Schnelle und genaue Routenplanung

Peter Sanders **Dominik Schultes**

Institut für Theoretische Informatik – Algorithmik II

Universität Karlsruhe

Uni für Einsteiger, 22. November 2006

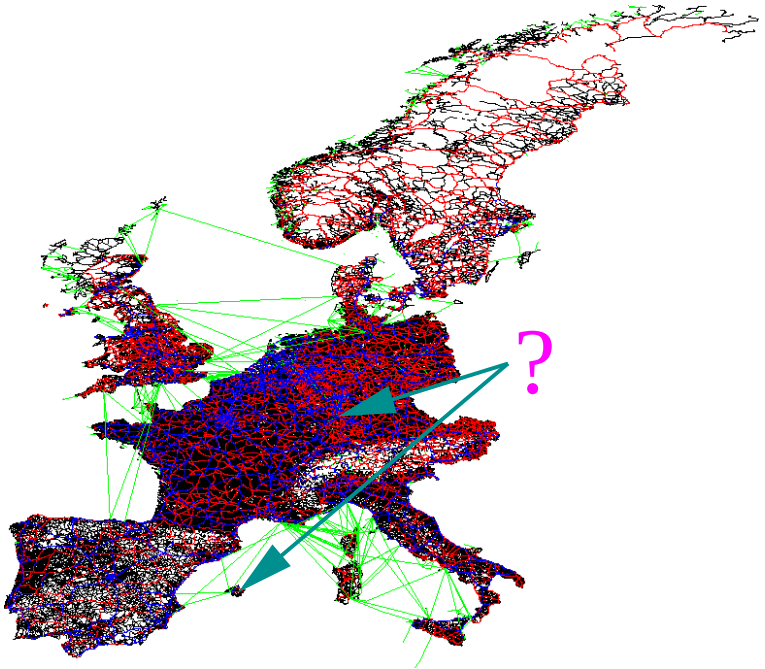
Wie komme ich von A nach B ?

Anwendungen

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

Anforderungen

- genaue schnellste Routen
- schnelle Berechnung
- geringer Speicherplatzverbrauch

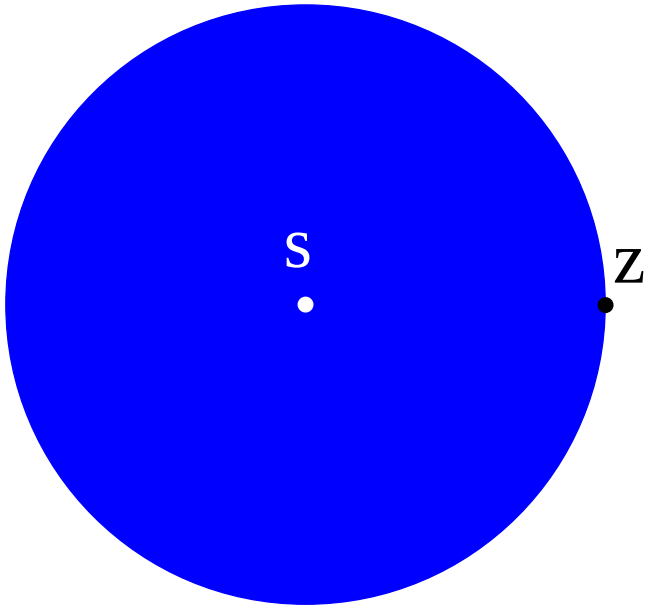


DIJKSTRAs Algorithmus

klassisches Verfahren aus der Graphentheorie

zur Berechnung von kürzesten Pfaden

Dijkstra



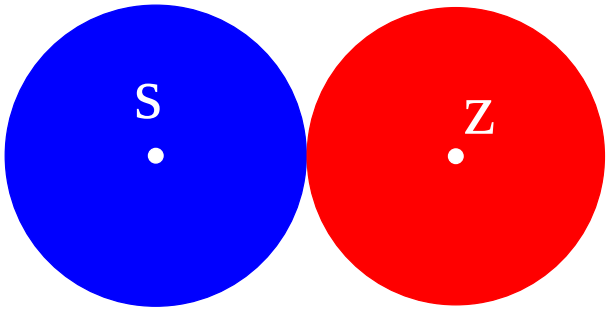
ungeeignet für große Straßengraphen

(z.B. Westeuropa: 42 Mio (gerichtete) Straßenabschnitte)

Bidirektionale Suche

Verbesserung von DIJKSTRAs Algorithmus

bidirektionaler Dijkstra



Halbierung des Suchraums möglich,

aber immer noch zu langsam

Naive Routenplanung

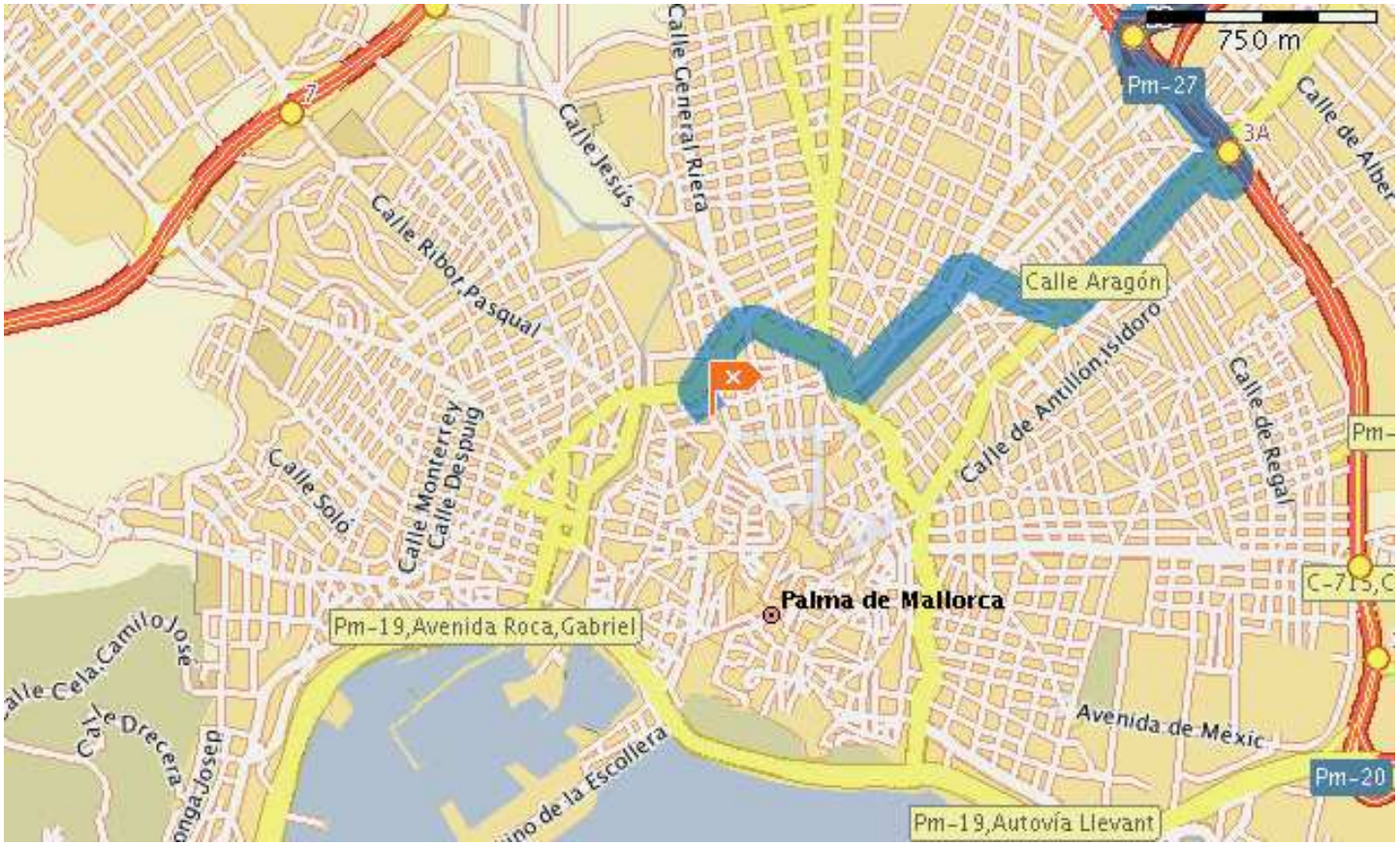
1. Suche nächste sinnvolle Autobahnauffahrt

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),

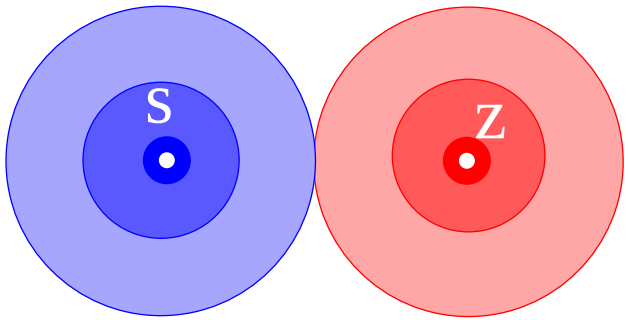
   betrachte dabei alle Straßen

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



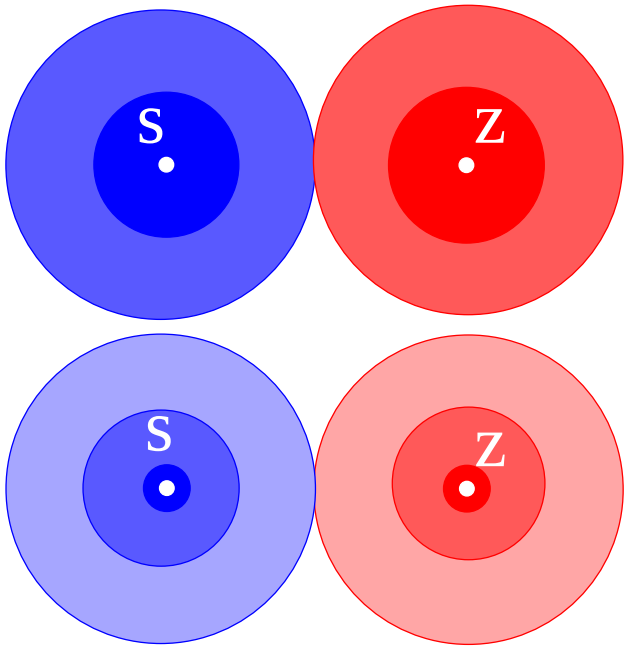
schnell, aber ungenau

Genaue Highway Hierarchie

vollständige Suche im lokalen Bereich

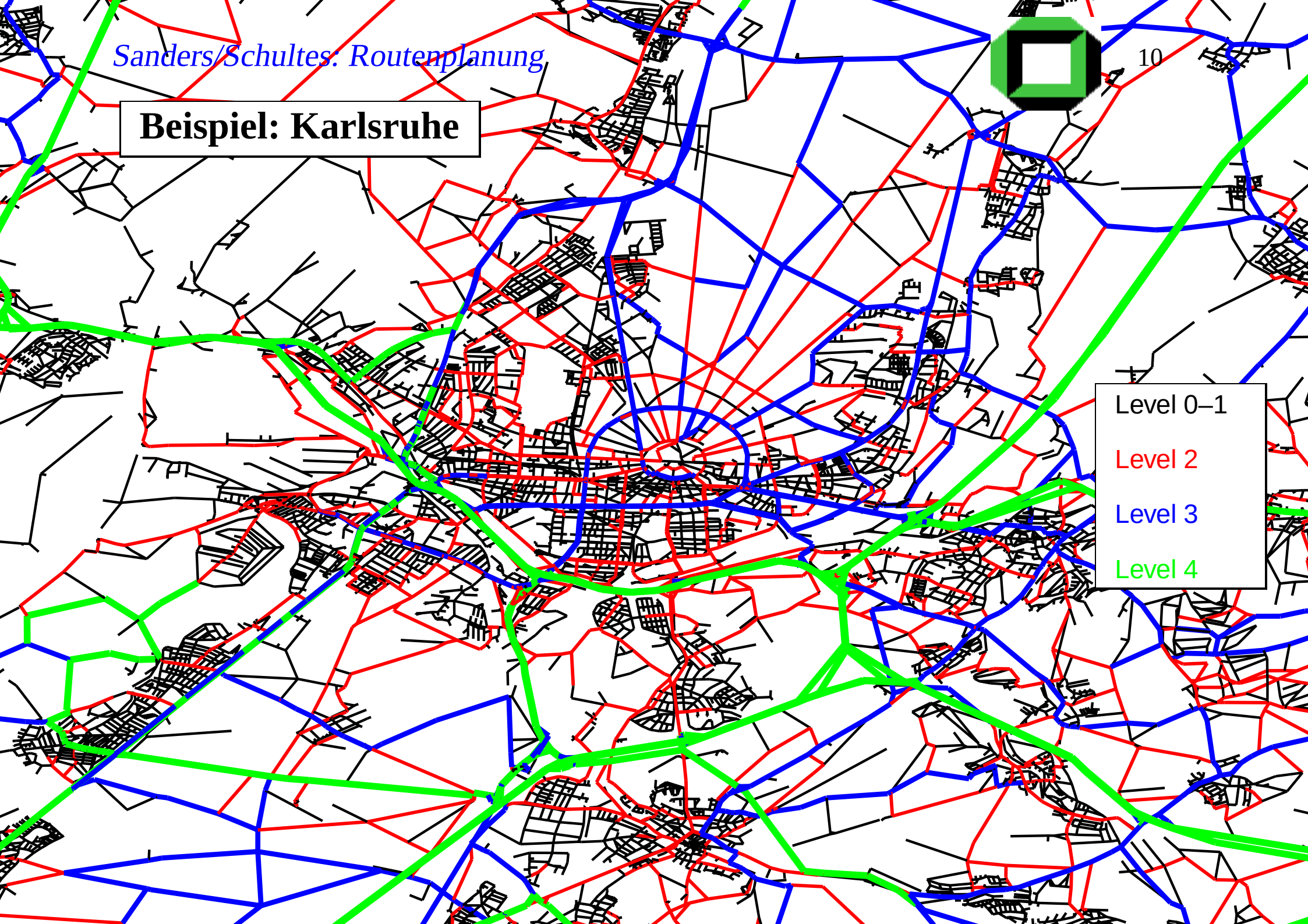
Suche im (dünneren) Highway Netzwerk

Verfahren iterieren $\rightsquigarrow$ Highway Hierarchie



Wichtig: Highway Netzwerk enthält genau die Straßenabschnitte,
die benötigt werden, um alle kürzesten Wege finden zu können,
ggf. auch *Schleichwege*!

Beispiel: Karlsruhe

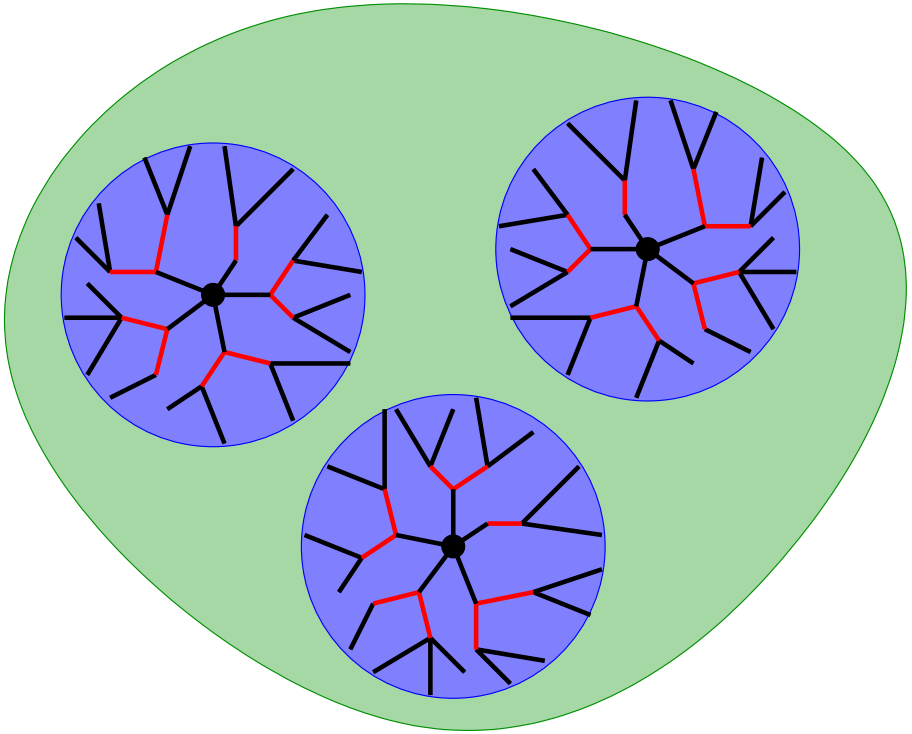


Schnelle Konstruktion

Herausforderung

Vermeidung der Vorberechnung von kürzesten Pfaden

zwischen allen Knotenpaaren



Lösung

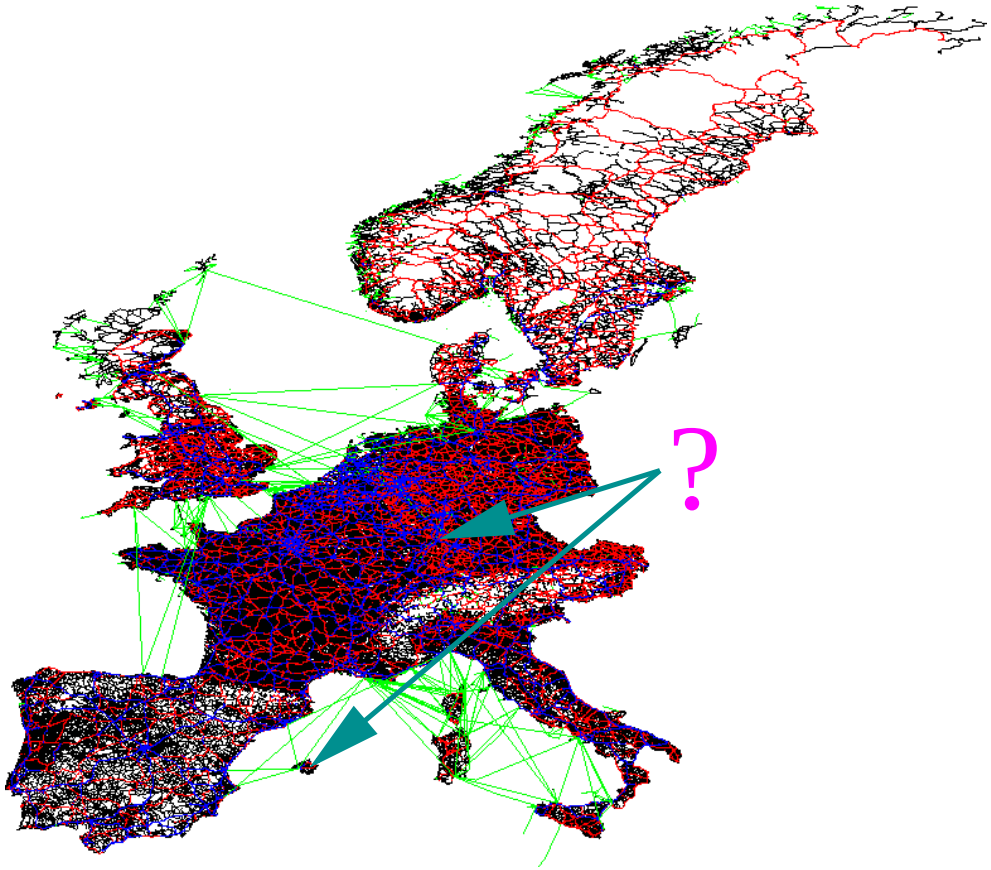
Von jedem Knoten aus:

Suche in einem lokalen Bereich

Suche

Beispiel: von Karlsruhe, Am Fasanengarten 5

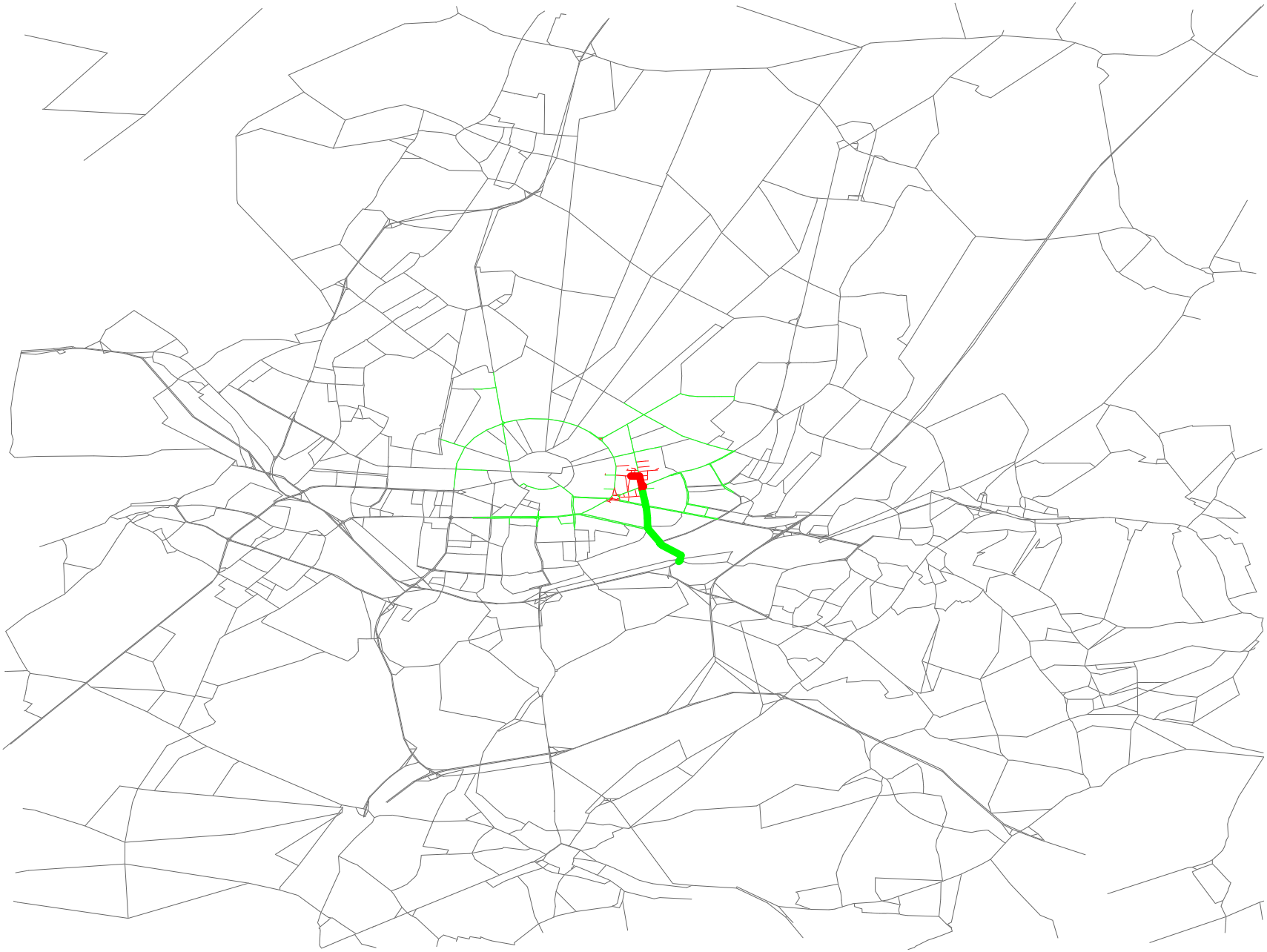
nach Palma de Mallorca



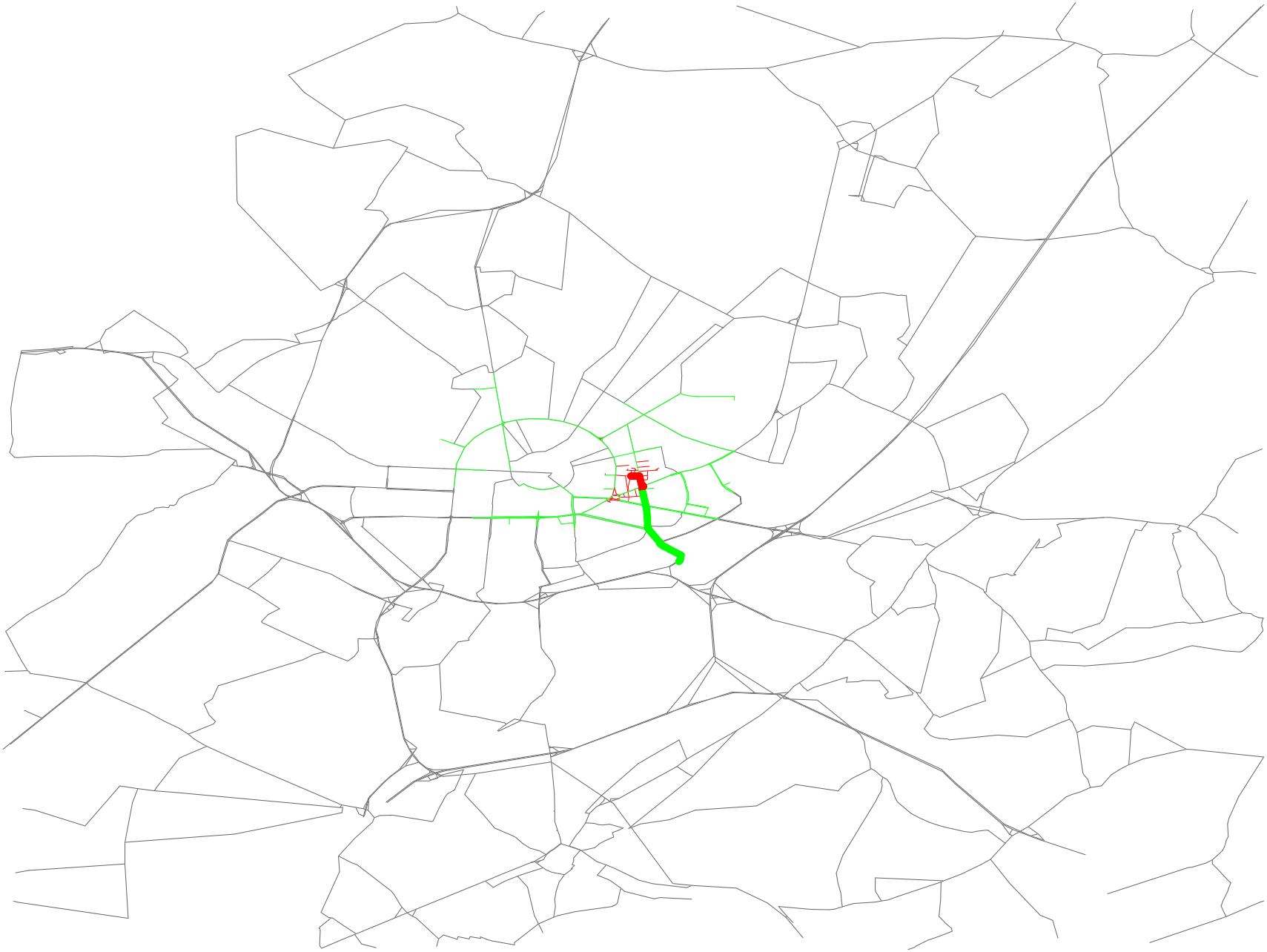
Anzeigehöhe: 20 km Level 0 Suchraum

Anzeigehöhe: 20 km Level 1

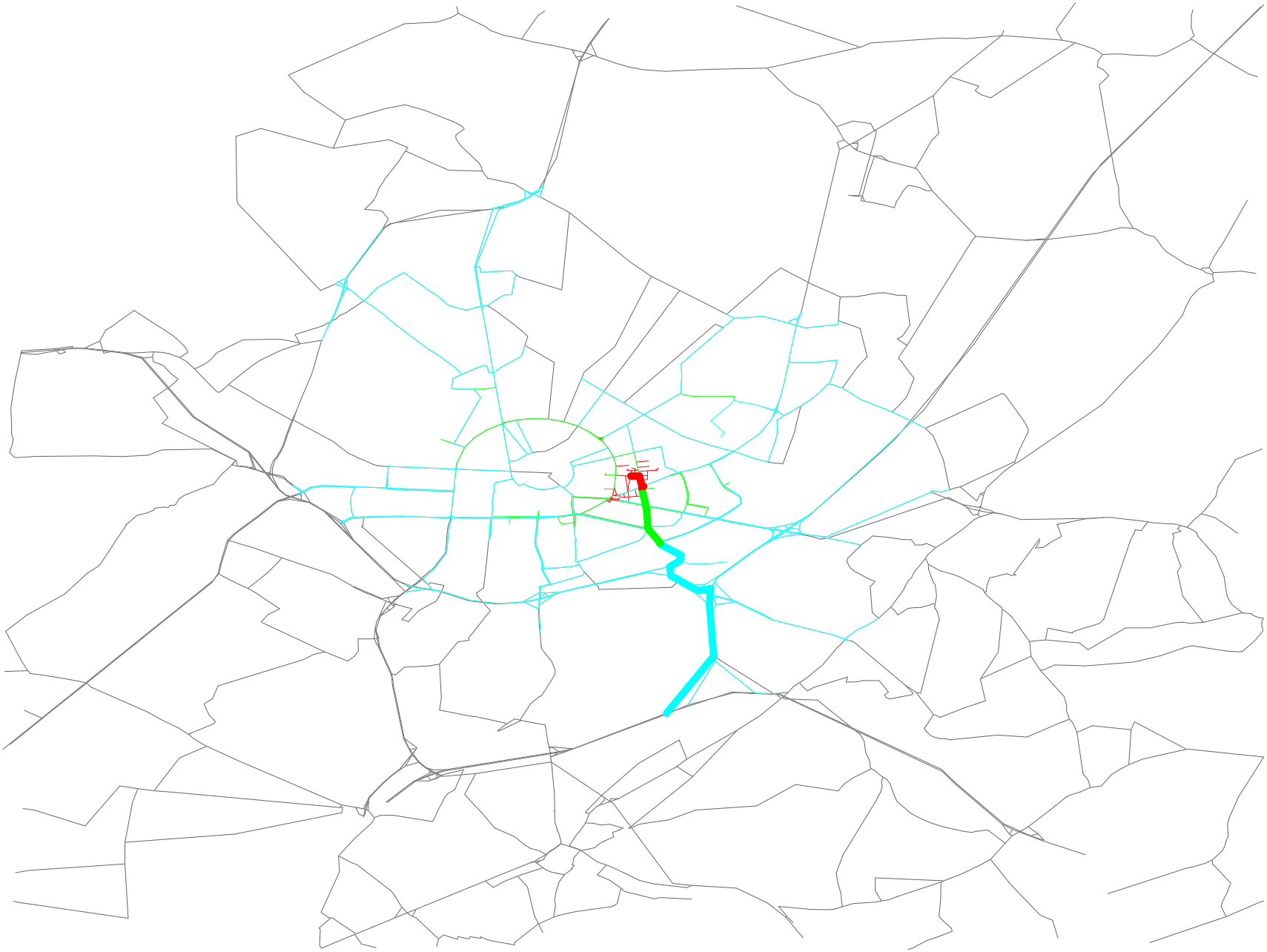
Anzeigehöhe: 20 km Level 1 Suchraum



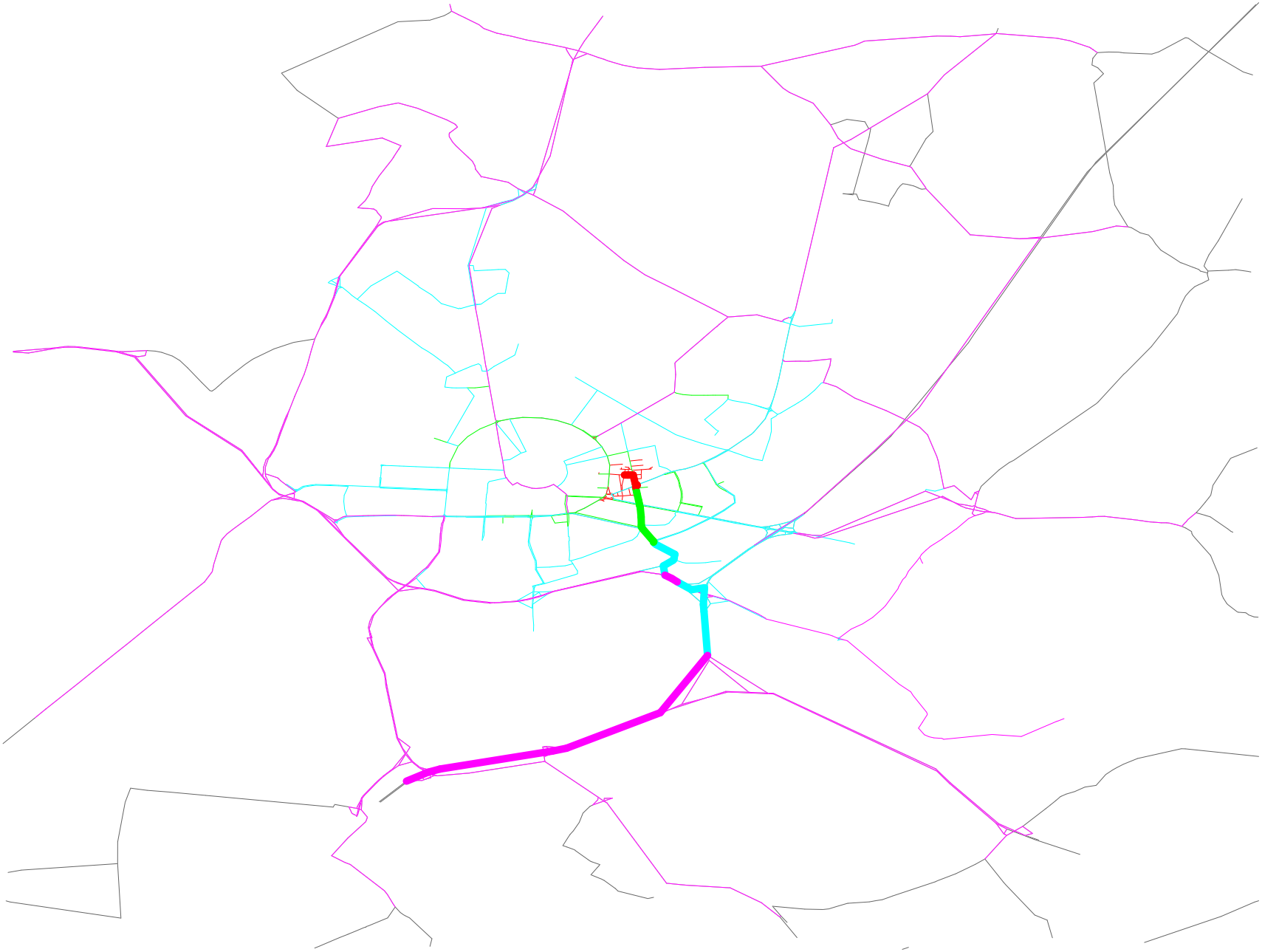
Anzeigehöhe: 20 km Level 2



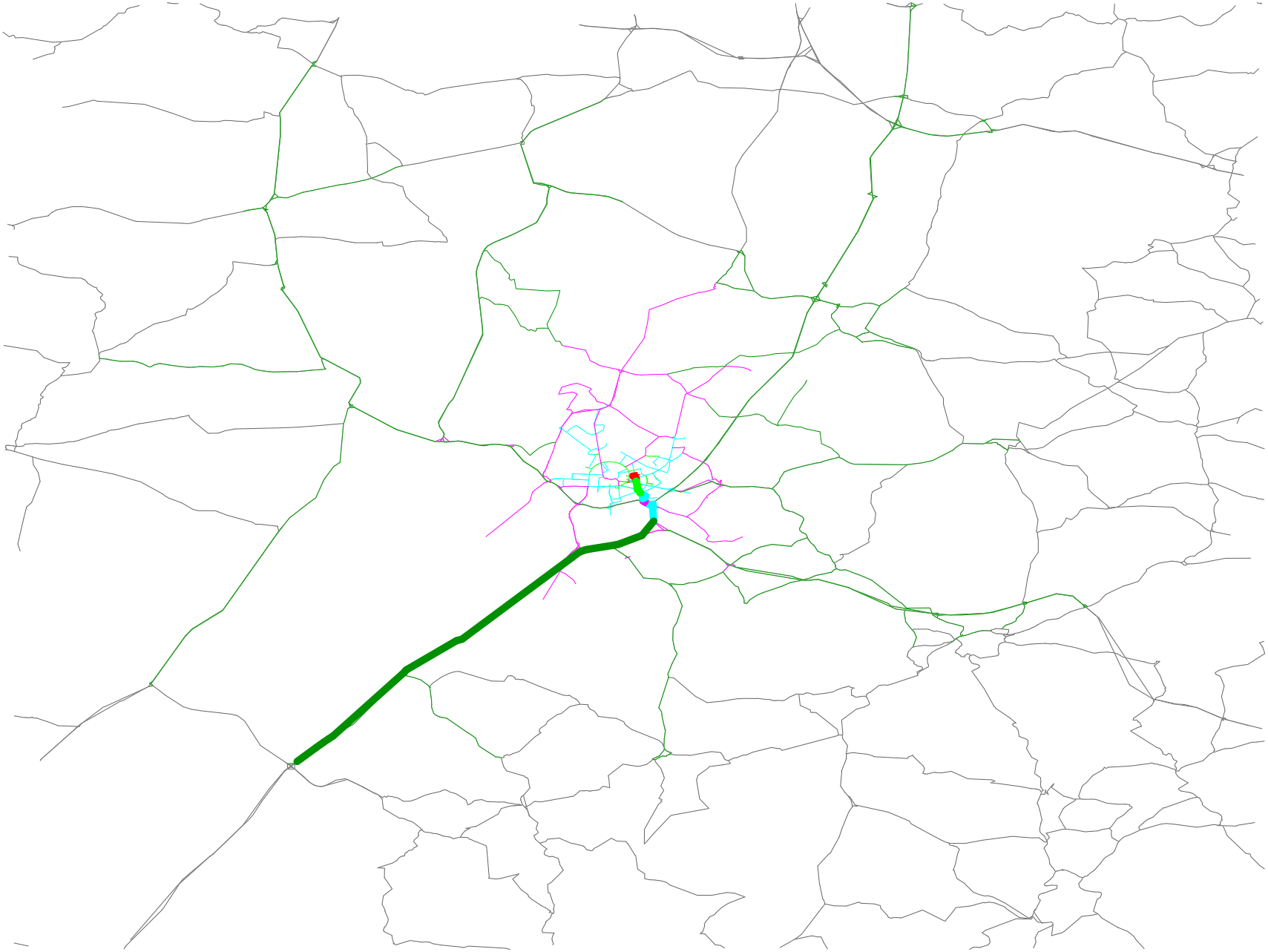
Anzeigehöhe: 20 km Level 2 Suchraum



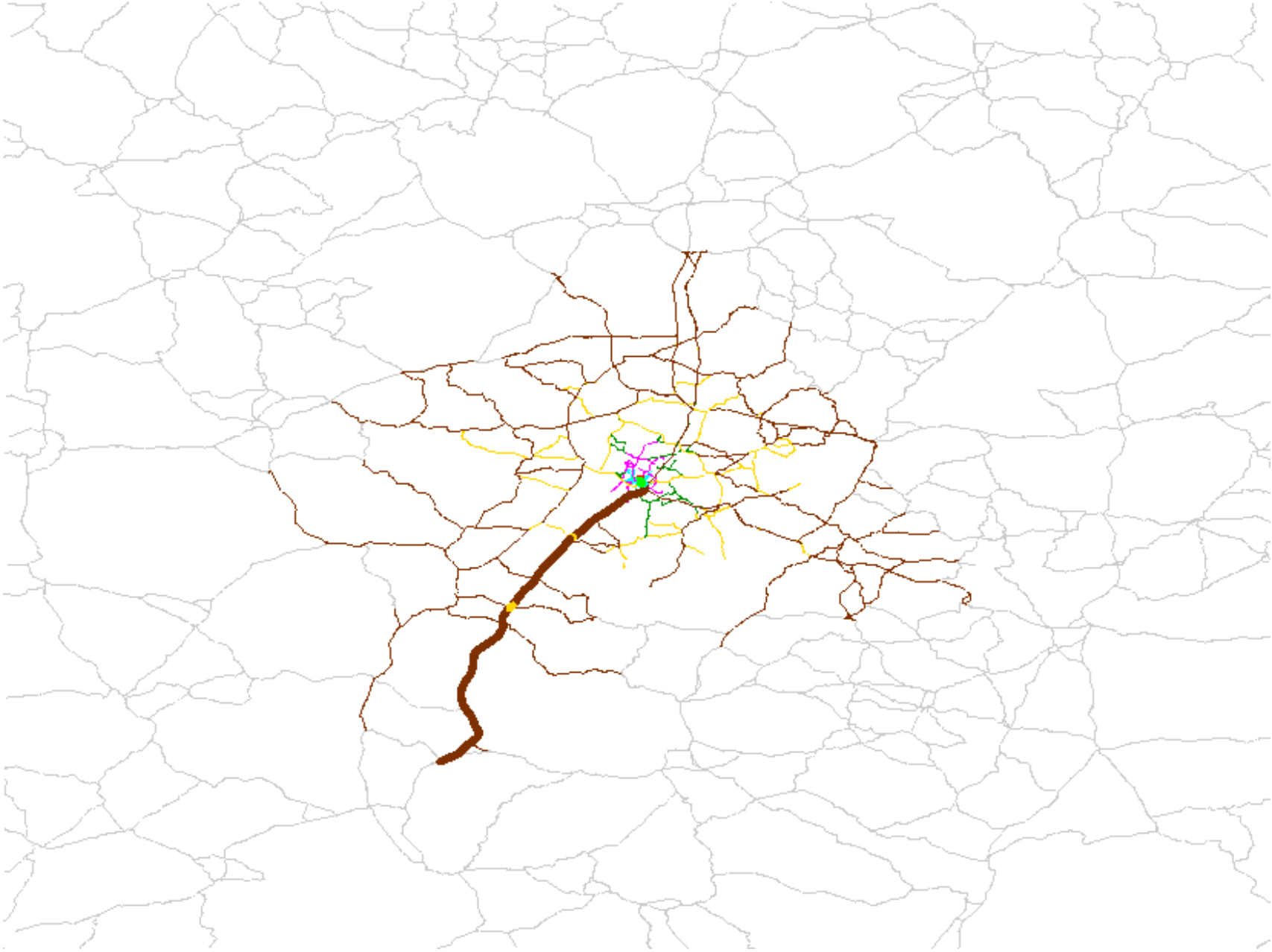
Anzeigehöhe: 20 km Level 3 Suchraum



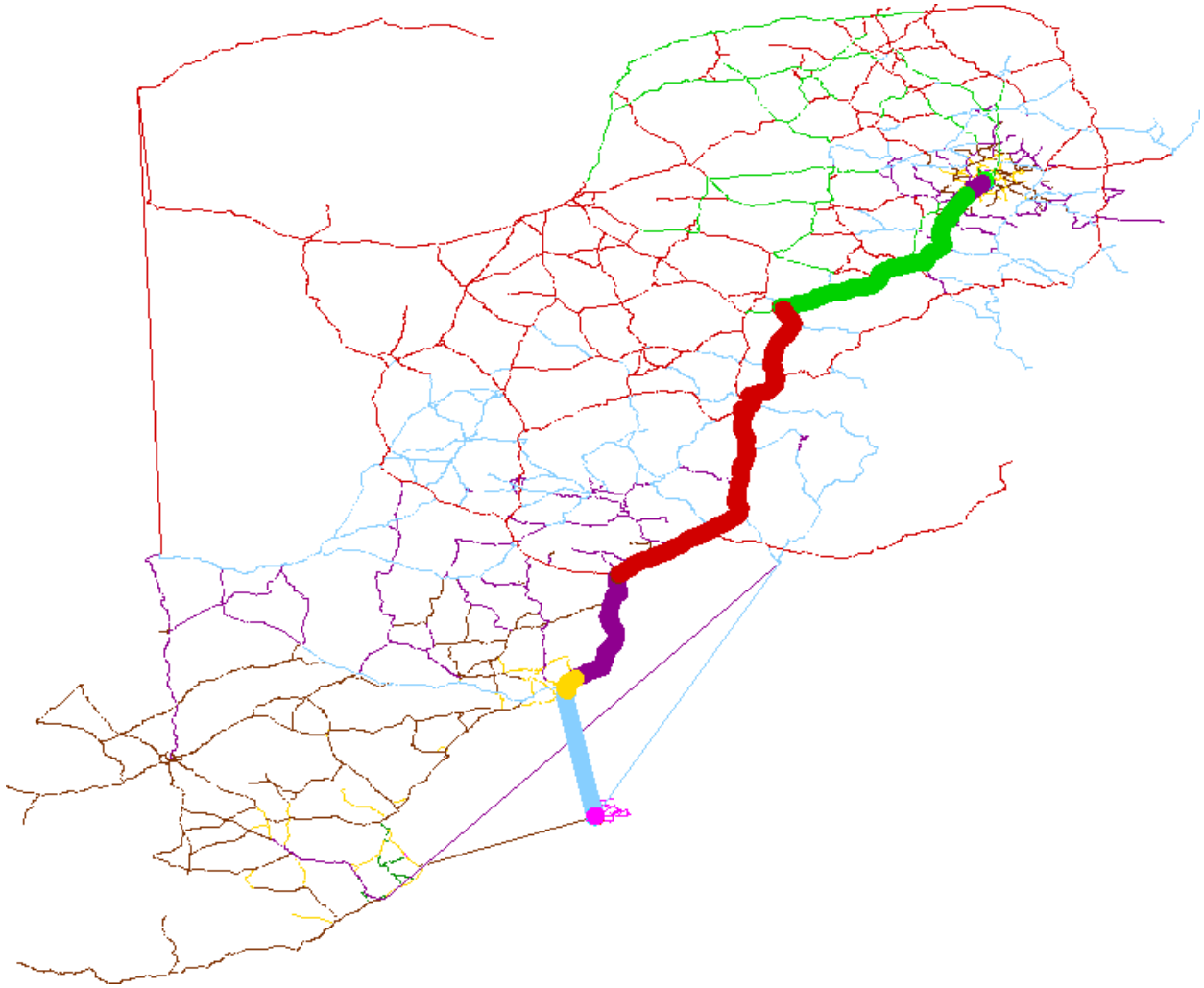
Anzeigehöhe: 80 km Level 4 Suchraum



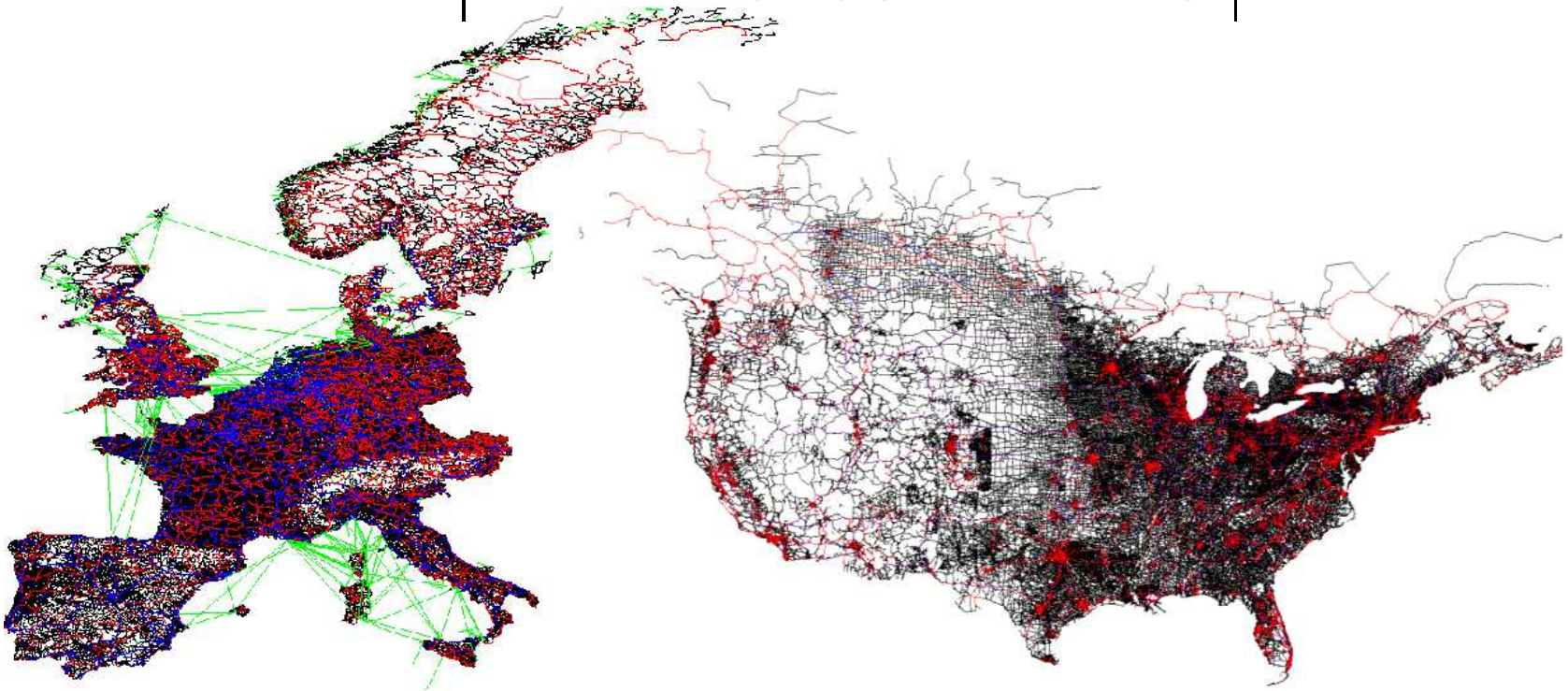
Anzeigehöhe: 400 km Level 6 Suchraum



Level 10 Suchraum



Westeuropa	**Testdaten**	USA/Kanada
18 029 721	#Knotenpunkte	18 741 705
42 199 587	#(gerichtete) Straßenabschnitte	47 244 849
15	Konstruktion [min]	20
0,76	Suchzeiten [ms]	0,90
8 320	Beschleunigung ($\leftrightarrow$ DIJKSTRA)	7 232



Naive Highspeed Variante

1. Suche nächste sinnvolle Autobahnauffahrt

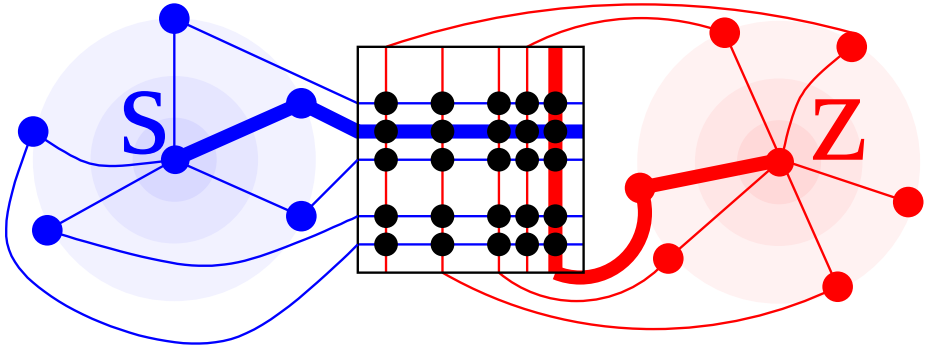
   $\rightsquigarrow$ Vorberechnung: speichere alle benachbarten Auffahrten

   Suche: schlage alle benachbarten Auffahrten nach

2. Fahre auf Autobahnen möglichst nahe ans Ziel heran

   $\rightsquigarrow$ Vorberechnung: speichere die Abstände zwischen allen Autobahnauffahrten

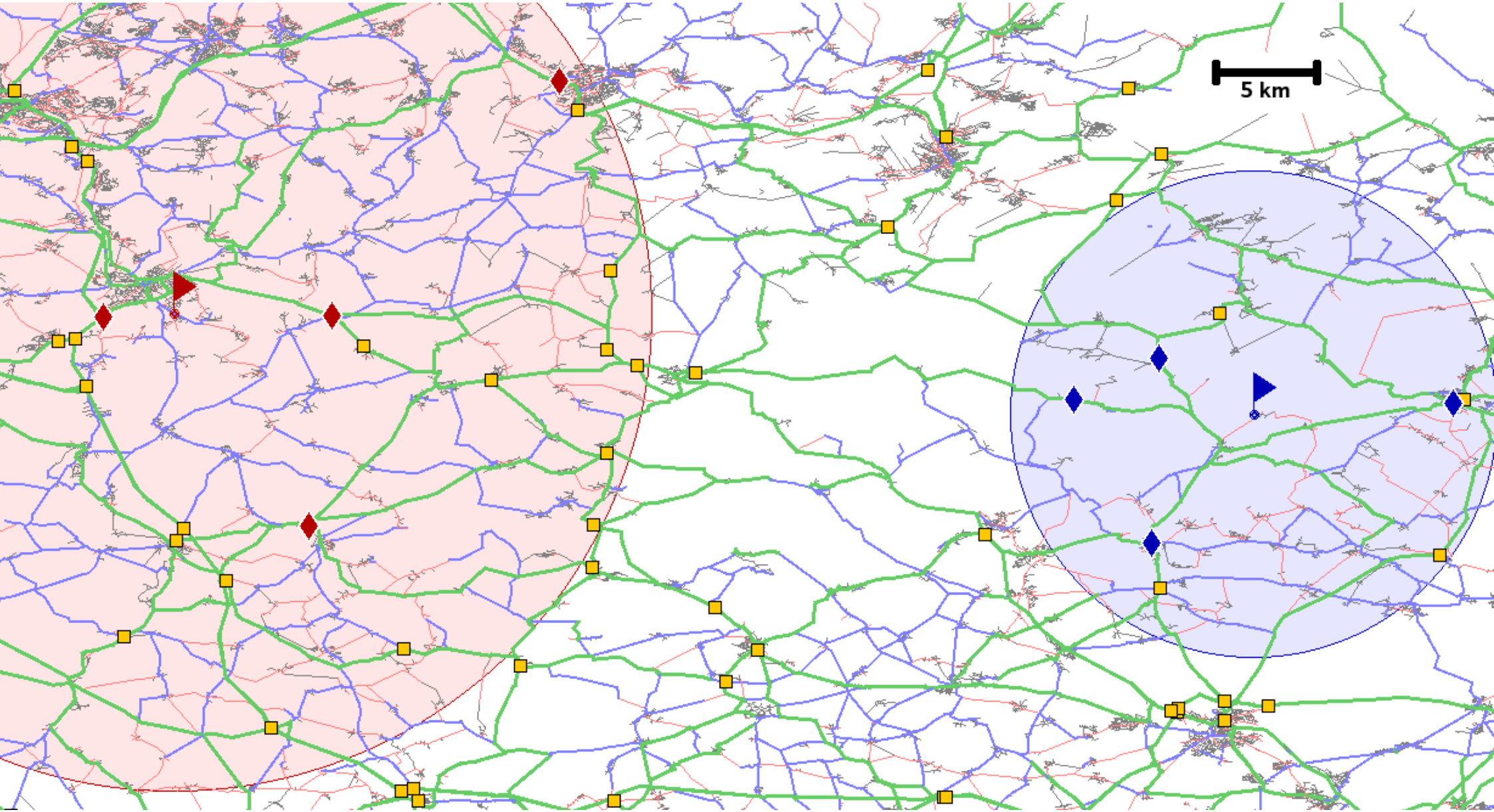
   Suche: schlage die Abstände nach



3. Suche von der Autobahnausfahrt den Weg zum Ziel

   $\rightsquigarrow$ analog zu 1.

Unsere Highspeed Variante



Zusammenfassung

- genaue Routen in großen Straßennetzen z.B. $\approx$ 42 Mio. Straßen
- schnelle Suche $\approx$ 1 ms / 10 μs
  $\rightsquigarrow$ Sieger beim DIMACS Wettbewerb (14. 11. 2006)
- schnelle Vorverarbeitung $\approx$ 15 min / 45 min
- geringer Speicherverbrauch
- naheliegende Verbesserung bestehender kommerzieller Lösungen

Ausblick

- schnelle, lokale Aktualisierung des Highway Netzwerks

  (z.B. bei Staus)

- Implementierung für Mobilgeräte

- Flexible, komplexere Zielfunktionen