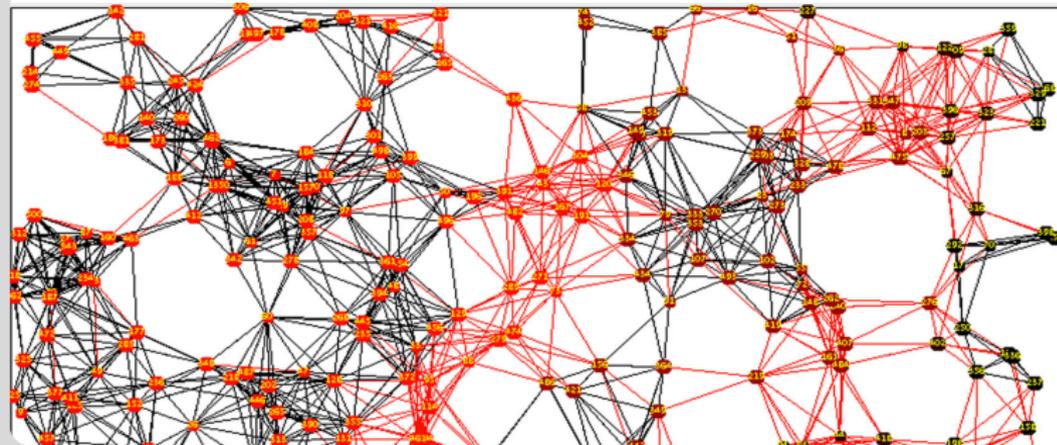


Algorithmen für Ad-hoc- und Sensornetze

Übung 3 – Topologiekontrolle und Routing

Fabian Fuchs | 24. November 2015 (Version 1)

INSTITUT FÜR THEORETISCHE INFORMATIK - LEHRSTUHL FÜR ALGORITHMIK (PROF. WAGNER)



Simulation Control

Round: 7

Rounds to do:

Refresh rate:

View

Output

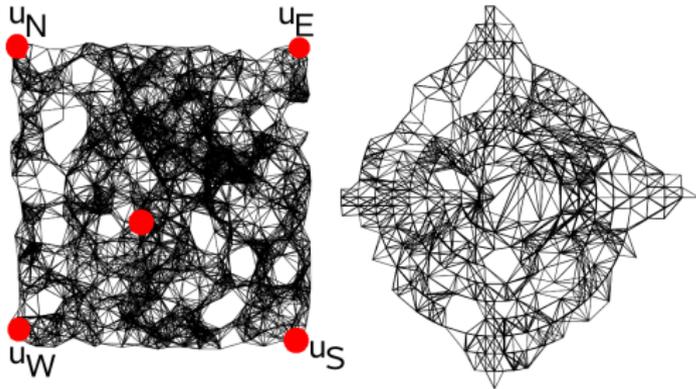
- Nachtrag: Heuristiken zur ankerfreien Lokalisierung
- Besprechung: Übung 2 - Greedy Routing
- Übung 3



Eine Heuristik: AFL (Initiallösung)

- 1 Finde vier extreme und einen zentralen Knoten
 - U_N, U_W, U_S, U_E, U_C
- 2 Bestimme Hop-Entfernungen zu diesen Knoten
- 3 Bette Knoten anhand von Polarkoordinaten ein

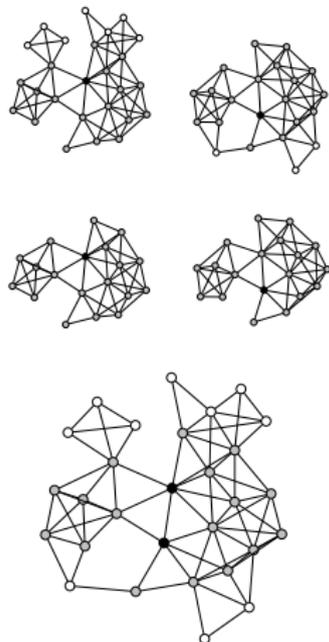
$$\rho_v = d_G(v, u_C) \quad \theta_v = \arctan \frac{d_G(v, U_N) - d_G(v, U_S)}{d_G(v, U_W) - d_G(v, U_E)}$$



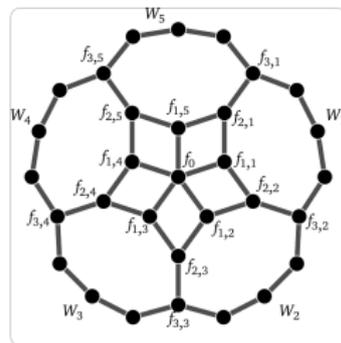
Funktioniert nicht so gut,
wenn das Gebiet komplexer
wird!

- Dichte Netze erlauben gute lokale Lösungen
 - viele lokale Lösungen zu berechnen skaliert gut!
 - setze Lösungen iterativ oder hierarchisch zusammen

- Erkennen von inneren Knoten und Randknoten
 - Ausnutzen von lokalen Strukturen
 - Ausnutzen von statistischen Eigenschaften



- Dichte Netze erlauben gute lokale Lösungen
 - viele lokale Lösungen zu berechnen skaliert gut!
 - setze Lösungen iterativ oder hierarchisch zusammen
- Erkennen von inneren Knoten und Randknoten
 - Ausnutzen von lokalen Strukturen
 - Ausnutzen von statistischen Eigenschaften



- Ankerbasierte Lokalisierung
 - fast ausschließlich Schätztheorie oder Heuristiken
 - kaum algorithmische Analyse (wegen unklarer Parameter?)
 - *Ausnahme*: Hop/HS zu eindimensionaler Lokalisierung

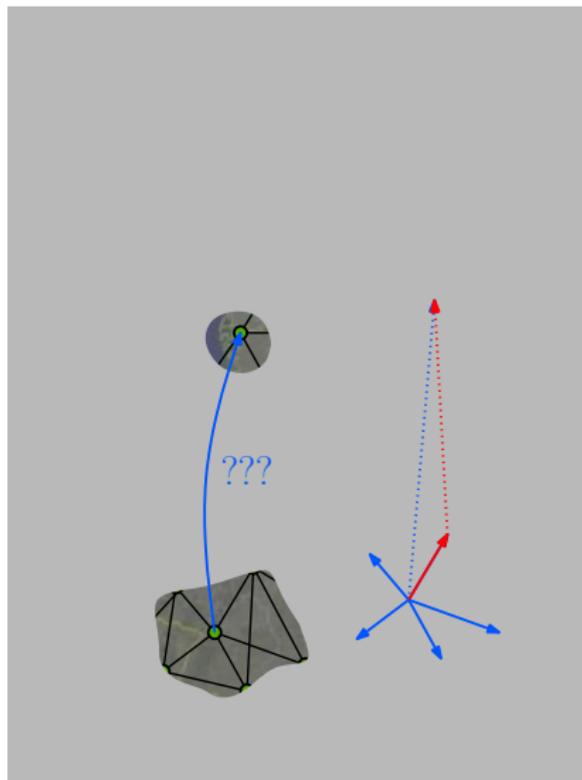
- Ankerfreie Lokalisierung
 - klarere algorithmische Probleme
 - vor allem Negativergebnisse
 - Heuristiken für dichte Graphen, aber ohne Garantien

- Nachtrag: Heuristiken zur ankerfreien Lokalisierung
- Besprechung: Übung 2 - Greedy Routing
- Übung 3

Greedy Routing

Leite Paket an den Nachbarn, der dem Ziel am nächsten ist.

Was, wenn es nicht weitergeht?

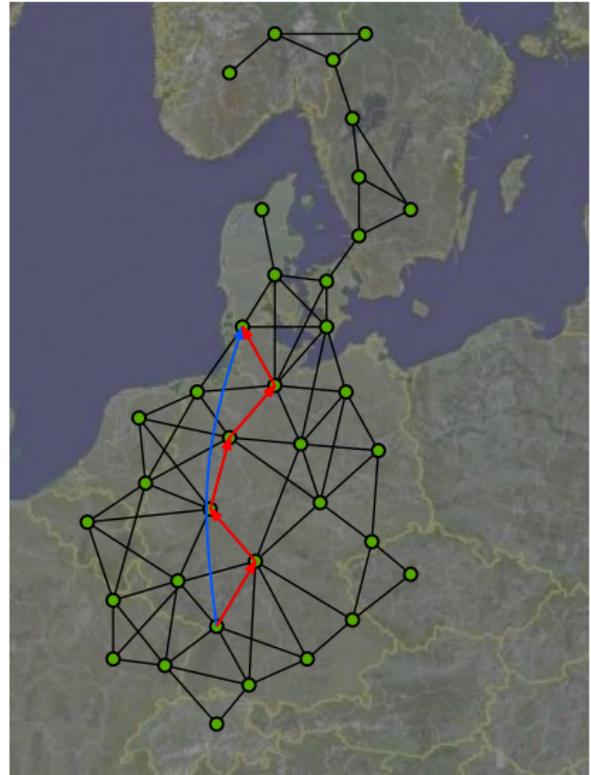


Greedy Routing

Greedy Routing

Leite Paket an den Nachbarn, der dem Ziel am nächsten ist.

Was, wenn es nicht weitergeht?

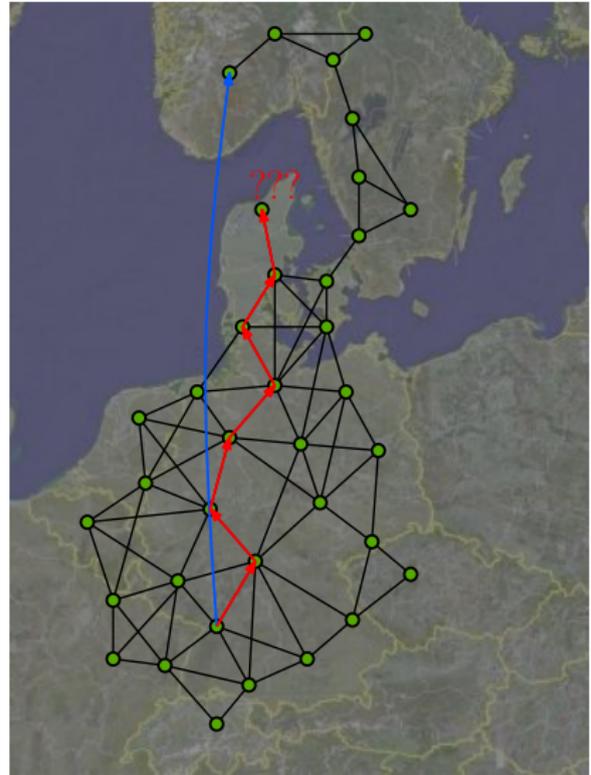


Greedy Routing

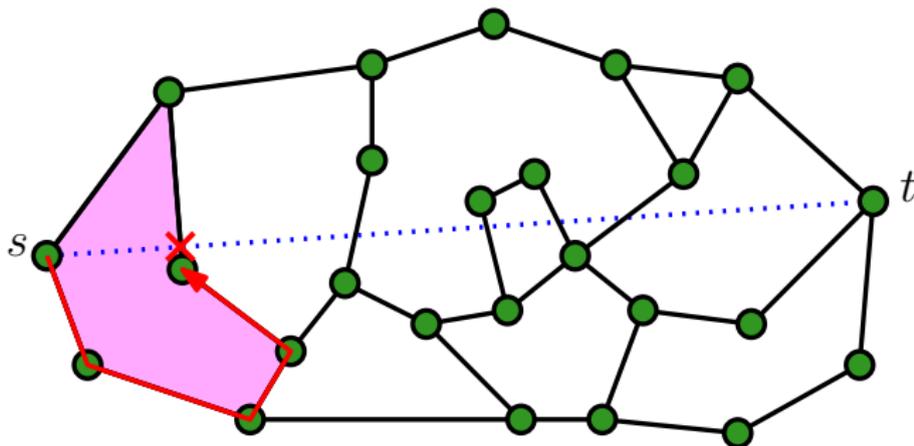
Greedy Routing

Leite Paket an den Nachbarn, der dem Ziel am nächsten ist.

Was, wenn es nicht weitergeht?

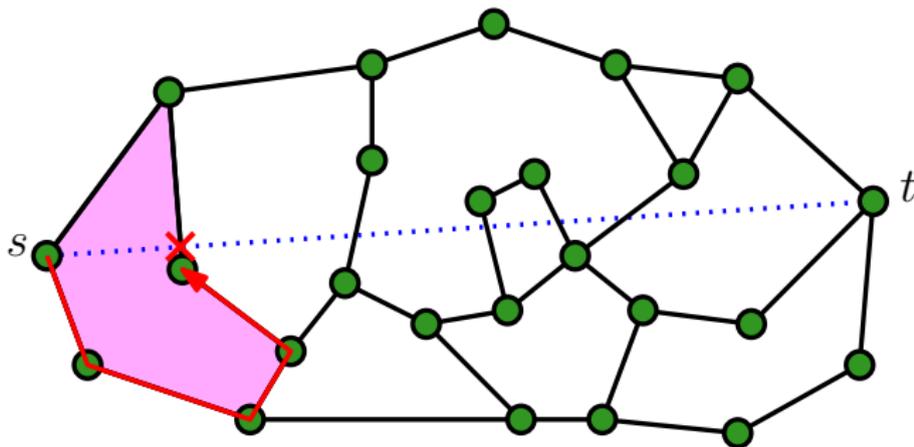


- Nachtrag: Heuristiken zur ankerfreien Lokalisierung
- Besprechung: Übung 2 - Greedy Routing
- Übung 3



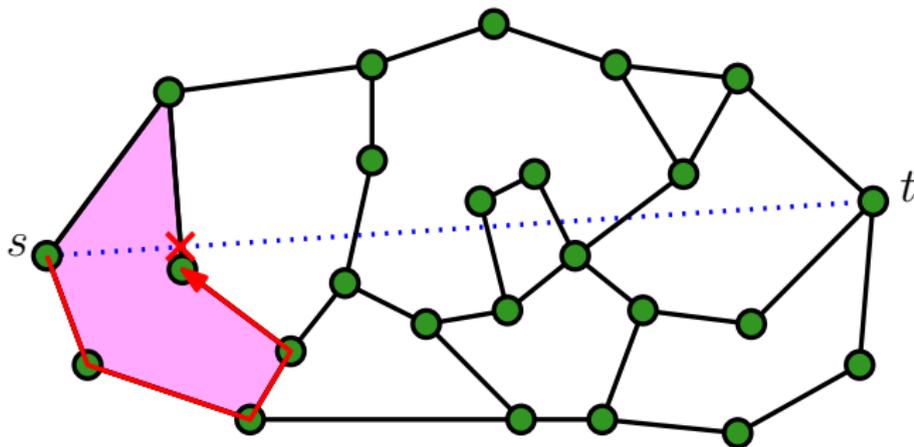
Herkömmliches Facettenrouting

- Erkunde Facette, in die \vec{st} zeigt
- Kehre zu Schnitt mit \vec{st} zurück, der t am nächsten ist
- Wechsle zur angrenzenden Facette, in die \vec{pt} zeigt.



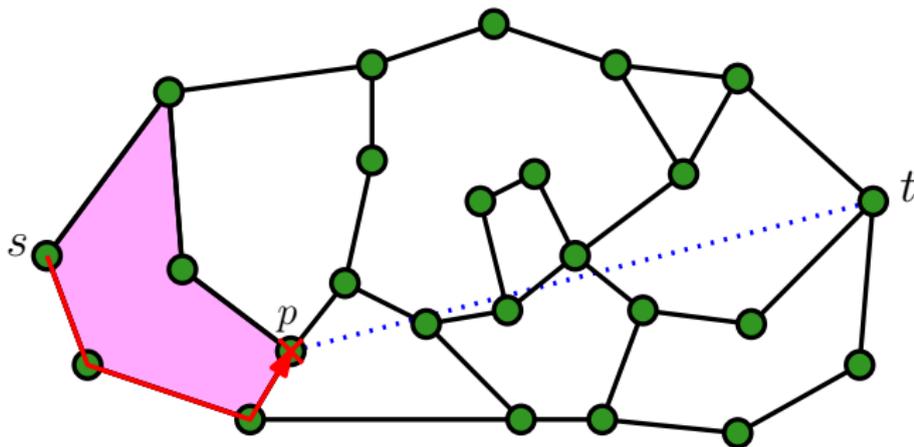
Optimiertes Facettenrouting [Wattenhofer et al. '03]

- Erkunde Facette, in die \vec{st} zeigt
- Kehre zum Punkt p zurück, der t am nächsten ist
- Wechsle zur angrenzenden Facette, in die \vec{pt} zeigt.



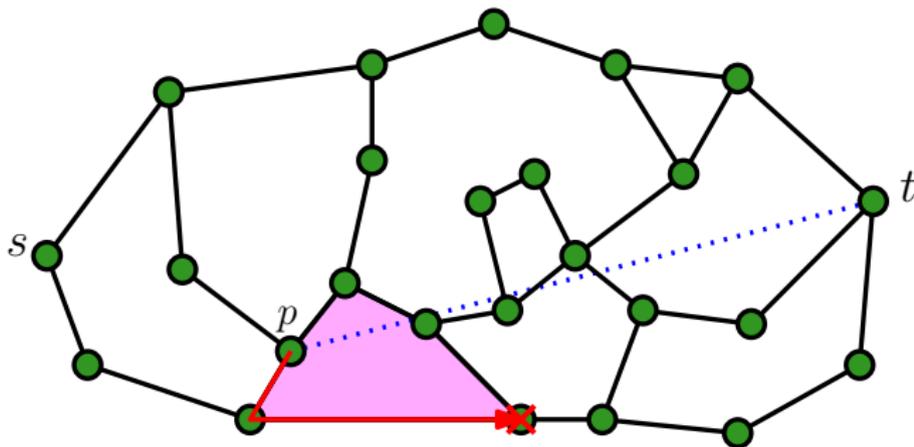
Optimiertes Facettenrouting in GG [Wattenhofer et al. '03]

- Erkunde Facette, in die \vec{st} zeigt
- Kehre zum **Knoten** p zurück, der t am nächsten ist (ohne Bew.)
- Wechsle zur angrenzenden Facette, in die \vec{pt} zeigt.



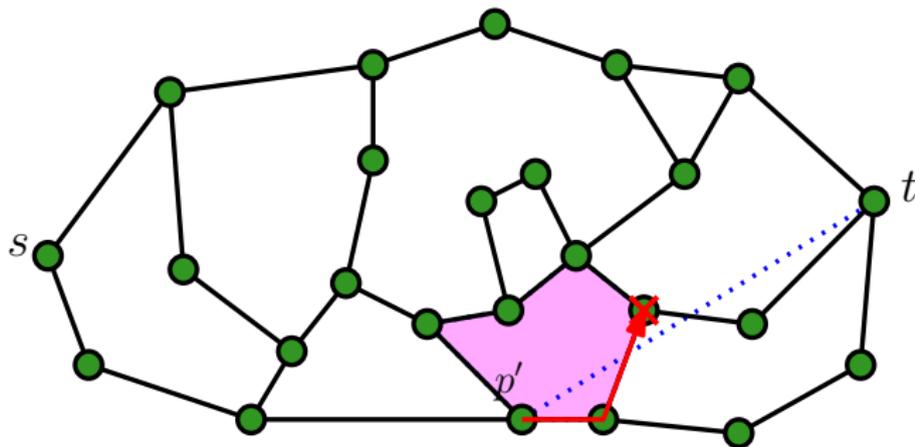
Optimiertes Facettenrouting in GG [Wattenhofer et al. '03]

- Erkunde Facette, in die \vec{st} zeigt
- Kehre zum **Knoten** p zurück, der t am nächsten ist (ohne Bew.)
- Wechsle zur angrenzenden Facette, in die \vec{pt} zeigt.



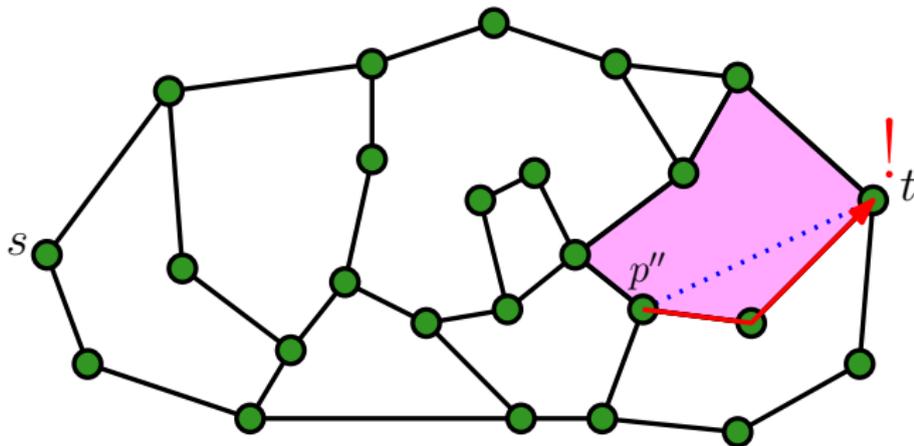
Optimiertes Facettenrouting in GG [Wattenhofer et al. '03]

- Erkunde Facette, in die \vec{st} zeigt
- Kehre zum **Knoten** p zurück, der t am nächsten ist (ohne Bew.)
- Wechsle zur angrenzenden Facette, in die \vec{pt} zeigt.



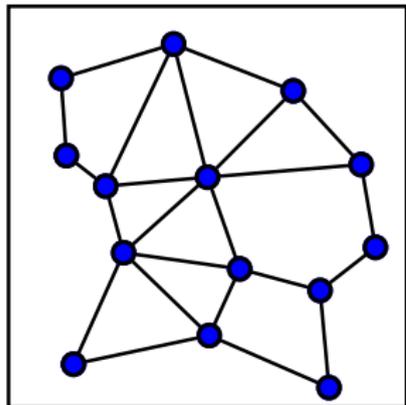
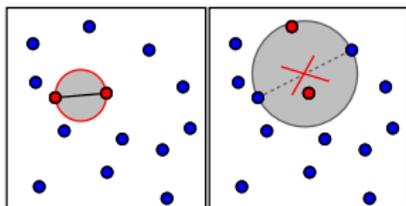
Optimiertes Facettenrouting in GG [Wattenhofer et al. '03]

- Erkunde Facette, in die \vec{st} zeigt
- Kehre zum **Knoten** p zurück, der t am nächsten ist (ohne Bew.)
- Wechsle zur angrenzenden Facette, in die \vec{pt} zeigt.



Optimiertes Facettenrouting in GG [Wattenhofer et al. '03]

- Erkunde Facette, in die \vec{st} zeigt
- Kehre zum **Knoten** p zurück, der t am nächsten ist (ohne Bew.)
- Wechsle zur angrenzenden Facette, in die \vec{pt} zeigt.



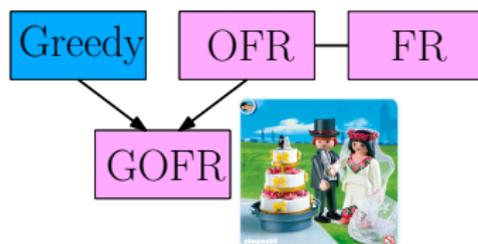
Gabriel Graph

Eine Kante ist im GG gdw. der Umkreis der Kante keine weiteren Knoten enthält.

GG von UDG:

- einfach lokal zu entscheiden
- planar, zshg., Durchschnittsgrad?
- Maximalgrad $n - 1!$ (Warum?)
- Entfernungsspannfaktor $O(\sqrt{n})$ (o.B.)
- **Energiespannfaktor 1!** (für $\alpha \geq 2$)

- Routinginformation komplett in Nachricht enthalten
 - Start- und Zielkoordinaten
 - bester Übergangspunkt zur nächsten Facette
 - letzte Knotenposition (Berechnung des nächsten Knotens)
- Vollständig lokal
 - Positionen der Nachbarknoten reichen aus
 - Facettenerkundung nur implizit durch Linke/Rechte-Hand-Regel



Ablauf:

- 1 Greedy-Routing bis es stockt, dann
- 2 ORF bis zum besten Knoten der Facette
- 3 Dann weiter per Greedy-Routing

- Übungsblatt auf VL-Homepage:
`http://i11www.iti.uni-karlsruhe.de/teaching/winter2015/sensornetze/index`
- Sinalgo Tutorial: `http://disco.ethz.ch/projects/sinalgo/tutorial/Documentation.html`
- Sinalgo Framework: VL-Homepage oder Sinalgo Tutorial Seite
- Tipp: Wer schon mit Arduino rumspielen möchte:
`https://123d.circuits.io/`