



Schnelle und genaue **Routenplanung**

Peter Sanders

Dominik Schultes

Institut für Theoretische Informatik – Algorithmik II

Universität Karlsruhe

Tag der Informatik, 15. Juli 2005



Wie komme ich von A nach B ?

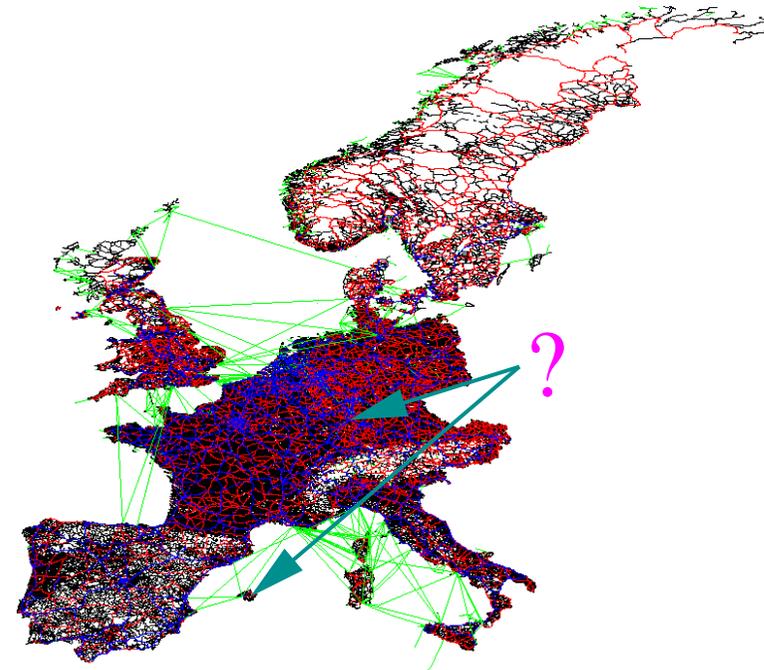
Anwendungen

- Routenplanung im Internet (z.B. www.map24.de)
- Navigationssystem im Auto
- ...



Anforderungen

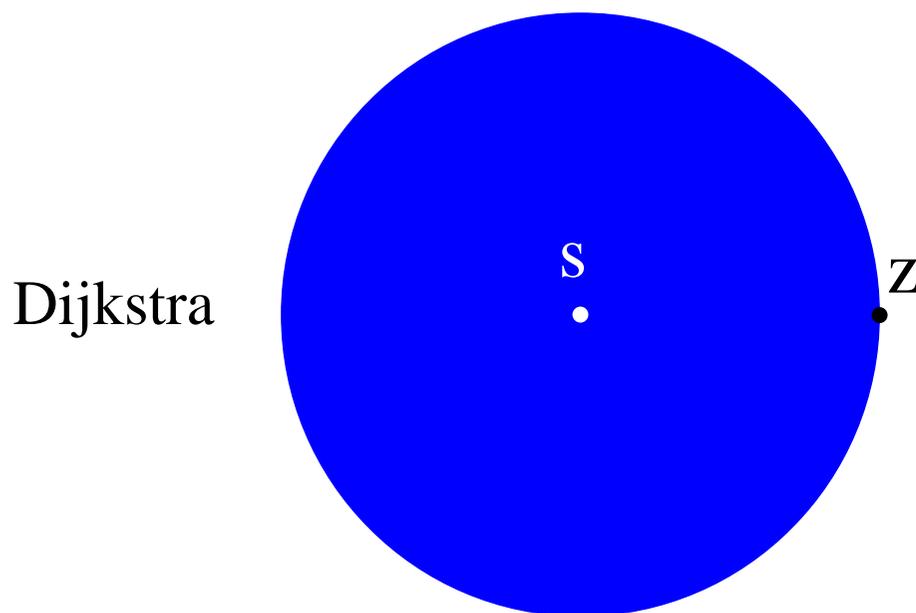
- genaue** schnellste Routen
- schnelle** Berechnung
- geringer** Speicherplatzverbrauch





DIJKSTRAS Algorithmus

klassisches Verfahren aus der Graphentheorie
zur **Berechnung von kürzesten Pfaden**



ungeeignet für große Straßengraphen

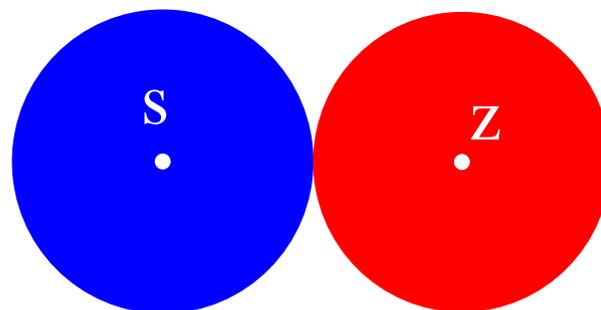
(z.B. Westeuropa: 22 Millionen Straßenabschnitte)



Bidirektionale Suche

Verbesserung von DIJKSTRAs Algorithmus

bidirektionaler
Dijkstra

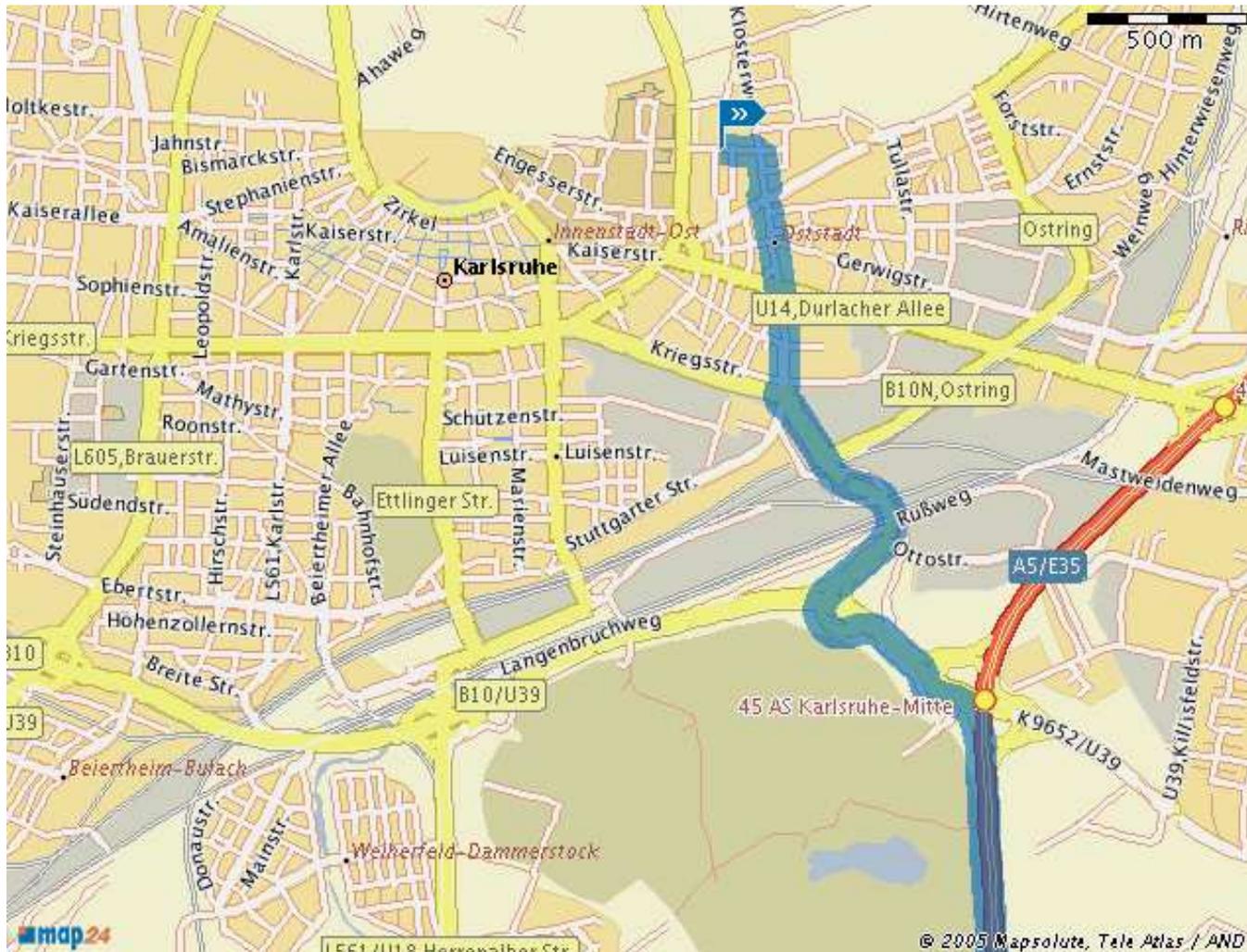


Halbierung des Suchraums möglich,
aber immer noch **zu langsam**



Naive Routenplanung

1. Suche nächste sinnvolle Autobahnauffahrt





Naive Routenplanung

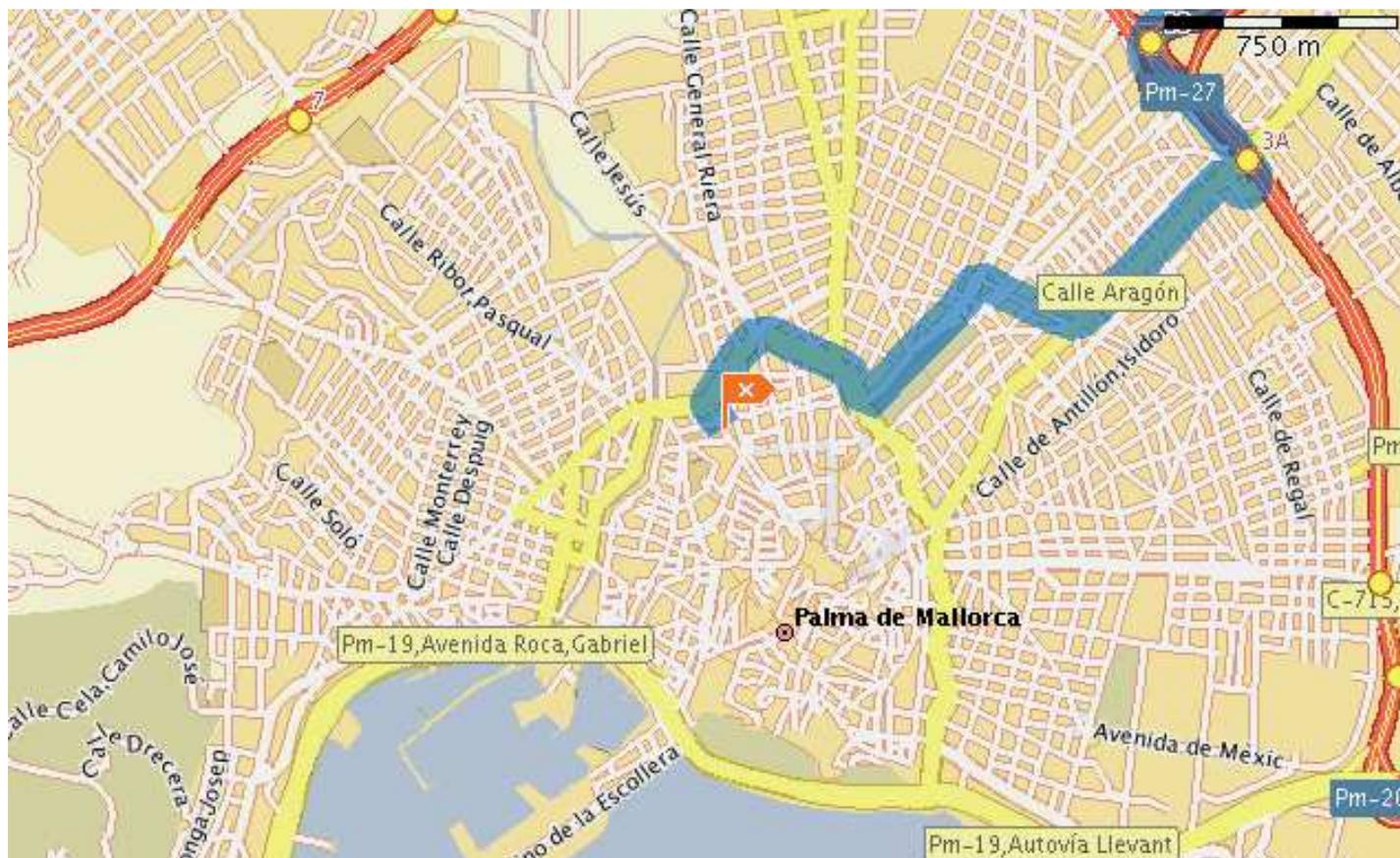
1. Suche nächste sinnvolle Autobahnauffahrt
2. Fahre auf Autobahnen möglichst nahe ans Ziel heran





Naive Routenplanung

1. Suche nächste sinnvolle Autobahnauffahrt
2. Fahre auf Autobahnen möglichst nahe ans Ziel heran
3. Suche von der Autobahnausfahrt den Weg zum Ziel





Kommerzielle Systeme

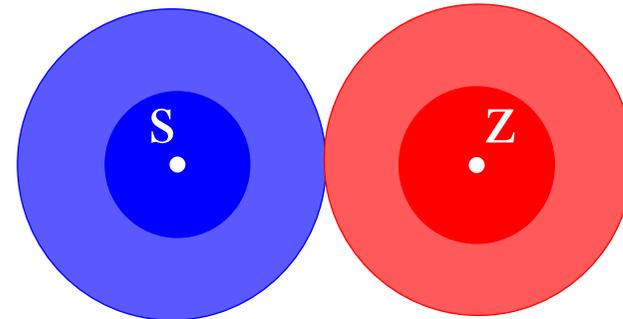
1. Suche von Start und Ziel aus (**bidirektional**)
in einem bestimmten Radius (z.B. **20 km**),
betrachte dabei **alle Straßen**
2. Suche dann in einem größeren Radius (z.B. **100 km**) weiter,
betrachte dabei nur **Land-/Bundesstraßen und Autobahnen**
3. Suche weiter,
betrachte nur noch **Autobahnen**

schnell, aber ungenau

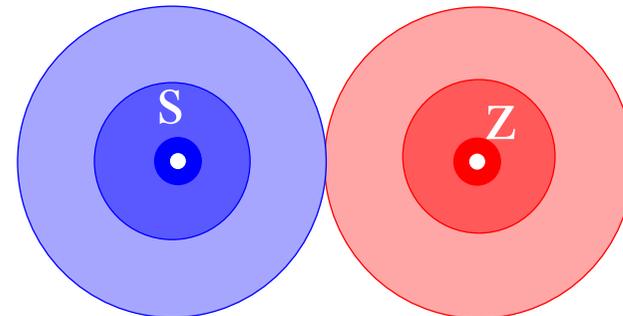


Genaue Highway Hierarchie

vollständige Suche im **lokalen** Bereich
Suche im (**dünnere**n) **Highway Netzwerk**



Verfahren iterieren \rightsquigarrow **Highway Hierarchie**

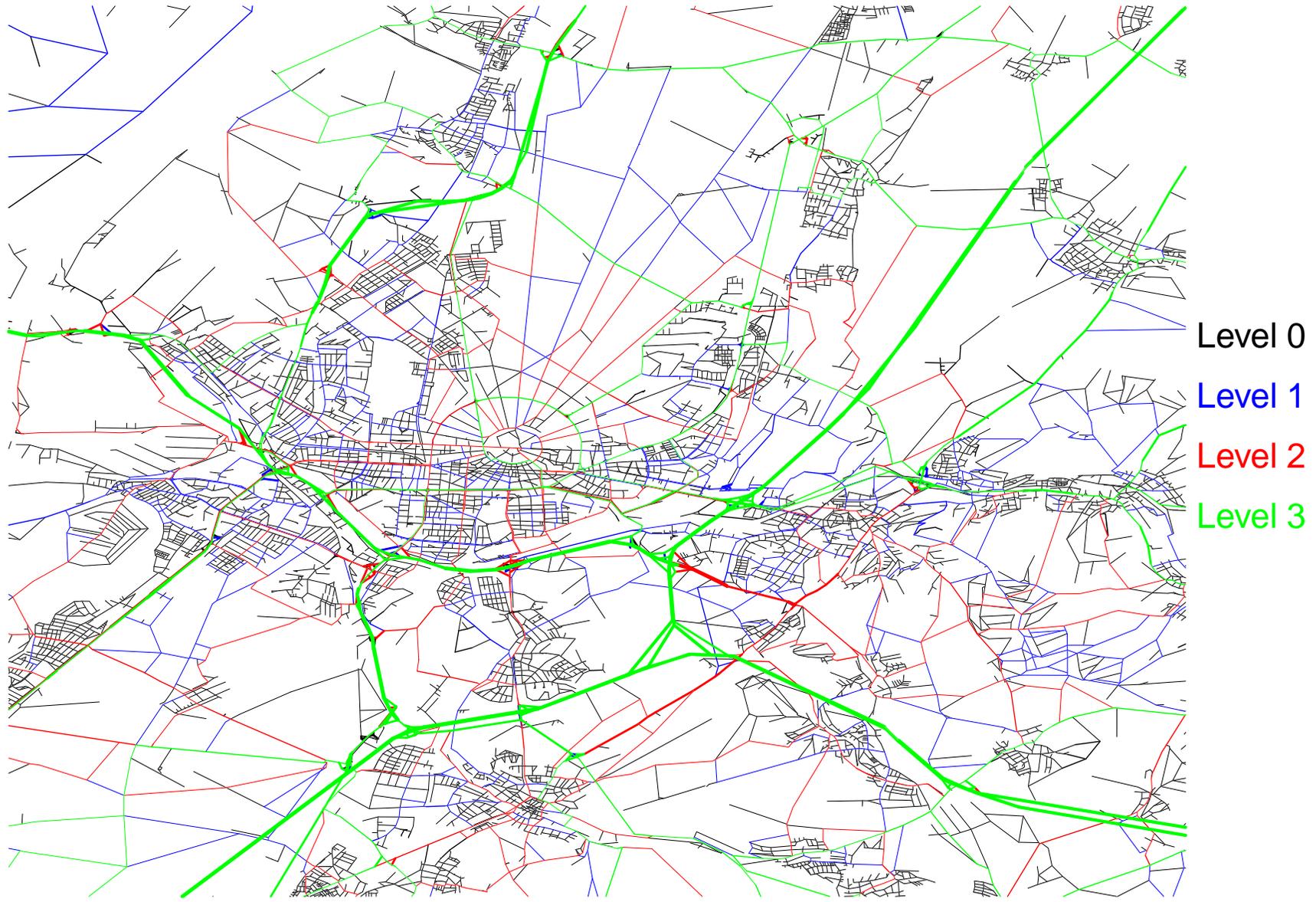


Highway Netzwerk = **minimaler** Graph,

bei dem alle kürzesten Wege **erhalten** bleiben



Beispiel: Karlsruhe





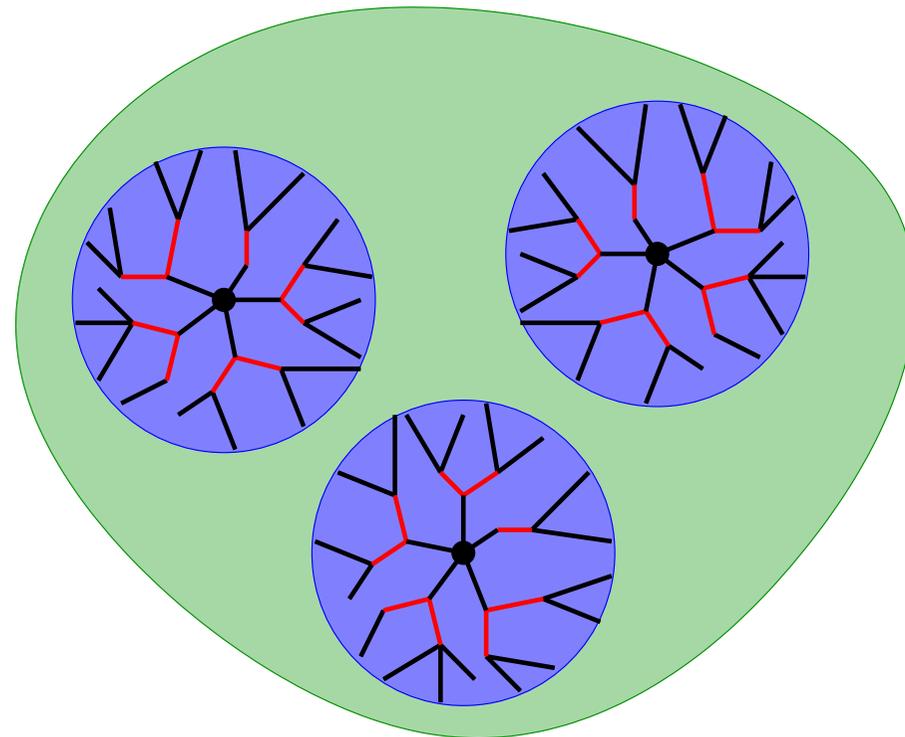
Schnelle Konstruktion

Herausforderung

Vermeidung der Vorberechnung von kürzesten Pfaden
zwischen **allen** Knotenpaaren

Lösung

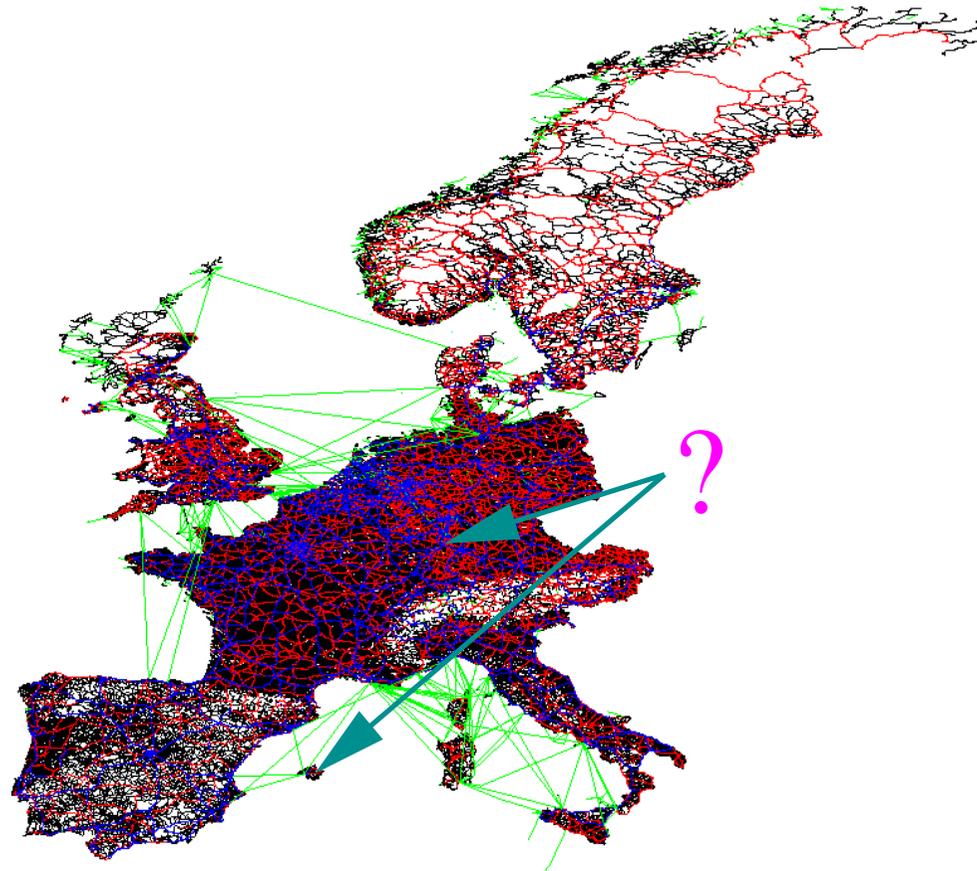
Von jedem Knoten aus:
Suche in einem **lokalen** Bereich





Suche

Beispiel: von **Karlsruhe**, Am Fasanengarten 5
nach **Palma de Mallorca**

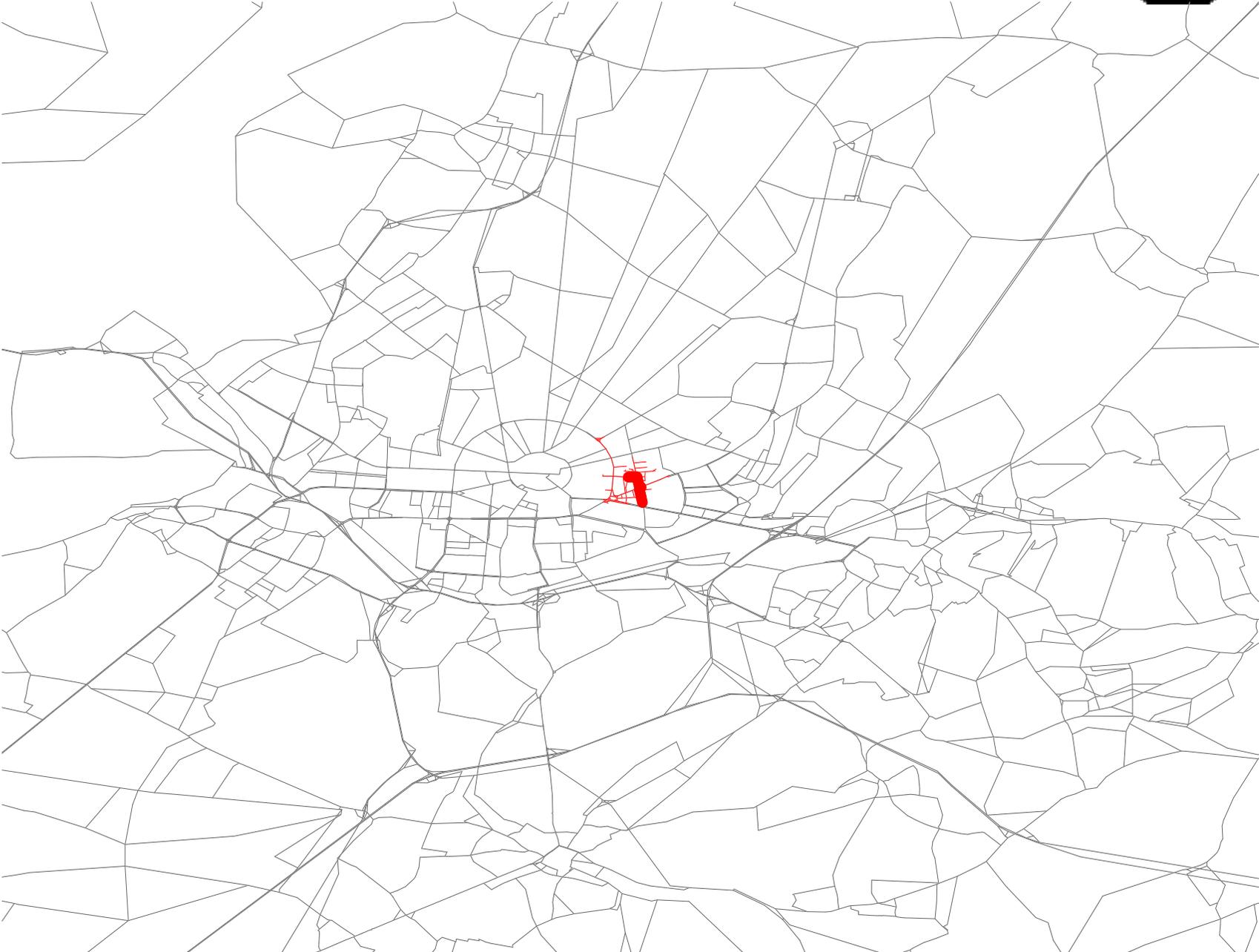


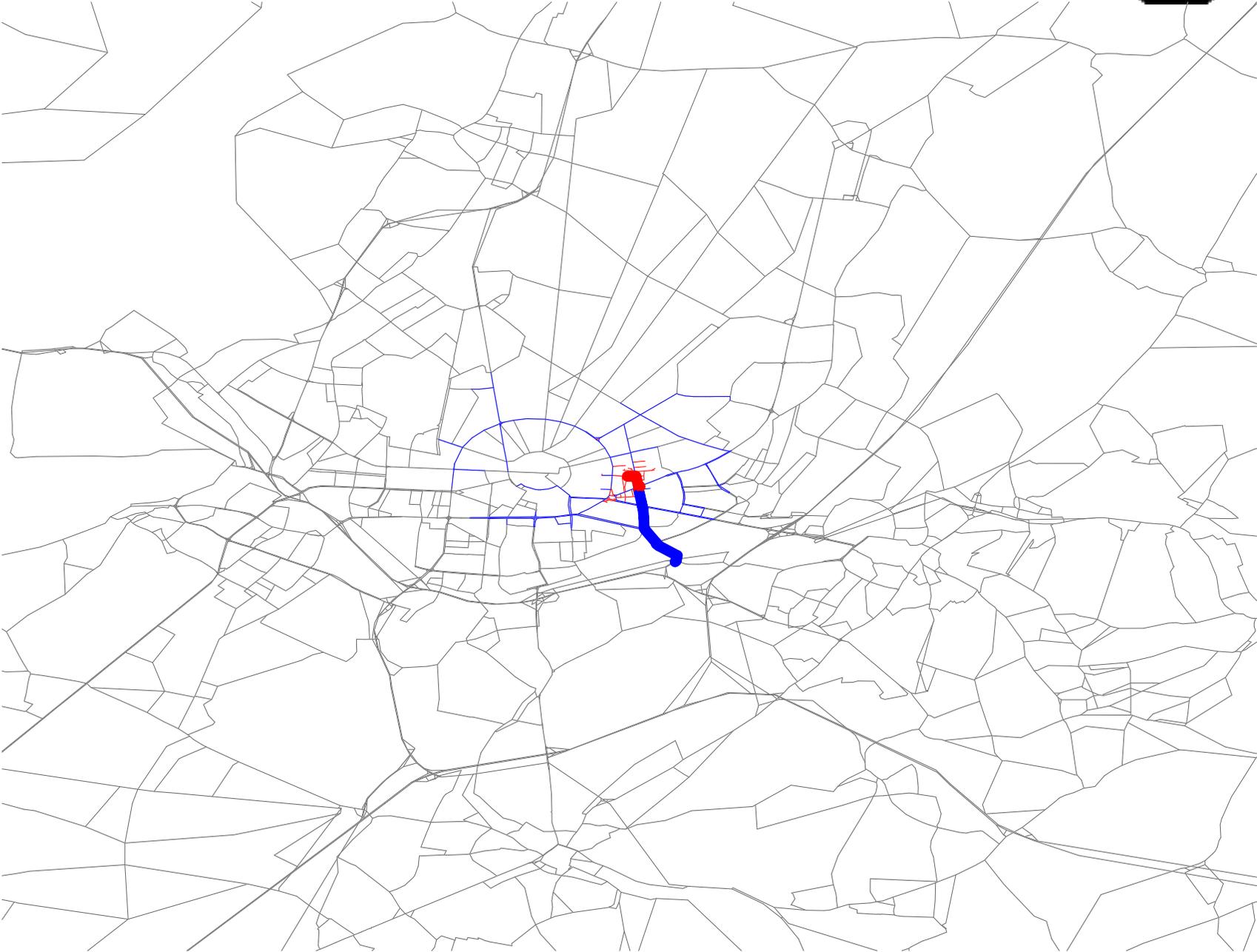


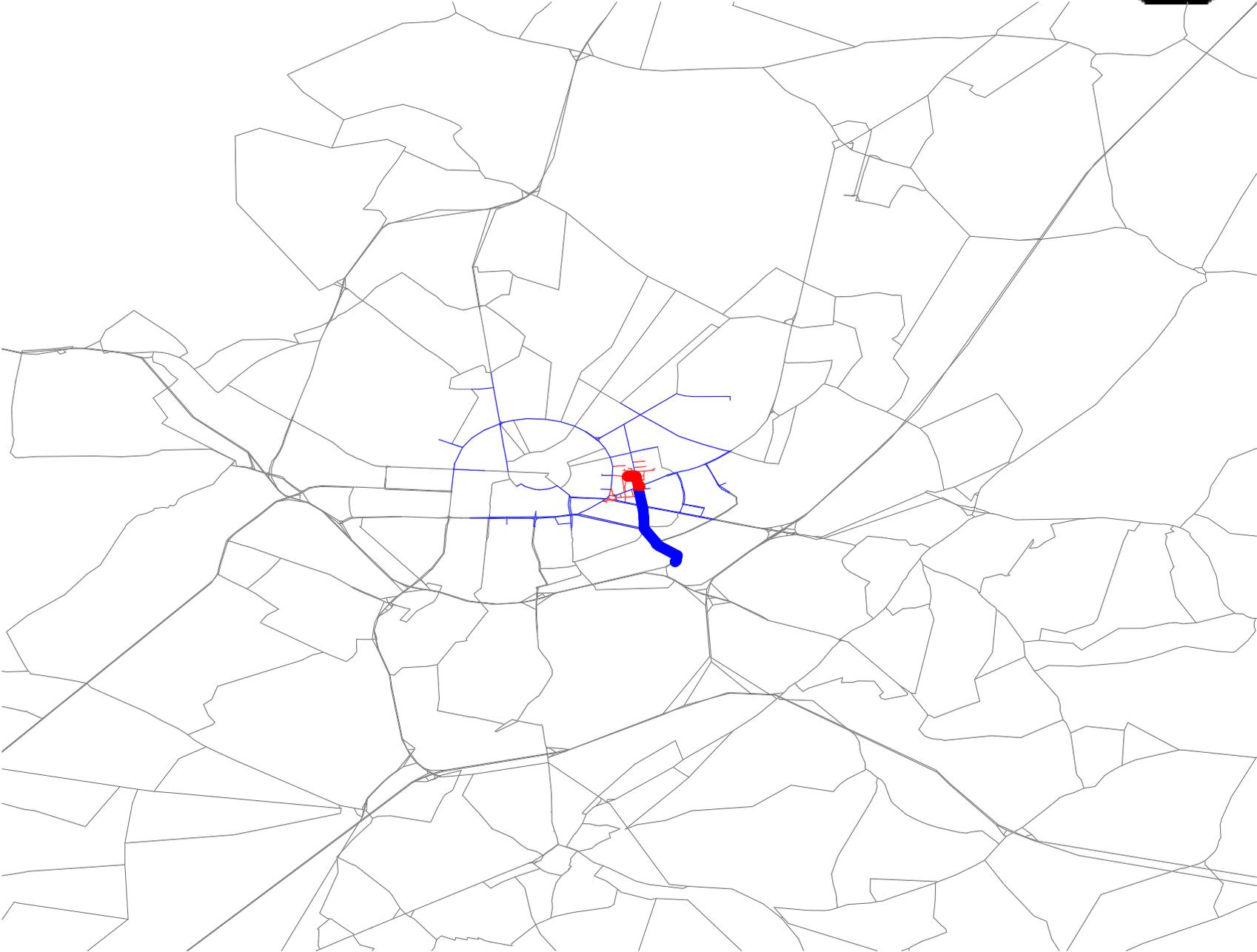


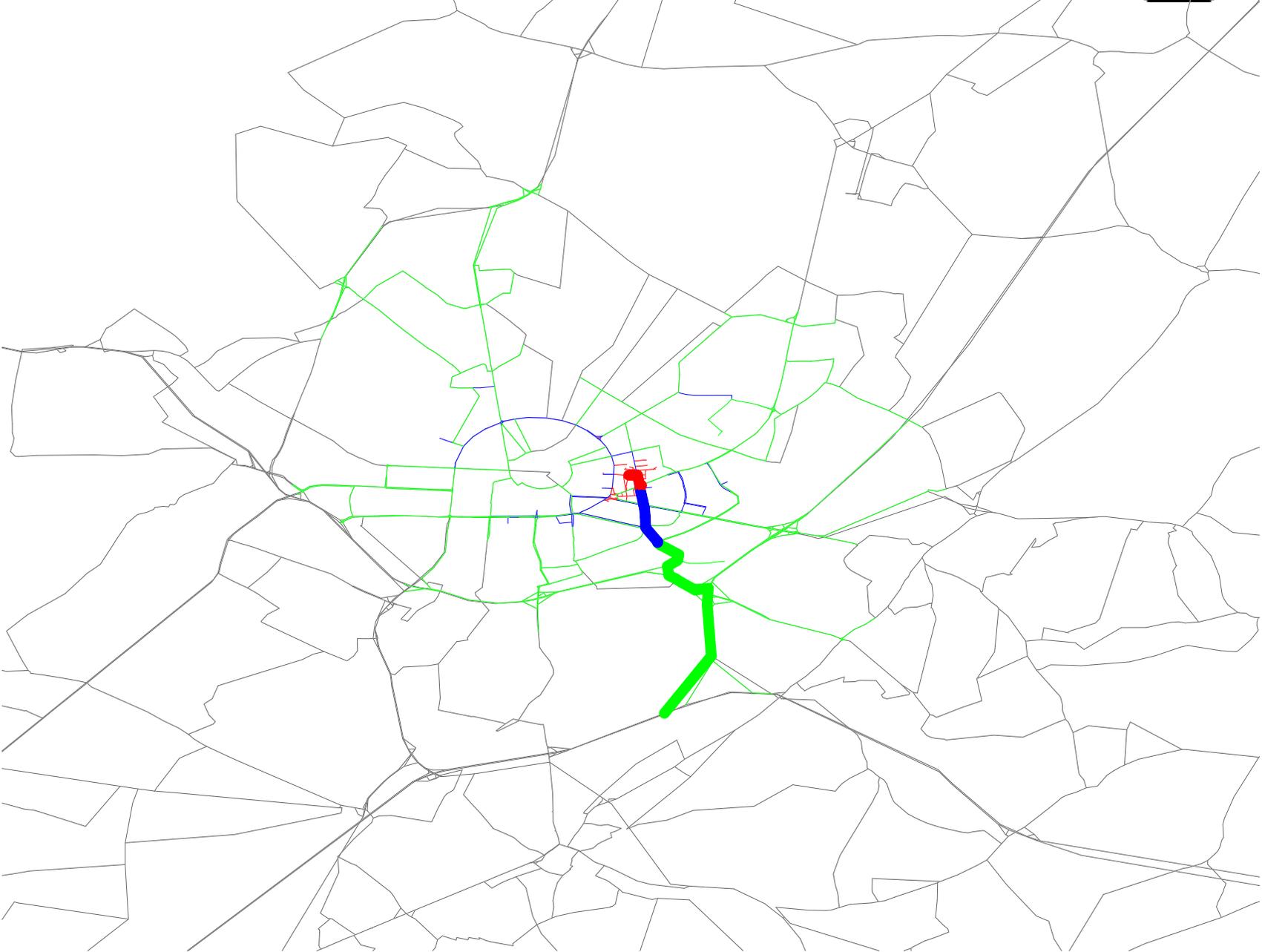
Level 0 Suchraum





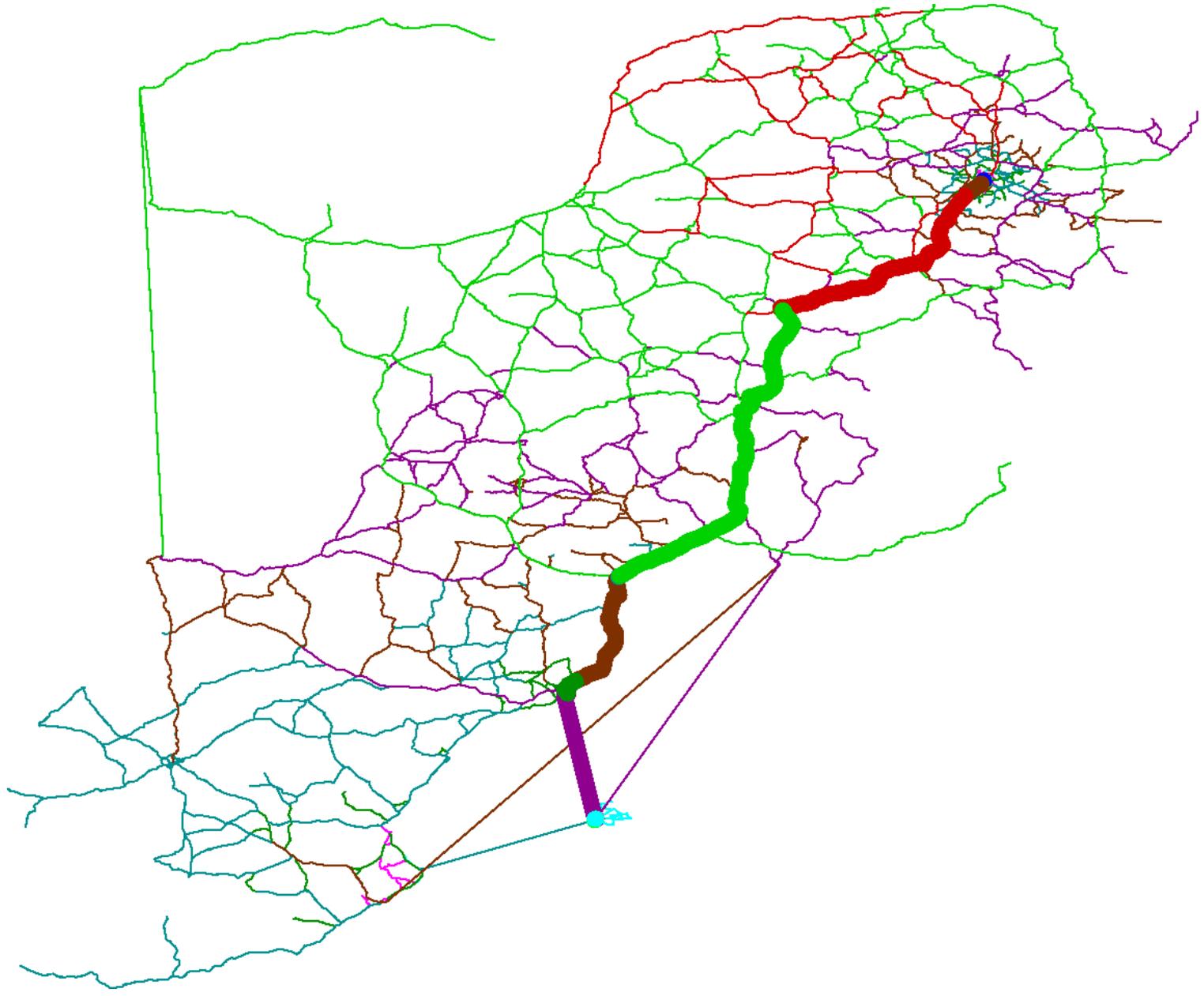






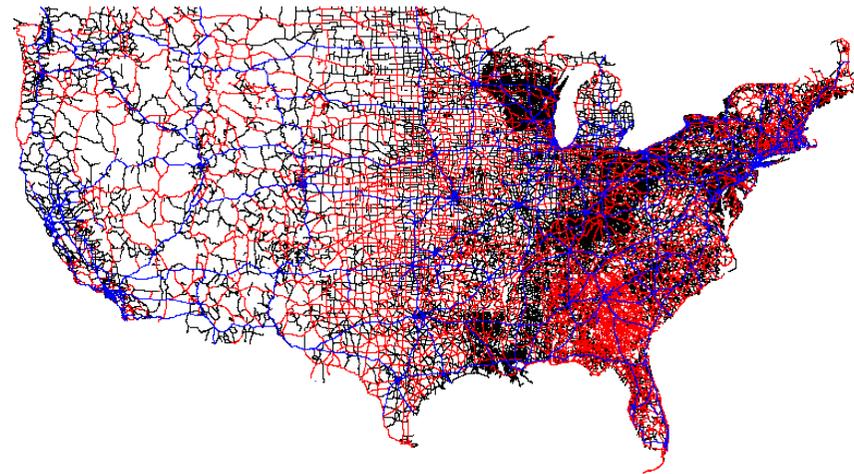
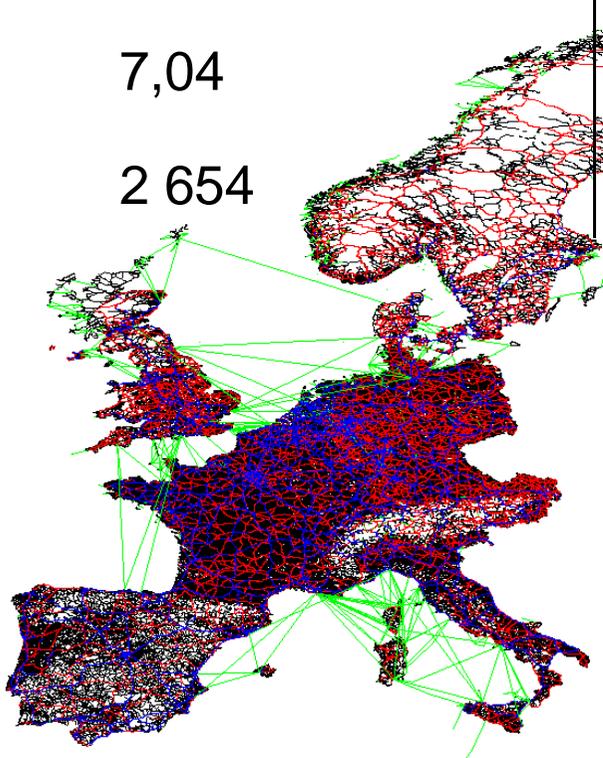


Level 10 Suchraum



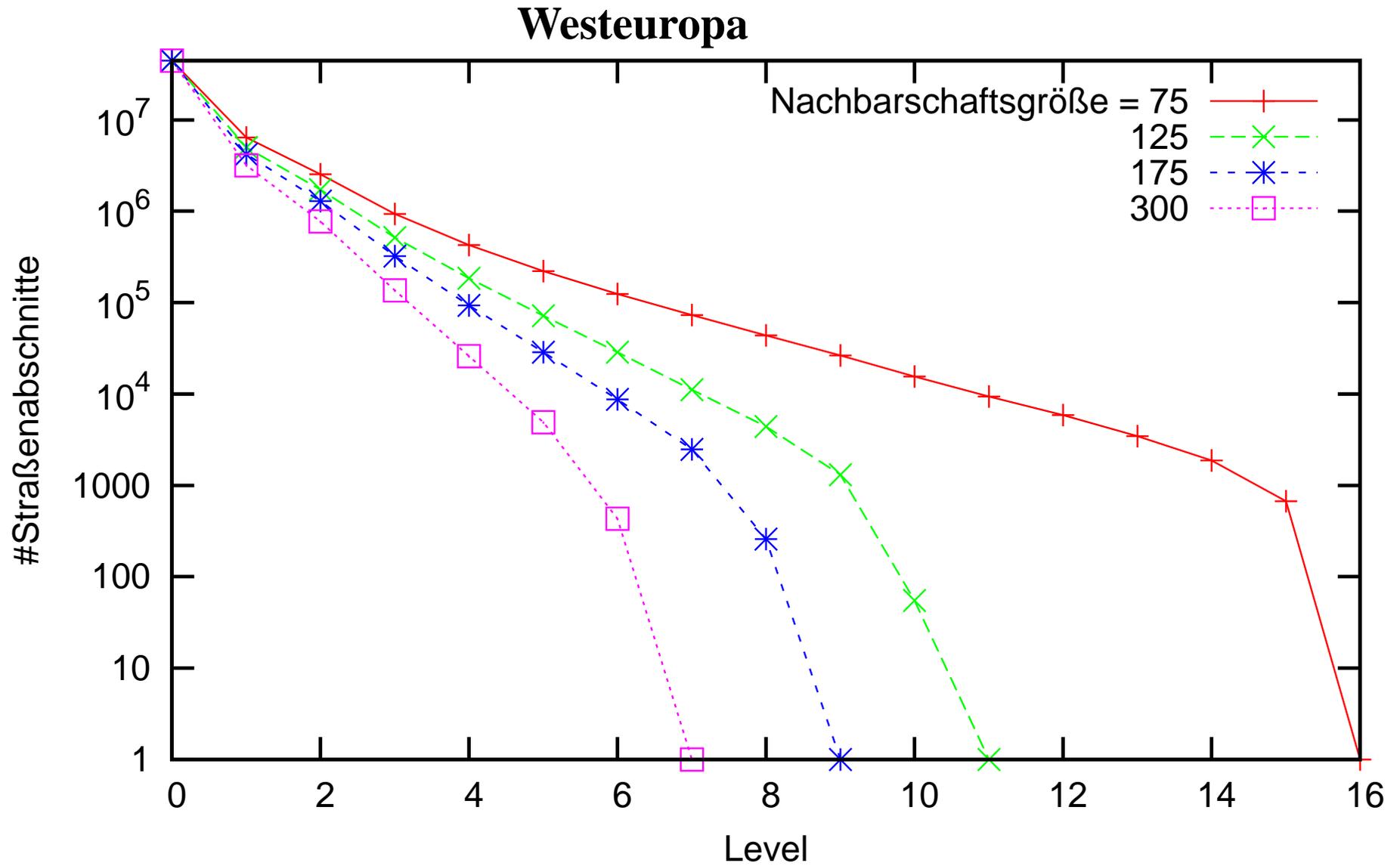


Westeuropa	Testdaten	USA
18 029 721	#Knotenpunkte	24 278 285
22 217 686	#Straßenabschnitte	29 106 596
2:43	Konstruktion [h]	4:20
7,04	Suchzeiten [ms]	7,38
2 654	Beschleunigung (↔ DIJKSTRA)	2 645





Schrumpfen der Highway Netzwerke





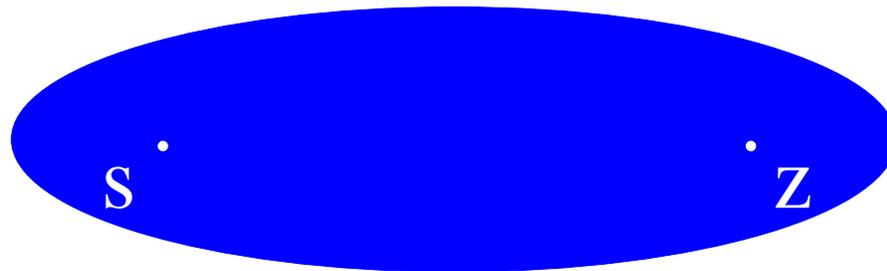
Zusammenfassung

- genaue** schnellste Routen in **großen** Straßennetzen
Highway Netzwerk z.B. Westeuropa \approx 22 Mill. Straßenabschnitte
erhält alle kürzesten Wege
- schnelle** Suche
< 8 ms im Durchschnitt
- schnelle** Vorverarbeitung (Konstruktion der Highway Hierarchie)
3 Stunden
- Platzverbrauch hält sich **in Grenzen**
1.8 GB (noch verbesserungsfähig)



Ausblick

- Kombination mit **zielgerichteten** Verfahren
(→ Arbeitsgruppe Prof. Wagner)



Suchraum

eines

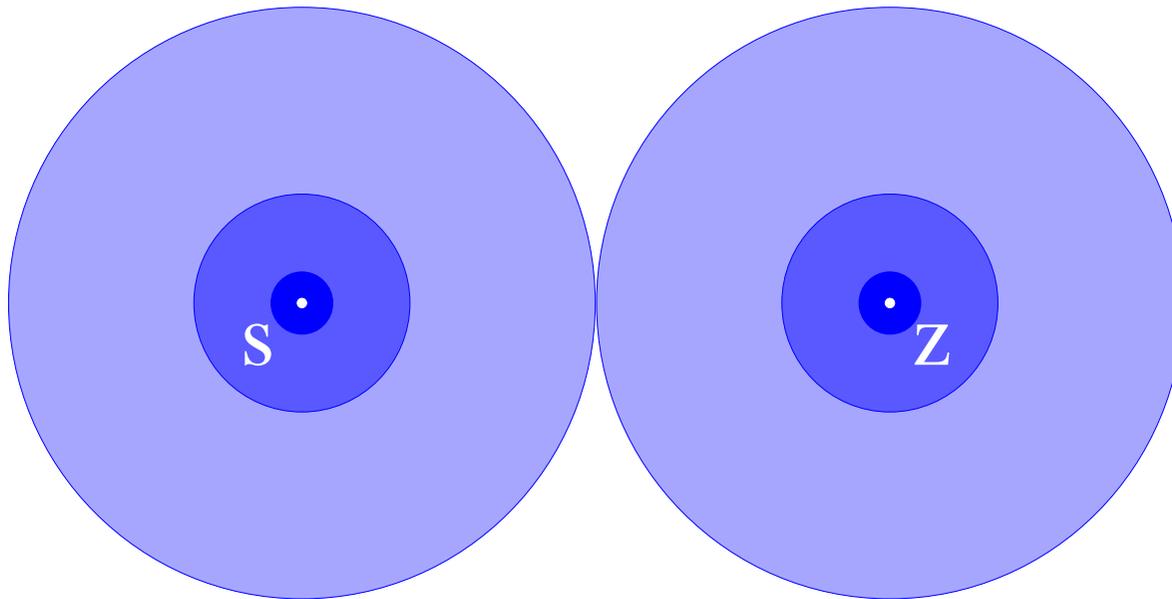
zielgerichteten

Verfahrens



Ausblick

- Kombination mit **zielgerichteten** Verfahren
(→ Arbeitsgruppe Prof. Wagner)



zum Vergleich:

Suchraum

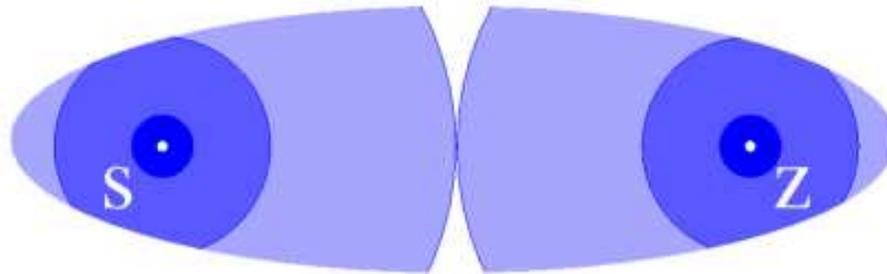
unseres

Ansatzes



Ausblick

- Kombination mit **zielgerichteten** Verfahren
(→ Arbeitsgruppe Prof. Wagner)



Ziel:

Schnittmenge

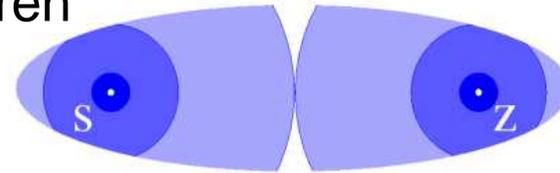
beider

Suchräume



Ausblick

- Kombination mit **zielgerichteten** Verfahren
(→ Arbeitsgruppe Prof. Wagner)



- schnelle, **lokale Aktualisierung** des Highway Netzwerks
(bspw. aufgrund eines Staus)



- Implementierung für **Mobilgeräte**



- ...