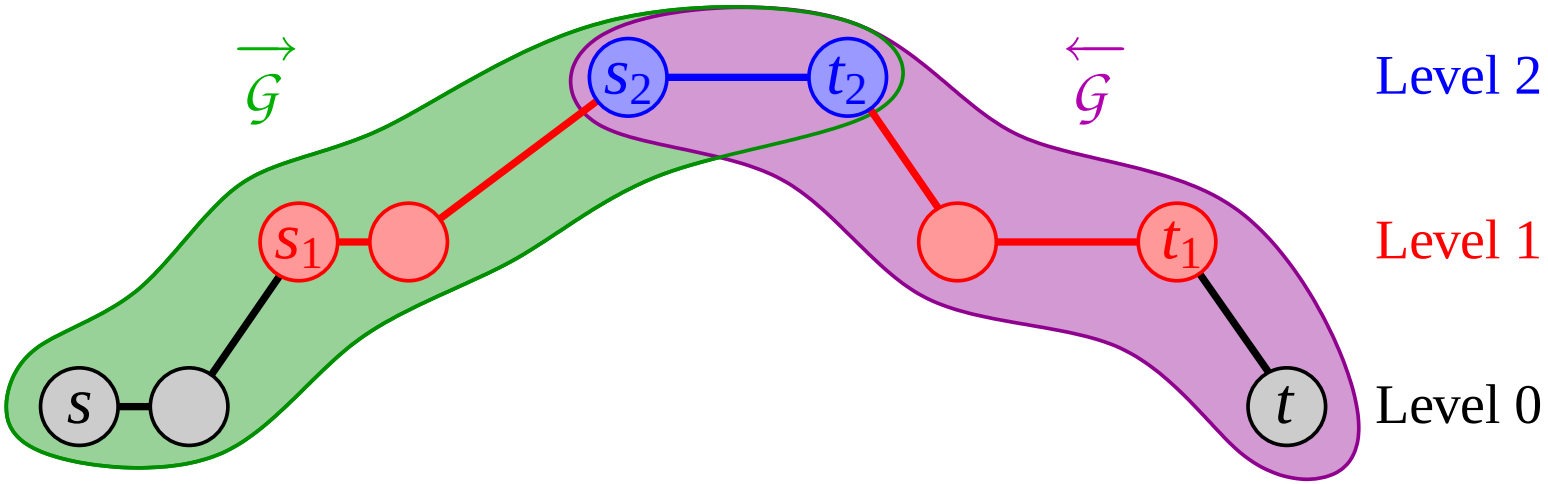
  $G_0 = G = (V,E), G_1 = (S_1,E_1), G_2 = (S_2,E_2), \ldots, G_L = (S_L,E_L)$

  (fortgeschrittene Techniken benötigt)

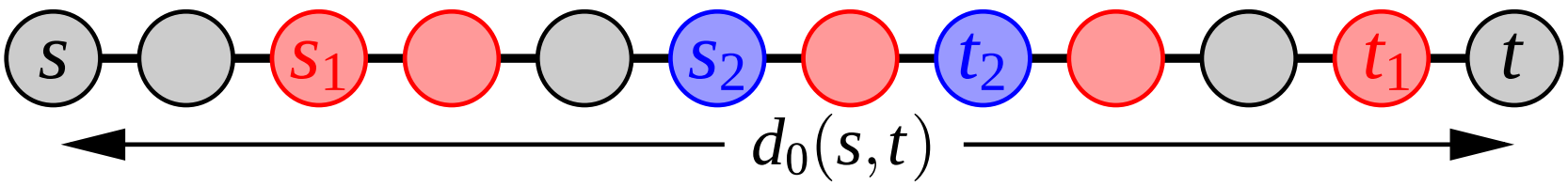
Schultes, Sanders. WEA 2007.

Suchalgorithmus

- Knotenlevel $\ell(u) := \max\{\ell \mid u \in S_\ell\}$
- Vorwärtssuchgraph $\overrightarrow{G} := \left(V, \left\{(u,v) \mid (u,v) \in \bigcup_{i=\ell(u)}^{L} E_i\right\}\right)$
- Rückwärtssuchgraph $\overleftarrow{G} := \left(V, \left\{(u,v) \mid (v,u) \in \bigcup_{i=\ell(u)}^{L} E_i\right\}\right)$
- eine einfache Dijkstra-Suche in $\overrightarrow{G}$ und eine in $\overleftarrow{G}$

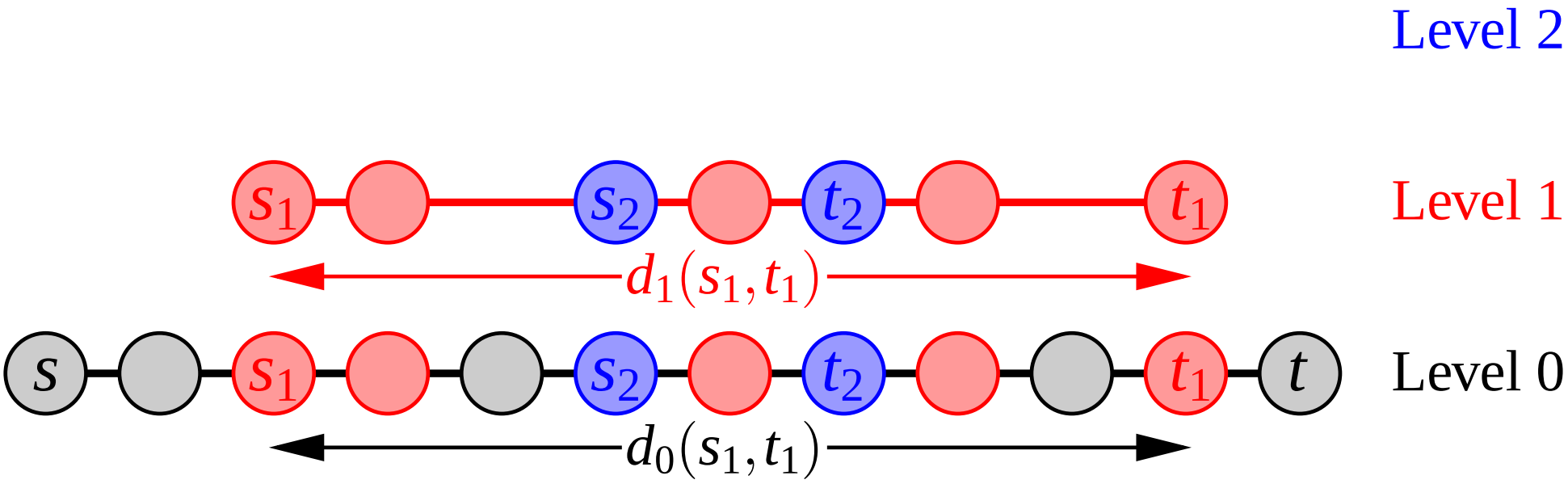


Korrektheitsbeweis



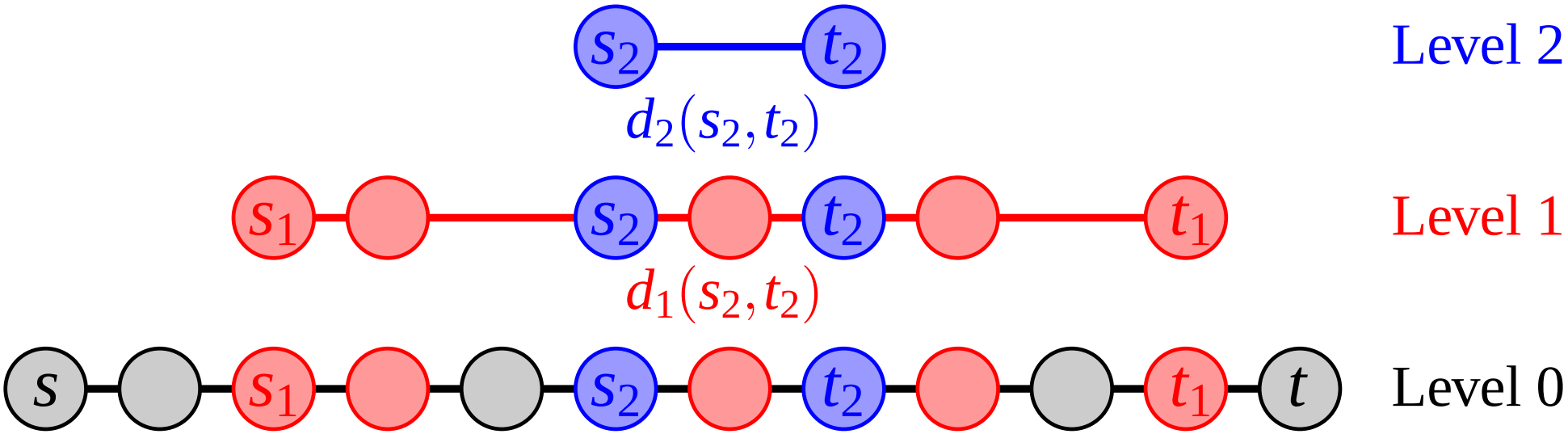
kürzester Weg von s nach t in $G = G_0$

Korrektheitsbeweis



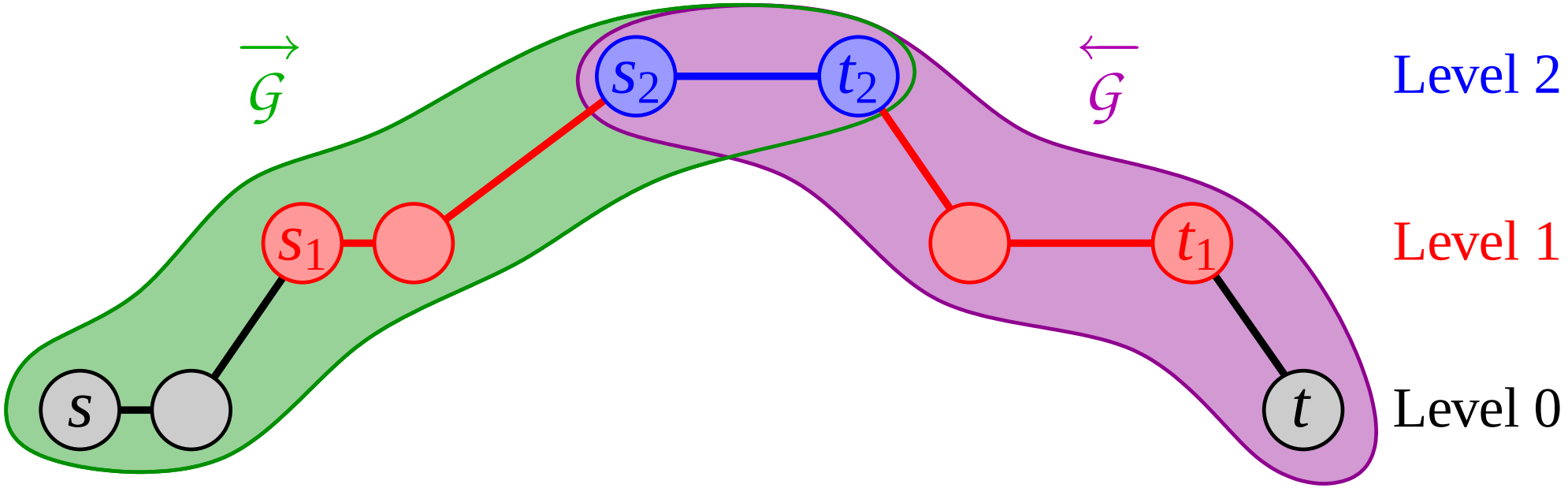
Overlaygraph G_1 erhält den Abstand von $s_1 \in S_1$ nach $t_1 \in S_1$

Korrektheitsbeweis



Overlaygraph G_2 erhält den Abstand von $s_2 \in S_2$ nach $t_2 \in S_2$

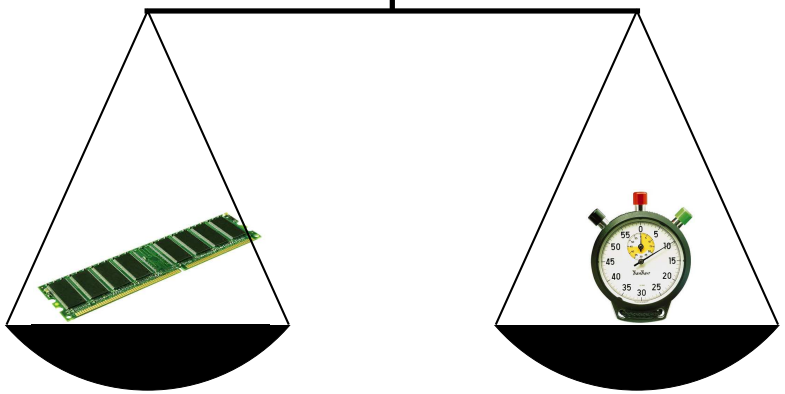
Korrektheitsbeweis



$$\overrightarrow{G} := \left(V, \left\{(u,v) \mid (u,v) \in \bigcup_{i=\ell(u)}^{L} E_i\right\}\right)$$
$$\overleftarrow{G} := \left(V, \left\{(u,v) \mid (v,u) \in \bigcup_{i=\ell(u)}^{L} E_i\right\}\right)$$

Speicherverbrauch / Suchzeiten

unterschiedliche Kompromisse



zum Beispiel:

- 9.5 Byte pro Knoten Mehrverbrauch $\longrightarrow$ 0.89 ms

  kompletten mehrstufigen Overlaygraphen speichern

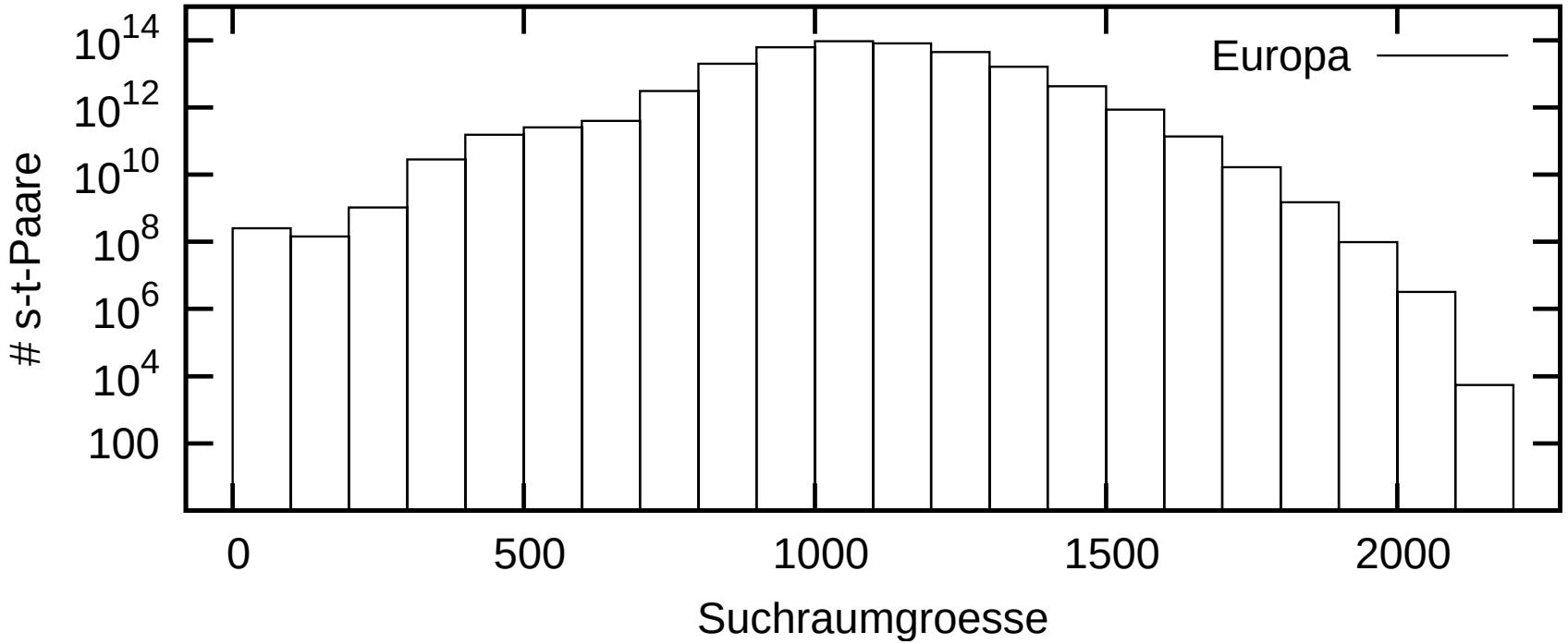
- 0.7 Byte pro Knoten Mehrverbrauch $\longrightarrow$ 1.44 ms

  nur Vorwärts- und Rückwärts-Suchgraphen $\overrightarrow{\mathcal{G}}$ und $\overleftarrow{\mathcal{G}}$ speichern

  ($\overrightarrow{\mathcal{G}}$ und $\overleftarrow{\mathcal{G}}$ sind unabhängig von s und t)

Suchzeiten unter Verwendung der Stall-on-Demand Technik

Obere Schranken



Garantie für Europa: maximale Suchraumgröße = 2 148 Knoten

Dynamische Szenarien

- tausche Kostenfunktion aus

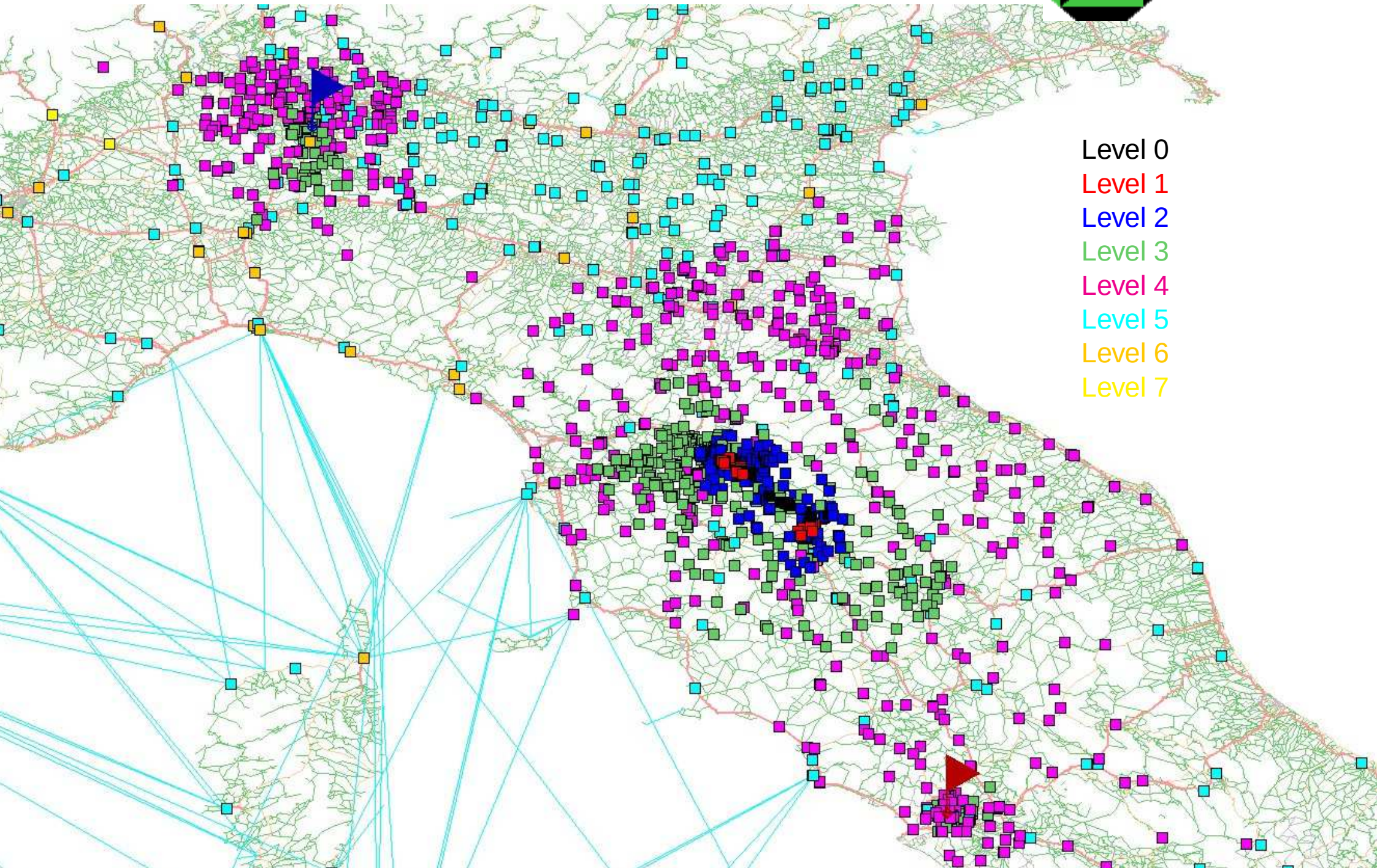
  typischerweise < 2 min

- ändere einzelne Kantengewichte
  - aktualisiere die Datenstrukturen

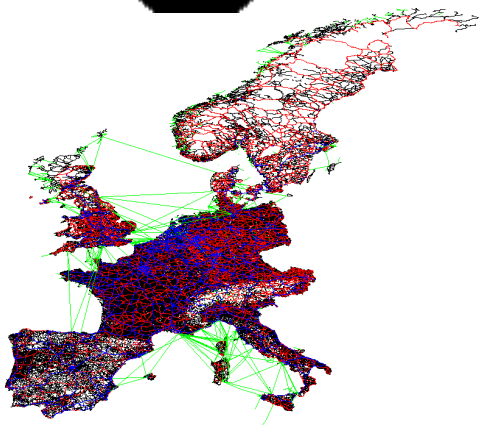
    2 – 40 ms pro Kantenänderung

  ODER
  - umfahre die Staus

    z.B. 3.6 ms bei 100 Staus


Level 0
Level 1
Level 2
Level 3
Level 4
Level 5
Level 6
Level 7

Zusammenfassung



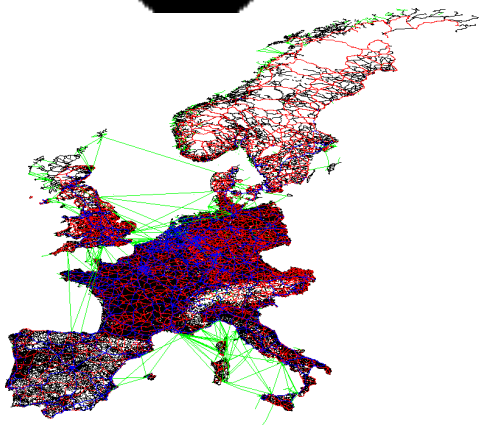
- **Umgang mit sehr großen Straßennetzen**
- **statisches Punkt-zu-Punkt Routing**
  - schnellste Suchzeiten — Transit-Node Routing
  - schnellste Vorberechnung — Highway Hierarchien
  - geringster Speicherverbrauch — Highway-Node Routing
- **dynamisches Punkt-zu-Punkt Routing**
  - Austausch der Kostenfunktion
  - Änderung einzelner Kantengewichte

  } Highway-Node Routing
- **Distanztabellen-Berechnung** — Many-to-Many

Ausblick: Wissenschaft

- R. Jacob and S. Sachdeva. I/O efficieny of highway hierarchies.
  Technical Report, ETH Zürich, 2006.
- A. V. Goldberg, H. Kaplan, R. F. Werneck. Better landmarks within reach.
  In *6th Workshop on Experimental Algorithms (WEA)*, 2007.
- R. Bauer and D. Delling. SHARC: Fast and robust unidirectional routing.
  In *Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2008.
- G. Nannicini et al. Fast paths in large-scale dynamic road networks.
  Computational Optimization and Applications, accepted for publication.
- R. Geisberger, P. Sanders, D. Schultes, D. Delling. Contraction hierarchies . . .
  Submitted for publication, 2008.
- P. Sanders, D. Schultes, C. Vetter. Mobile route planning. Submitted for publication, 2008.
- R. Bauer, D. Delling, P. Sanders, D. Schieferdecker, D. Schultes, D. Wagner.
  Combining Hierarchical and Goal-Directed Speed-Up Techniques for Dijkstra's Algorithm.
  Submitted for publication, 2008.

Zusammenfassung



- **Umgang mit sehr großen Straßennetzen**
- **statisches Punkt-zu-Punkt Routing**
  - schnellste Suchzeiten — Transit-Node Routing
  - schnellste Vorberechnung — Highway Hierarchien
  - geringster Speicherverbrauch — Highway-Node Routing
- **dynamisches Punkt-zu-Punkt Routing**
  - Austausch der Kostenfunktion
  - Änderung einzelner Kantengewichte

  } Highway-Node Routing
- **Distanztabellen-Berechnung** — Many-to-Many