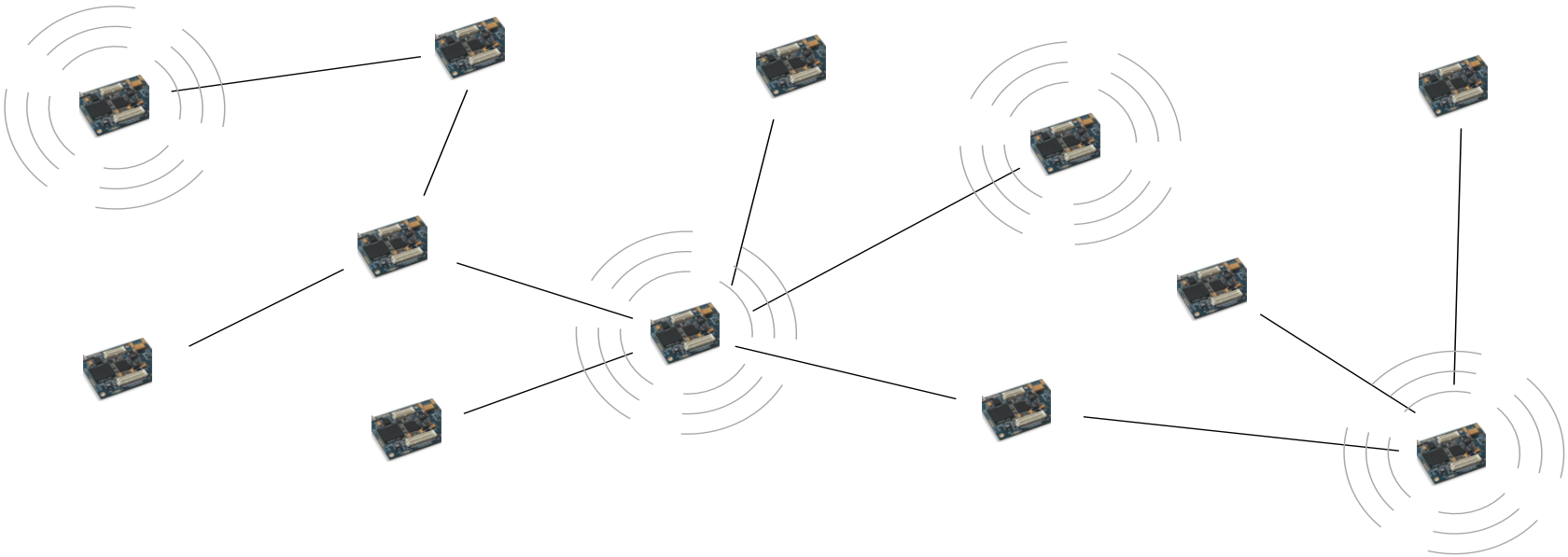
Karlsruher Institut für Technologie

Algorithmen für Ad-hoc- und Sensornetze

VL 02 – Geographisches Routing

Roman Prutkin | 22. Oktober 2015 (Version 1)

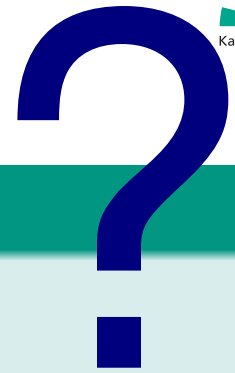
INSTITUT FÜR THEORETISCHE INFORMATIK - LEHRSTUHL FÜR ALGORITHMIK (PROF. WAGNER)



Überblick

- Ergänzungen zur VL 1: Einführung und Beispiele
  - Leader Election in allgemeinen Graphen
  - Nachtrag Full Link Reversal
  - Dynamisches Full Link Reversal
- Geographisches Routing
  - Einschub: Drahtlose Kommunikation
  - Unit-Disk-Graphen und Quasi-Unit-Disk-Graphen
  - Greedy Routing
  - Gabriel Graph
  - Facettenrouting (FR)
  - Beschränktes Facettenrouting (BFR)
  - Das $\Omega(1)$-Modell
  - Adaptives Facettenrouting (AFR)
  - Greedy adaptives Facettenrouting (GAFR)
  - Optimiertes Facettenrouting (OFR)
  - Greedy optimiertes Facettenrouting (GOFR)
  - Optimiertes adaptives Facettenrouting (OAFR)
  - Greedy optimiertes adaptives Facettenrouting (GOAFR)

Leader Election - Knobelaufgabe



Erinnerung: Leader Election

Gegeben: Symmetrischer, zusammenhängender Kommunikationsgraph G.

Problem: Genau ein Prozessor soll als *Leader* ausgezeichnet werden.

Erinnerung: Idee Leader Election in Ringen

Jeder Knoten schickt seine ID nach links und reicht höhere IDs weiter. Wer seine eigene ID hört, schickt Terminierungsnachricht in den Ring und terminiert als Leader.

Wie könnte ein Leader-Election-Algorithmus für allgemeine Graphen aussehen?

Leader Election in allg. Graphen

Leader Election, Knoten p_i kennt $\texttt{ID}_i$

sende $\texttt{ID}_i$ an alle Nachbarn
setze $\texttt{ID}_+ \leftarrow \texttt{ID}_i$ und $\texttt{parent} \leftarrow \perp$
wenn $\texttt{ID}$*s empfangen wurden* **dann**
  wähle maximale empfangene $\texttt{ID}$ und zugehörigen Sender s
  wenn $\texttt{ID} > \texttt{ID}_+$ **dann**
    setze $\texttt{ID}_+ \leftarrow \texttt{ID}$ und $\texttt{parent} \leftarrow s$
    sende Nachricht „new follower“ an $\texttt{parent}$
    sende $\texttt{ID}_+$ an restliche Nachbarn

wenn *„new follower“ empfangen wurde und* $\texttt{parent} \neq \perp$ **dann**
  sende Nachricht „new follower“ an $\texttt{parent}$[a]

[a] außer in selber Runde schon geschehen

Wann darf sich ein Knoten mit $\texttt{parent} = \perp$ zum Leader erklären?
Wenn zwei Runden keine follower-Nachricht kam! (Ohne Beweis)

Korrektheit Full Link Reversal

Satz

Full Link Reversal führt $O(n^2)$ Umorientierungen durch und terminiert nach $O(n^2)$ Schritten in einem t-zielorientierten DAG.

Beweis:

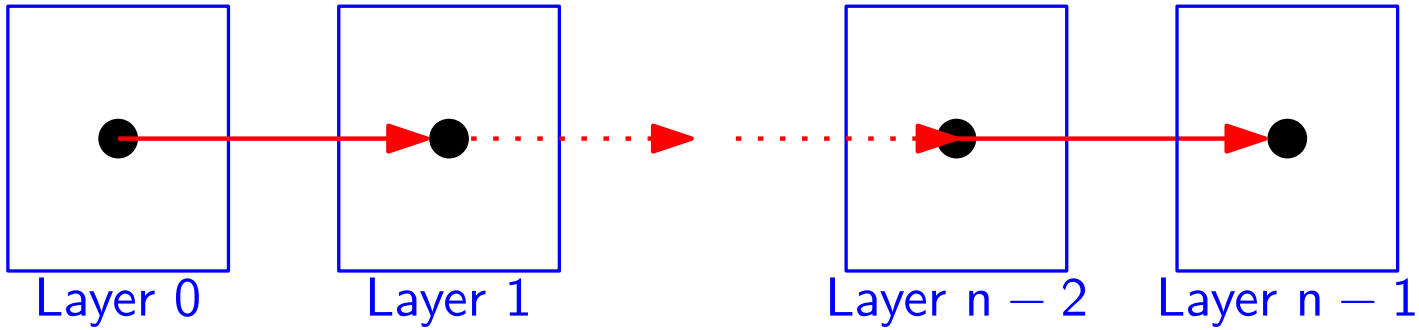
- in jedem Schritt ist der Graph ein DAG
- kein Knoten kann in einem Layer $> n$ starten
- in jedem Schritt verringert mindestens ein Knoten sein Layer
- nach spätestens n^2 Schritten sind alle Knoten in Layer 0
- wenn alle Knoten in Layer 0 sind, ist der DAG t-zielorientiert

Worst-Case-Netzwerke 1

Satz

Es gibt Initiallösungen auf Graphen bei denen der Full-Reversal-Algorithmus insgesamt $\Theta(n^2)$ Umorientierungen ausführt.

- sogar ganz einfache: falsch orientierte Pfade



- $\sum_{i=1}^{n}(i-1) \in \Theta(n^2)$ Umorientierungen, da ein Knoten in Layer i exakt i mal umorientiert werden muss

$\Rightarrow$ Schranke für die Zahl der Umorientierungen scharf!

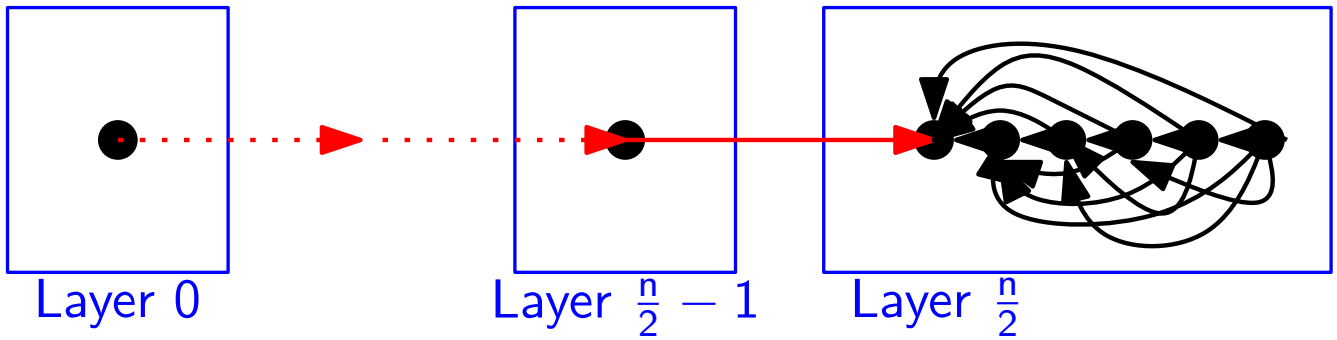
- Und die Laufzeitschranke?
  - In diesem Beispiel nur $\Theta(n)$ Schritte

Worst-Case-Netzwerke 2

Satz

Es gibt Initiallösungen auf Graphen bei denen der Full-Reversal-Algorithmus insgesamt $\Theta(n^2)$ Schritte benötigt.

- etwas komplexere:



- vollständiger DAG auf $n/2$ Knoten in Layer $n/2$
- jeder davon braucht $n/2$ Umorientierungen
- immer nur einer kann zur Zeit Senke sein

$\Rightarrow$ Laufzeit $\Omega(n^2)$, Laufzeitschranke $O(n^2)$ damit auch scharf!

Dynamisches Full Link Reversal

Erinnerung: Full Link Reversal

- Richte alle Kanten von größerem Label (ID) zu kleinerem
- Wenn ein Knoten $v \neq t$ keine ausgehenden Kanten hat
  - drehe alle inzidenten Kanten um
  - wähle neues Label größer als bei allen Nachbarn

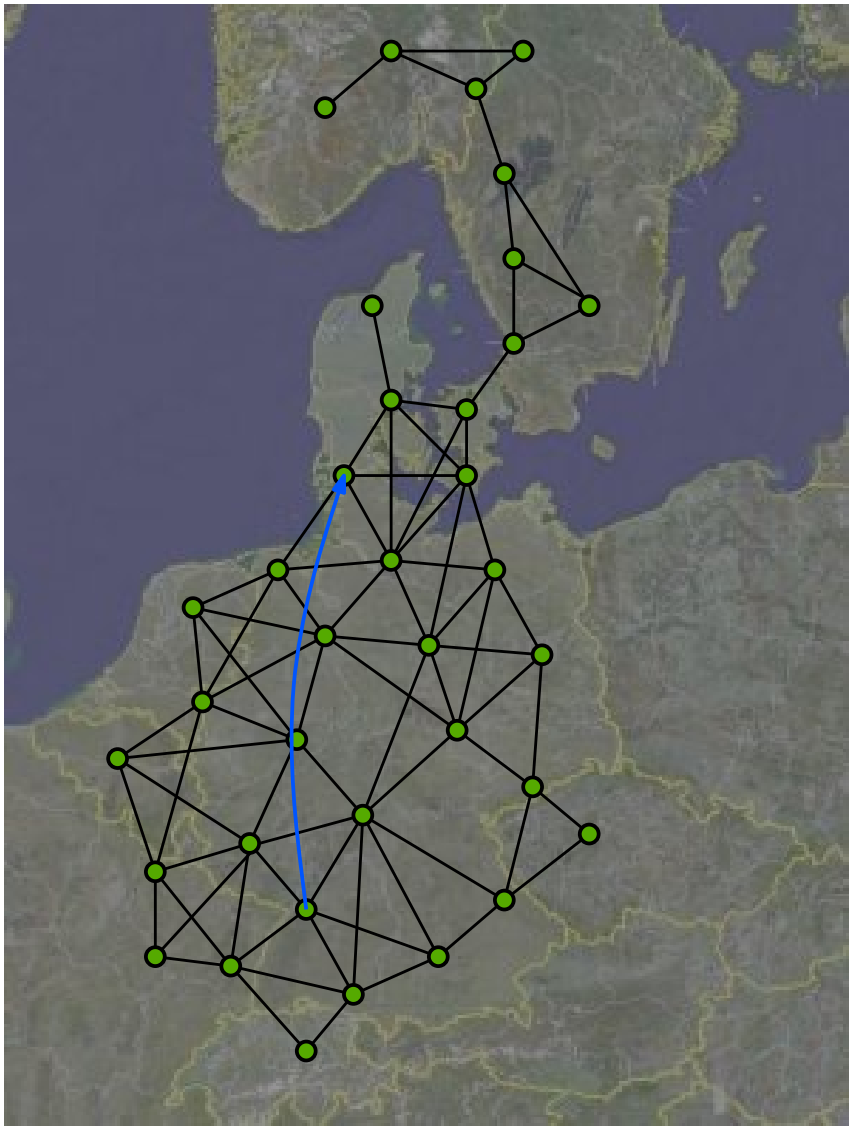
Graphen azyklisch und t-gerichtet halten:

- Kanten oder Knoten fallen weg
  - entstehende Senken werden durch Link Reversal behoben
- neuer Knoten u
  - alle Kanten von u weg richten
  - wähle Label größer als bei allen Nachbarn
- neue Kante entsteht
  - falsche Richtung könnte Kreis induzieren.
  - richte von größerem Label zu kleinerem, bei Gleichstand betrachte ursprüngliche ID

Überblick

- Ergänzungen zur VL 1: Einführung und Beispiele
  - Leader Election in allgemeinen Graphen
  - Nachtrag Full Link Reversal
  - Dynamisches Full Link Reversal
- Geographisches Routing
  - Einschub: Drahtlose Kommunikation
  - Unit-Disk-Graphen und Quasi-Unit-Disk-Graphen
  - Greedy Routing
  - Gabriel Graph
  - Facettenrouting (FR)
  - Beschränktes Facettenrouting (BFR)
  - Das $\Omega(1)$-Modell
  - Adaptives Facettenrouting (AFR)
  - Greedy adaptives Facettenrouting (GAFR)
  - Optimiertes Facettenrouting (OFR)
  - Greedy optimiertes Facettenrouting (GOFR)
  - Optimiertes adaptives Facettenrouting (OAFR)
  - Greedy optimiertes adaptives Facettenrouting (GOAFR)

Geographisches Routing



Geographisches Routing

Geographisches Routing

- Knoten kennen ihre Geokoordinaten (und die ihrer Nachbarn)
- Pakete enthalten Start- und Zielkoordinaten
- Vollkommen reaktiv: keine Routingtabellen

- Geographisches Routing ist realistisch
  - Knotenpositionen bekannt durch GPS/Galileo oder Lokalisierungsverfahren
  - Zielpositionen bekannt
    - wenn statt Labels Positionen verwendet werden (statisch)
    - wenn Informationen an Zielkoordinaten geroutet werden
    - Beispiel: Location Services (VL03)

Positionsbewusstsein

Definition

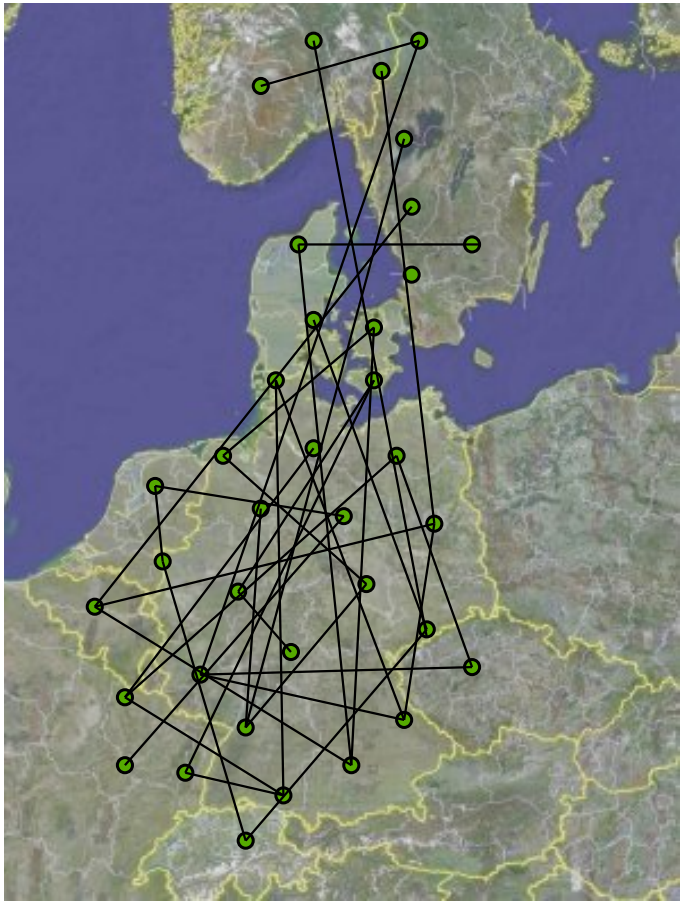
Eine Einbettung eines Graphen $G = (V, E)$ ist eine Abbildung $\mathbf{p} : V \to \mathbb{R}^d$.

- Sensornetze bringen ihre Einbettung gleich mit!
- zunächst nur in der Ebene: $d = 2$

Definition

Ein verteilter Algorithmus auf einem eingebetteten Graphen heißt *positionsbewusst*, wenn jeder Knoten v seine Position $\mathbf{p}(v)$ und die seiner Nachbarn kennt.

- zunächst nur statisch: $\mathbf{p}$ ändert sich nicht



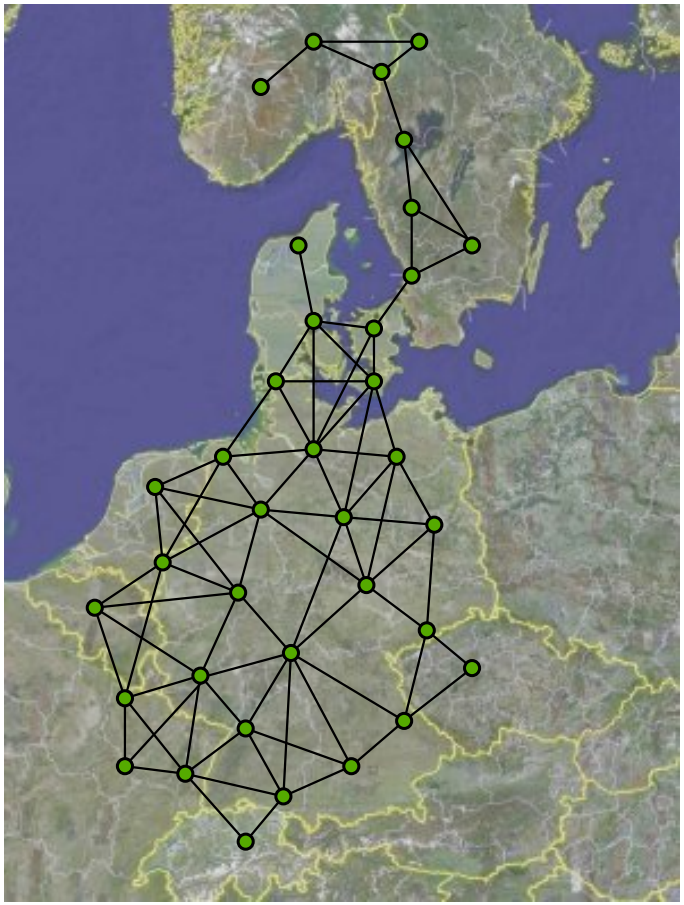
Was sind Koordinaten wert, wenn die Vernetzung nichts mit ihnen zu tun hat?

Drahtlose Kommunikation

- Signal fällt mit Entfernung ab
- Empfänger braucht gewisse Signalstärke

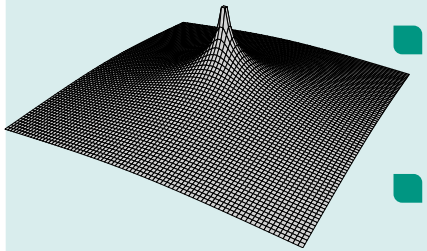
Verbindungen

beliebige Graphen	???	???



Was sind Koordinaten wert, wenn die Vernetzung nichts mit ihnen zu tun hat?

Drahtlose Kommunikation



- Signal fällt mit Entfernung ab
- Empfänger braucht gewisse Signalstärke

Verbindungen

beliebige Graphen ??? UDG

Einschub: Drahtlose Kommunikation

Grundlagen Signalausbreitung

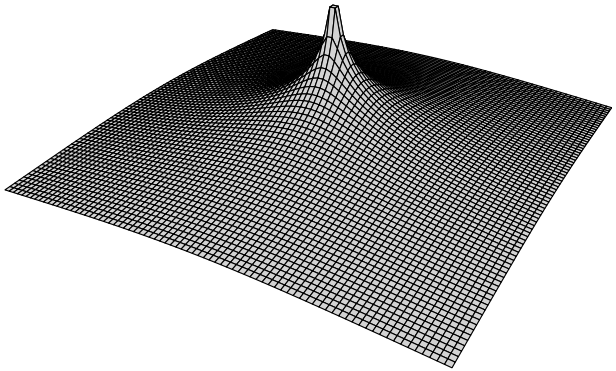
- gemeinsames Medium
- Signalabfall
  - „free space“

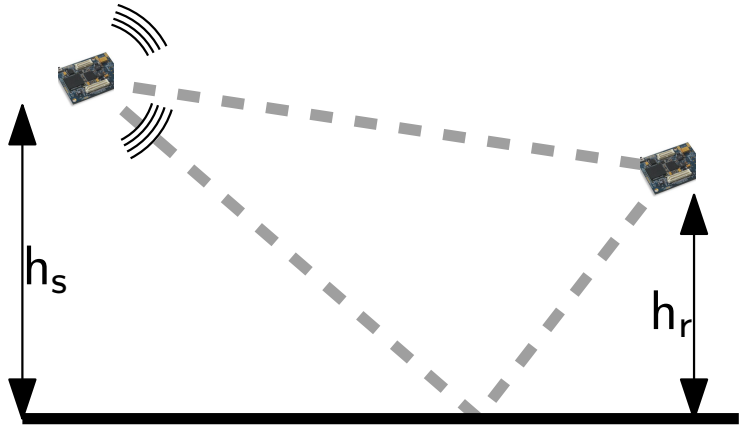
$$P_r = \frac{P_s G_s G_r \lambda^2}{(4\pi)^2 L d(r,s)^2}$$

  - „two ray ground“

$$P_r \sim \frac{P_s G_s G_r h_s^2 h_r^2}{d(r,s)^4}$$

- plus alle möglichen Effekte
  - Abschattung, Reflexionen, Diffusion, Antennenform,...





Einschub: Drahtlose Kommunikation

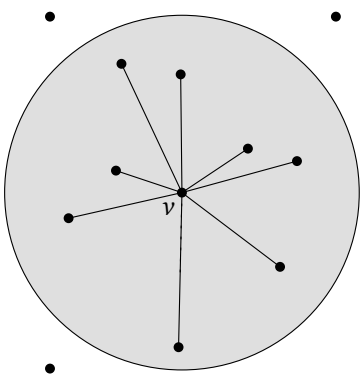
- Signalstärke am Empfänger kann ausreichen
  - um die Nachricht zu entschlüsseln
  - um zu erkennen, dass eine Nachricht gesendet wurde
  - um andere Nachrichten zu stören
- Verbindungen („Links“)
  - Knoten können prinzipiell Nachrichten austauschen
  - (oft nur: geringe Fehlerwahrscheinlichkeit)
  - heute: einfache Verbindungsmodelle (UDG, QUDG)
- Interferenz
  - Kommunikation kann nicht beliebig parallel stattfinden
  - einfache Interferenzmodelle ab übernächster Woche
  - komplexere Modelle in der zweiten Hälfte

Unit-Disk-Graph

Definition *Unit-Disk-Graph*

Ein Graph heißt *Unit-Disk-Graph (UDG)*, wenn es eine Einbettung $\mathbf{p}$ in der Ebene gibt, so dass jeder Knoten genau mit allen Knoten im Abstand ≤ 1 verbunden ist, d. h.

$$\{u, v\} \in E \Leftrightarrow |\mathbf{p}(u) - \mathbf{p}(v)| \leq 1$$



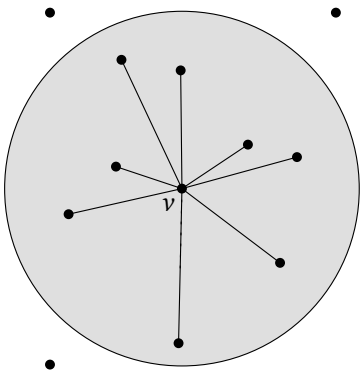
- Achtung:
  - UDG zu sein ist eine kombinatorische Eigenschaft
  - Ein Unit-Disk-Graph bleibt ein Unit-Disk-Graph, auch wenn man ihn anders einbettet!
  - Zu wissen, dass ein Graph ein UDG ist, heißt nicht, eine entsprechende Einbettung zu kennen
- Wir gehen von entsprechender Einbettung aus!

Unit-Disk-Graph

Definition *Unit-Disk-Graph*

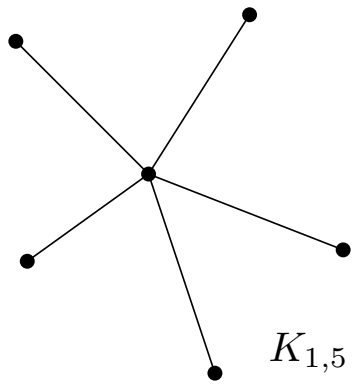
Ein Graph heißt *Unit-Disk-Graph (UDG)*, wenn es eine Einbettung $\mathbf{p}$ in der Ebene gibt, so dass jeder Knoten genau mit allen Knoten im Abstand ≤ 1 verbunden ist, d. h.

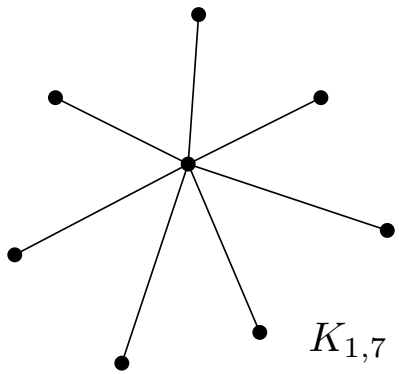
$$\{u, v\} \in E \Leftrightarrow |\mathbf{p}(u) - \mathbf{p}(v)| \leq 1$$



Knobelaufgabe

Ist $K_{1,5}$ bzw. $K_{1,7}$ ein UDG?





Unit-Disk-Graph-Modell

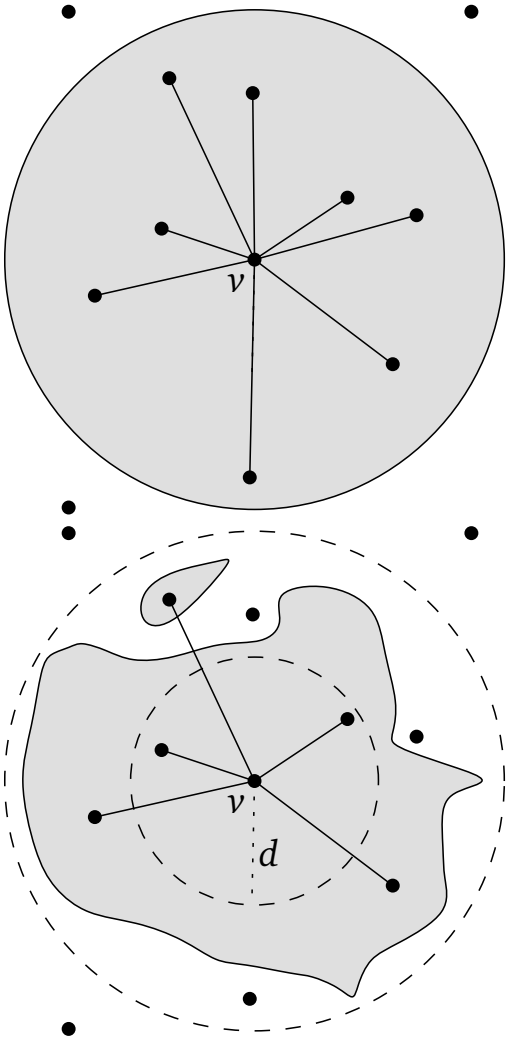
Definition *Unit-Disk-Graph-Modell*

Im *Unit-Disk-Graph-Modell* belegen die Knotenpositionen $\mathbf{p}$ die Eigenschaft des Kommunikationsgraphen, ein Unit-Disk-Graph zu sein, d. h. es gibt einen Kommunikationsradius R, so dass

$$\{u, v\} \in E \Leftrightarrow |\mathbf{p}(u) - \mathbf{p}(v)| \leq R$$

- beschreibt zu gegebenen Positionen und Sendereichweite R genau, wie der Kommunikationsgraph aussieht
- Vorstellung: alle Sender haben eine gemeinsame maximale Sendereichweite R
- unrealistisch, aber gut zu analysieren

d-Quasi-Unit-Disk-Graph



Definition *d-Quasi-Unit-Disk-Graph*

Ein Graph heißt *d-Unit-Disk-Graph (d-QUDG)* für $d \leq 1$, wenn es eine Einbettung $\mathbf{p}$ in die Ebene gibt, so dass jeder Knoten nur mit Knoten in Abstand ≤ 1 verbunden ist, darunter mit allen in Abstand $\leq d$, d. h.

$$\{u, v\} \in E \Rightarrow |\mathbf{p}(u) - \mathbf{p}(v)| \leq 1$$

und

$$\{u, v\} \notin E \Rightarrow |\mathbf{p}(u) - \mathbf{p}(v)| > d$$

Geographisches Routing: Problem

Gegeben: Eingebetteter Unit-Disk-Graph $G = (V, E)$, zwei Knoten $s, t \in V$.

Gesucht: Positionsbewusste Routingstrategie, um ein Paket, das die Position $\mathbf{p}(t)$ enthält, von s nach t zu bringen.

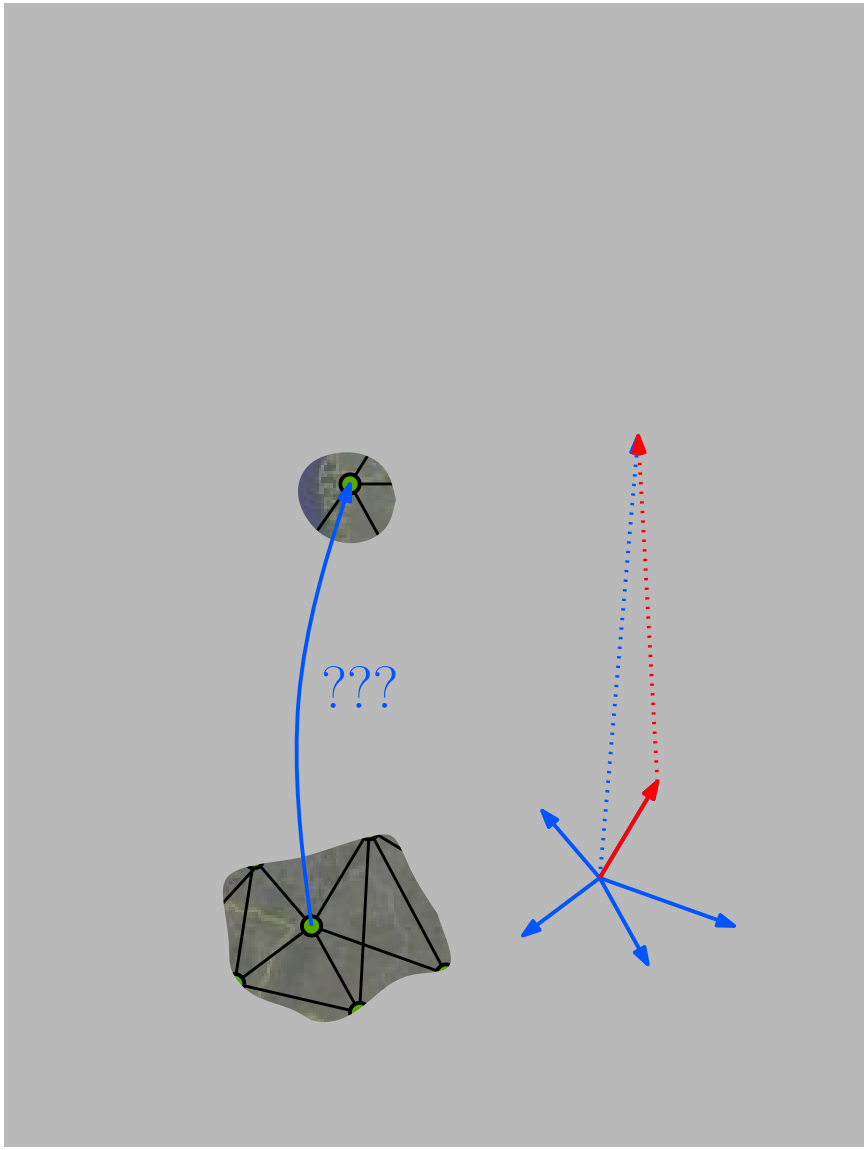
- Es ist erlaubt, „etwas“ Routinginformationen im Paket zu speichern (neben der Zielposition)

Greedy Routing

Greedy Routing

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

Was, wenn es nicht weitergeht?

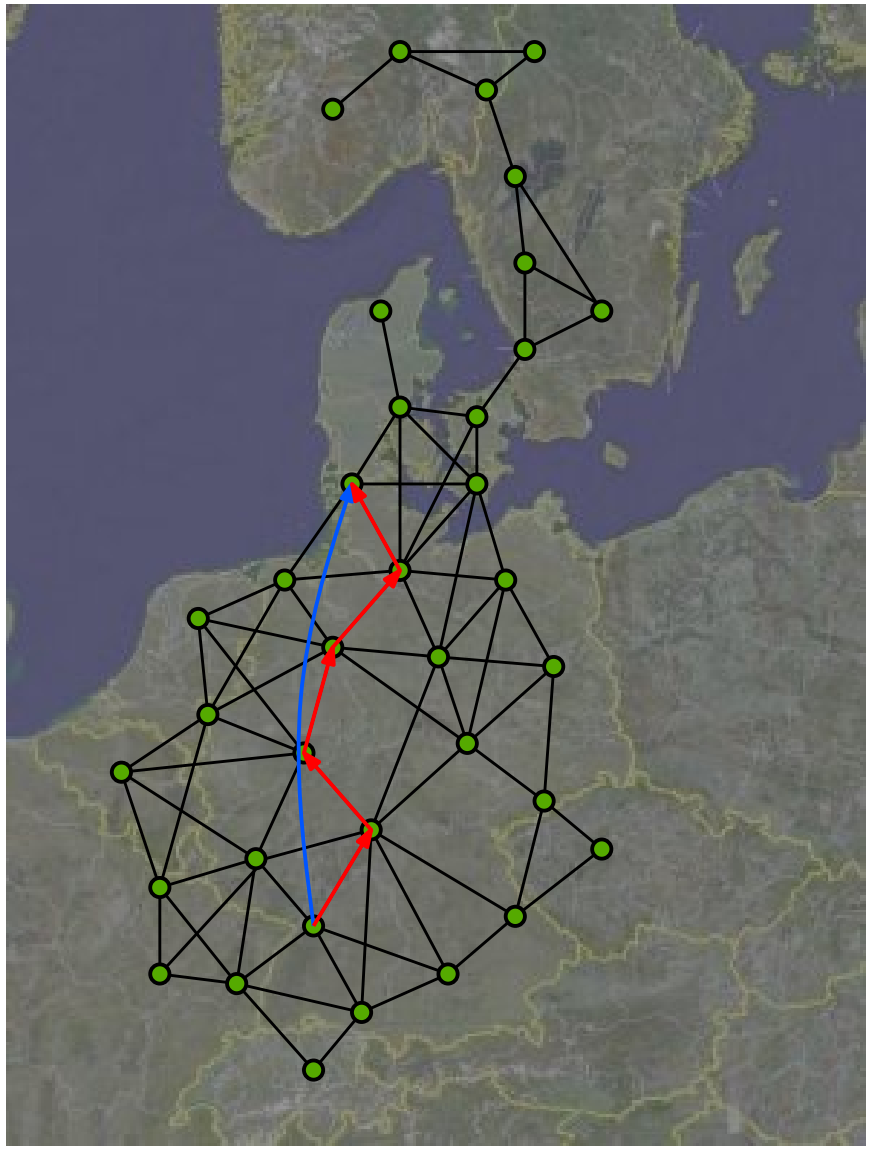


Greedy Routing

Greedy Routing

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

Was, wenn es nicht weitergeht?

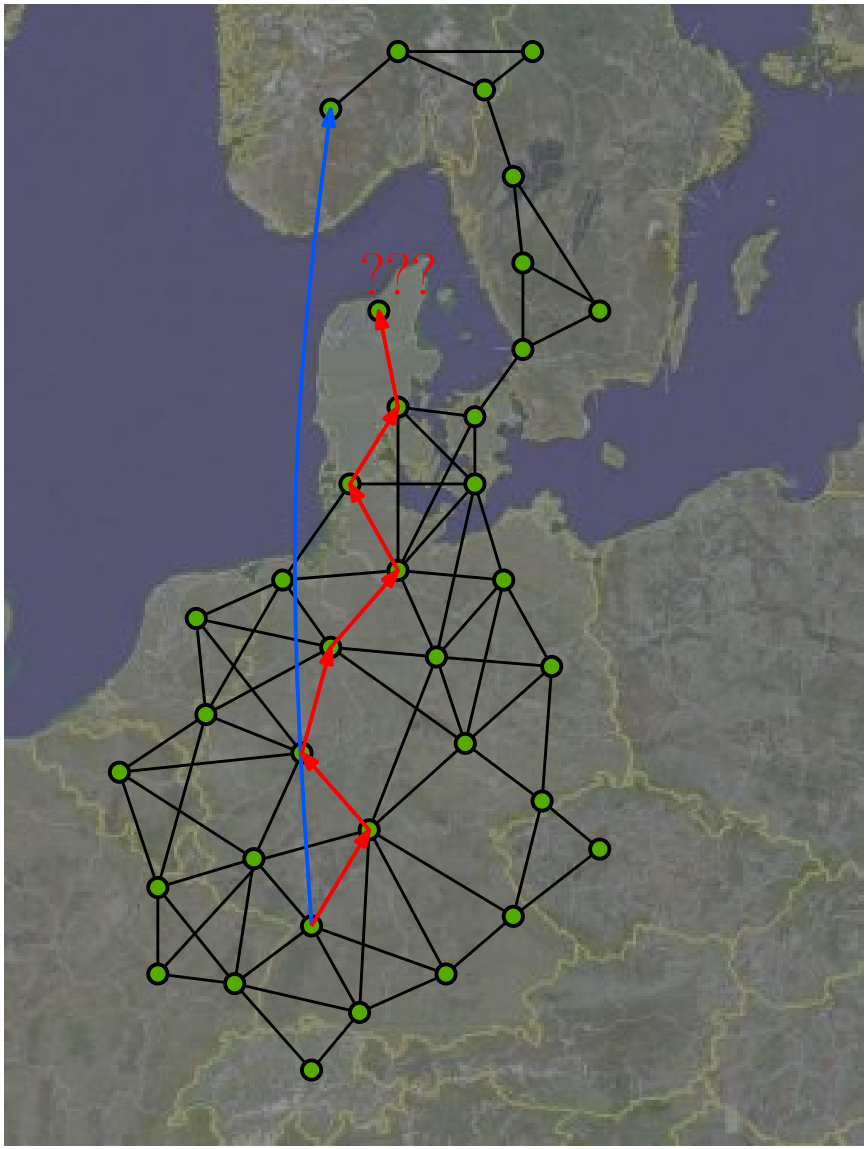


Greedy Routing

Greedy Routing

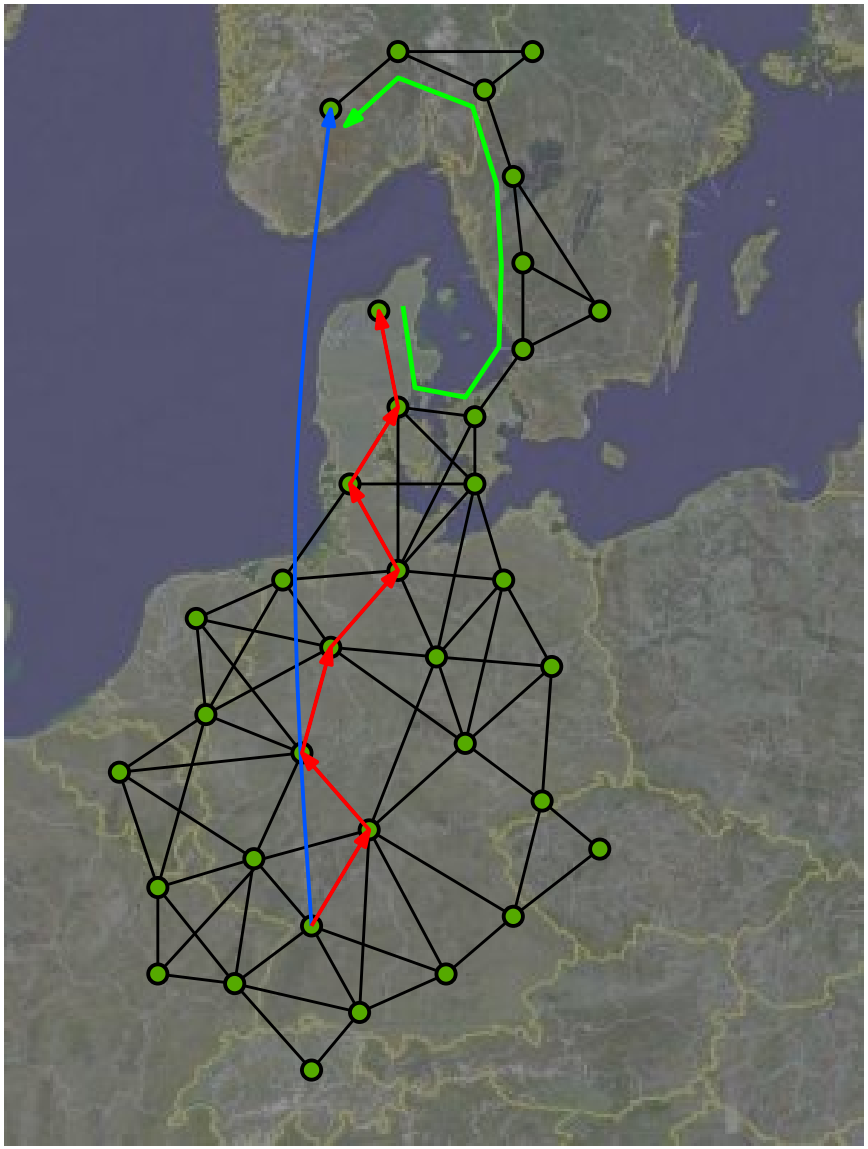
Leite Paket an den Nachbarn, der dem Ziel am nächsten ist.

Was, wenn es nicht weitergeht?



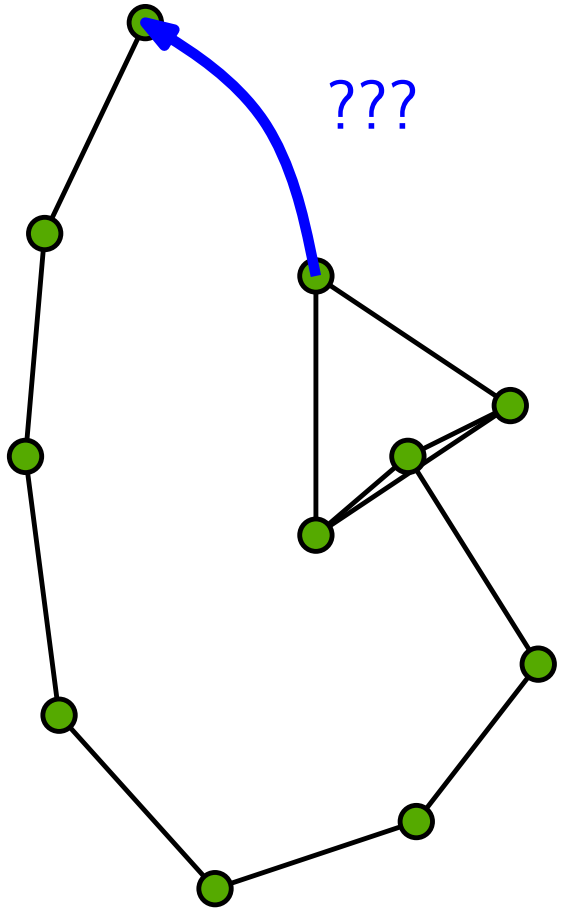
Alte Bekannte

Idee: Irgendwas mit Rechte-Hand-Regel?



Alte Bekannte

Idee: Irgendwas mit Rechte-Hand-Regel?



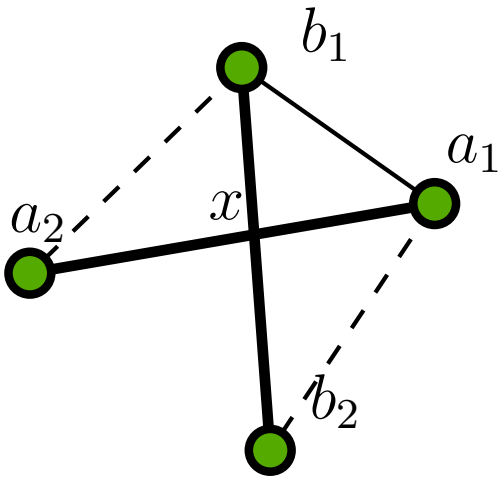
Was haben Irrgärten, was UDG nicht haben?

Planare Subgraphen

Lemma

Jeder eingebettete zusammenhängende Unit-Disk-Graph hat einen zusammenhängenden kreuzungsfreien Teilgraphen[a].

[a] hier immer: geradlinig gezeichnet



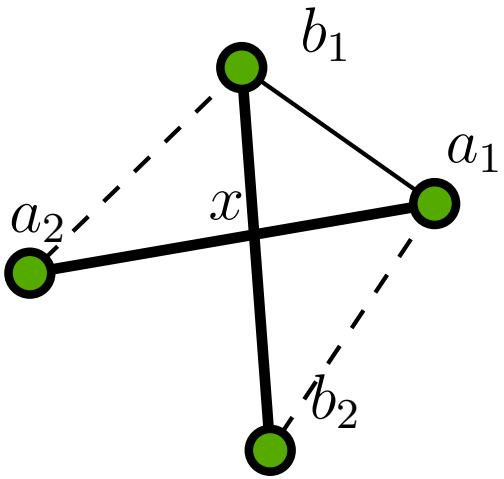
- Betrachte Kreuzung x von $\{a_1, a_2\}$ und $\{b_1, b_2\}$
- o.B.d.A. $|a_1 - x| \leq 0.5$ und $|b_1 - x| \leq 0.5$
  $\Rightarrow |b_1 - a_1| \leq 1$
- $|a_1 - b_2| + |a_2 - b_1| \leq 2$
  - $|a_1 - b_2| \leq 1$ oder $|a_2 - b_1| \leq 1$
- eine der kreuzenden Kanten kann entfernt werden
- Das Lemma gilt für d-QUDGs für alle $d > 1/\sqrt{2}$ (ohne Beweis)

Planare Subgraphen

Lemma

Jeder eingebettete zusammenhängende Unit-Disk-Graph hat einen zusammenhängenden kreuzungsfreien Teilgraphen[a].

[a] hier immer: geradlinig gezeichnet

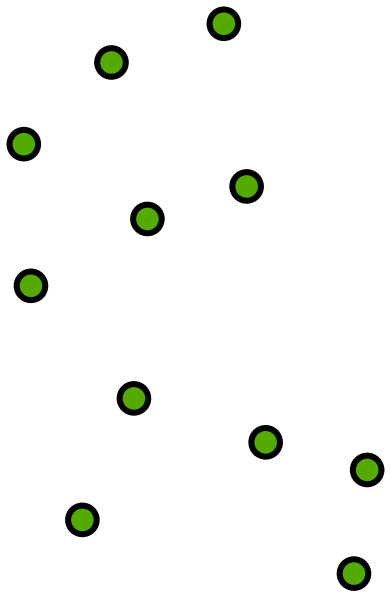


- Betrachte Kreuzung x von $\{a_1, a_2\}$ und $\{b_1, b_2\}$
- o.B.d.A. $|a_1 - x| \leq 0.5$ und $|b_1 - x| \leq 0.5$
  $\Rightarrow |b_1 - a_1| \leq 1$
- $|a_1 - b_2| + |a_2 - b_1| \leq 2$
  - $|a_1 - b_2| \leq 1$ oder $|a_2 - b_1| \leq 1$
- eine der kreuzenden Kanten kann entfernt werden
- Das Lemma gilt für d-QUDGs für alle $d > 1/\sqrt{2}$ (ohne Beweis)

Gabriel Graph

Definition

Sei P eine Menge von Punkten in der Ebene. Der *Gabriel Graph* enthält ein Punktepaar genau dann als Kante, wenn der kleinste Kreis, der beide Punkte enthält, keine weiteren Punkte enthält.

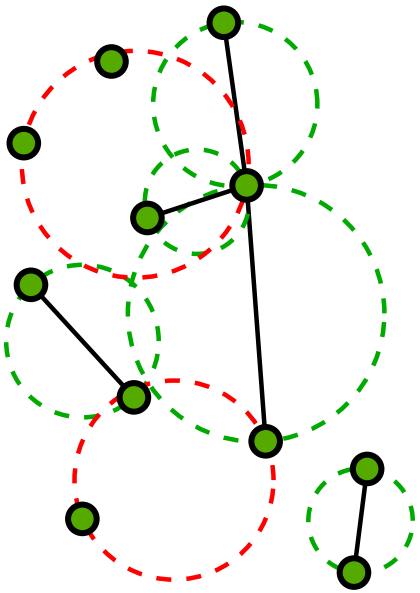


- Gabriel Graph ist planar, aber kann „lange“ Kanten enthalten
  - schränke UDG auf *GG*-Kanten ein!
  - $UDG \cap GG$ ist lokal entscheidbar
- $UDG \cap GG$ ist planar
- $UDG \cap GG$ ist zshg. $\Leftrightarrow$ UDG ist zshg.
- Beweise in VL 04
- QUDG: Cross-Link-Detection-Protokolle!

Gabriel Graph

Definition

Sei P eine Menge von Punkten in der Ebene. Der *Gabriel Graph* enthält ein Punktepaar genau dann als Kante, wenn der kleinste Kreis, der beide Punkte enthält, keine weiteren Punkte enthält.

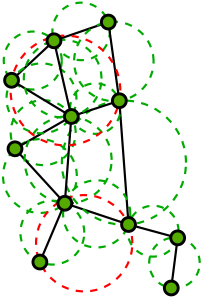


- Gabriel Graph ist planar, aber kann „lange“ Kanten enthalten
  - schränke UDG auf *GG*-Kanten ein!
  - $UDG \cap GG$ ist lokal entscheidbar
  - benutze Cross-Link-Detection-Protokolle
- $UDG \cap GG$ ist planar
- $UDG \cap GG$ ist zshg. $\Leftrightarrow$ UDG ist zshg.
- Beweise in VL 04
- QUDG: Cross-Link-Detection-Protokolle!

Gabriel Graph

Definition

Sei P eine Menge von Punkten in der Ebene. Der *Gabriel Graph* enthält ein Punktepaar genau dann als Kante, wenn der kleinste Kreis, der beide Punkte enthält, keine weiteren Punkte enthält.



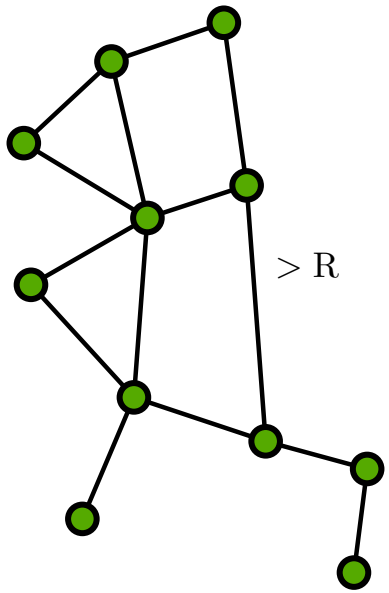
- Gabriel Graph ist planar, aber kann „lange“ Kanten enthalten
  - schränke UDG auf *GG*-Kanten ein!
  - $UDG \cap GG$ ist lokal entscheidbar

    beide Enden "sehen" störende Knoten
- $UDG \cap GG$ ist planar
- $UDG \cap GG$ ist zshg. $\Leftrightarrow$ UDG ist zshg.
- Beweise in VL 04
- QUDG: Cross-Link-Detection-Protokolle!

Gabriel Graph

Definition

Sei P eine Menge von Punkten in der Ebene. Der *Gabriel Graph* enthält ein Punktepaar genau dann als Kante, wenn der kleinste Kreis, der beide Punkte enthält, keine weiteren Punkte enthält.

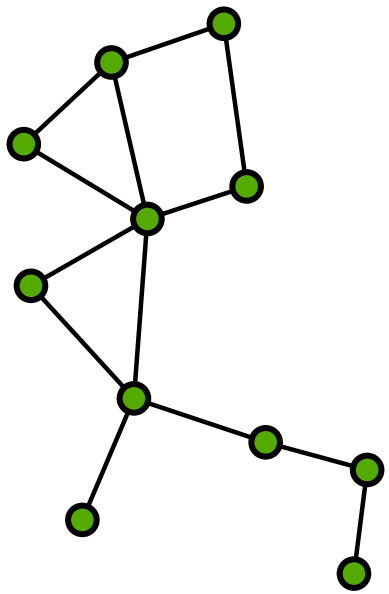


- Gabriel Graph ist planar, aber kann „lange“ Kanten enthalten
  - schränke UDG auf *GG*-Kanten ein!
  - *UDG* ∩ *GG* ist lokal entscheidbar
    - beide Enden “sehen“ störende Knoten
- *UDG* ∩ *GG* ist planar
- *UDG* ∩ *GG* ist zshg. ⇔ UDG ist zshg.
- Beweise in VL 04
- QUDG: Cross-Link-Detection-Protokolle!

Gabriel Graph

Definition

Sei *P* eine Menge von Punkten in der Ebene. Der *Gabriel Graph* enthält ein Punktepaar genau dann als Kante, wenn der kleinste Kreis, der beide Punkte enthält, keine weiteren Punkte enthält.

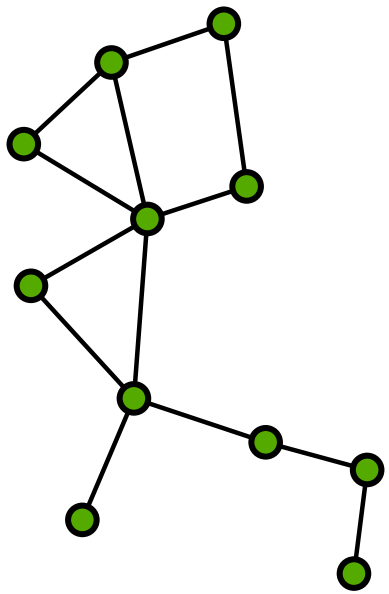


- Gabriel Graph ist planar, aber kann „lange“ Kanten enthalten
  - schränke UDG auf *GG*-Kanten ein!
  - *UDG* ∩ *GG* ist lokal entscheidbar
    - beide Enden “sehen“ störende Knoten
- *UDG* ∩ *GG* ist planar
- *UDG* ∩ *GG* ist zshg. ⇔ UDG ist zshg.
- Beweise in VL 04
- QUDG: Cross-Link-Detection-Protokolle!

Gabriel Graph

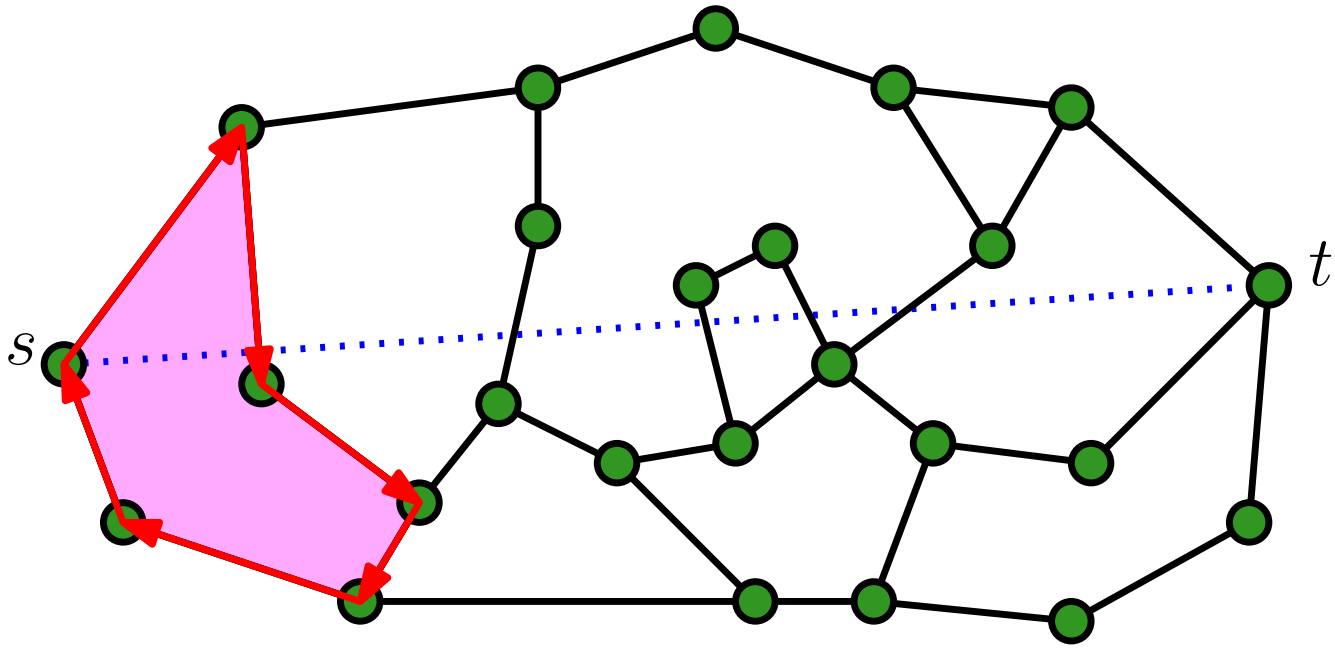
Definition

Sei P eine Menge von Punkten in der Ebene. Der *Gabriel Graph* enthält ein Punktepaar genau dann als Kante, wenn der kleinste Kreis, der beide Punkte enthält, keine weiteren Punkte enthält.



- Gabriel Graph ist planar, aber kann „lange“ Kanten enthalten
  - schränke UDG auf *GG*-Kanten ein!
  - $UDG \cap GG$ ist lokal entscheidbar
    - beide Enden “sehen“ störende Knoten
- $UDG \cap GG$ ist planar
- $UDG \cap GG$ ist zshg. $\Leftrightarrow$ UDG ist zshg.
- Beweise in VL 04
- QUDG: Cross-Link-Detection-Protokolle!

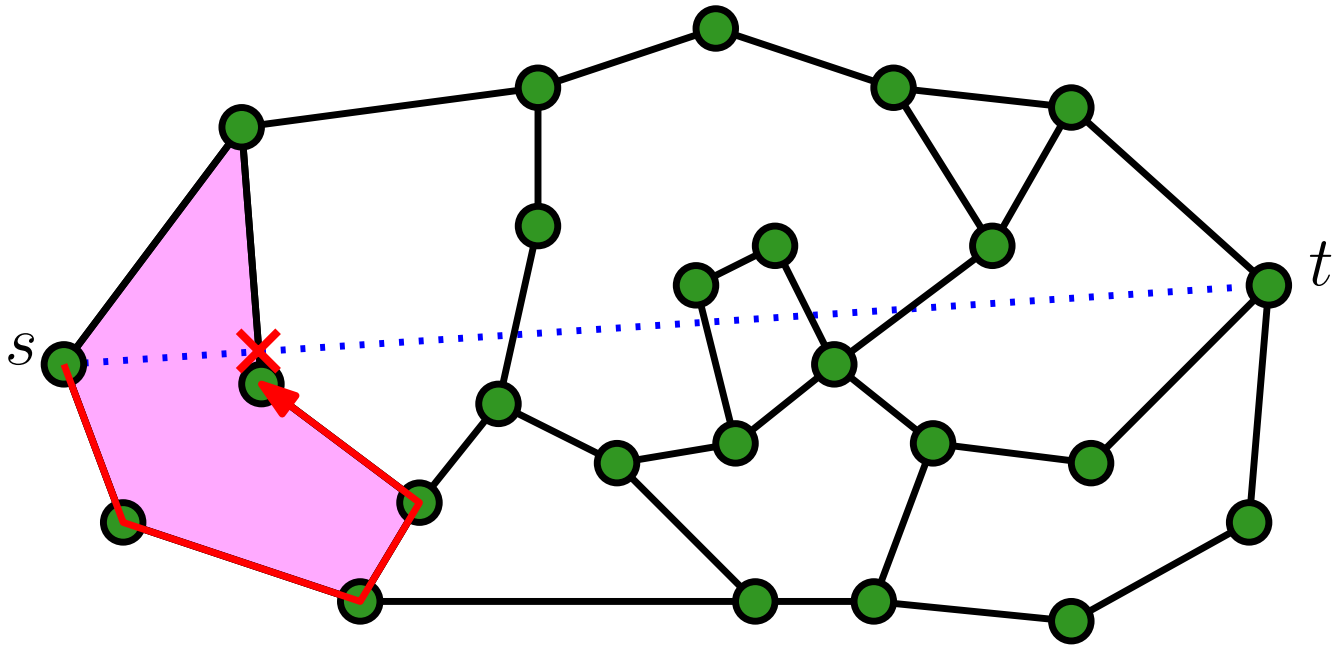
Facettenrouting (FR)



Facettenrouting [Kranakis et. al. ’99]

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

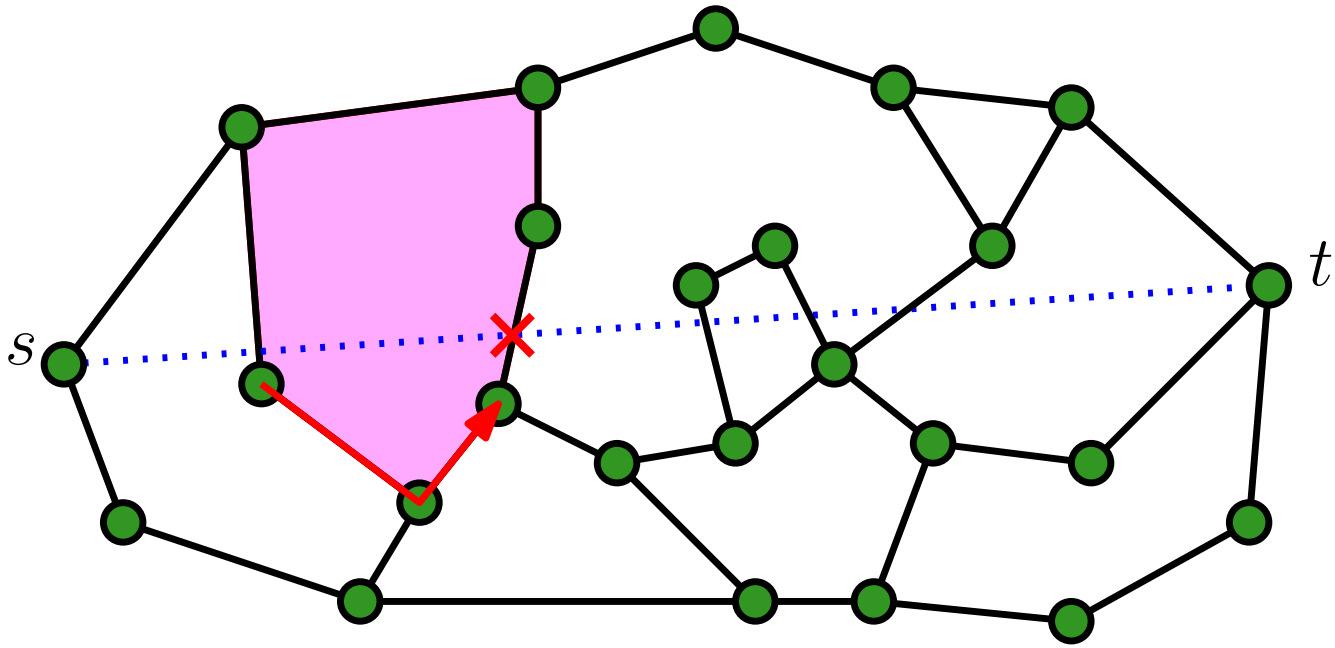
Facettenrouting (FR)



Facettenrouting [Kranakis et. al. ’99]

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

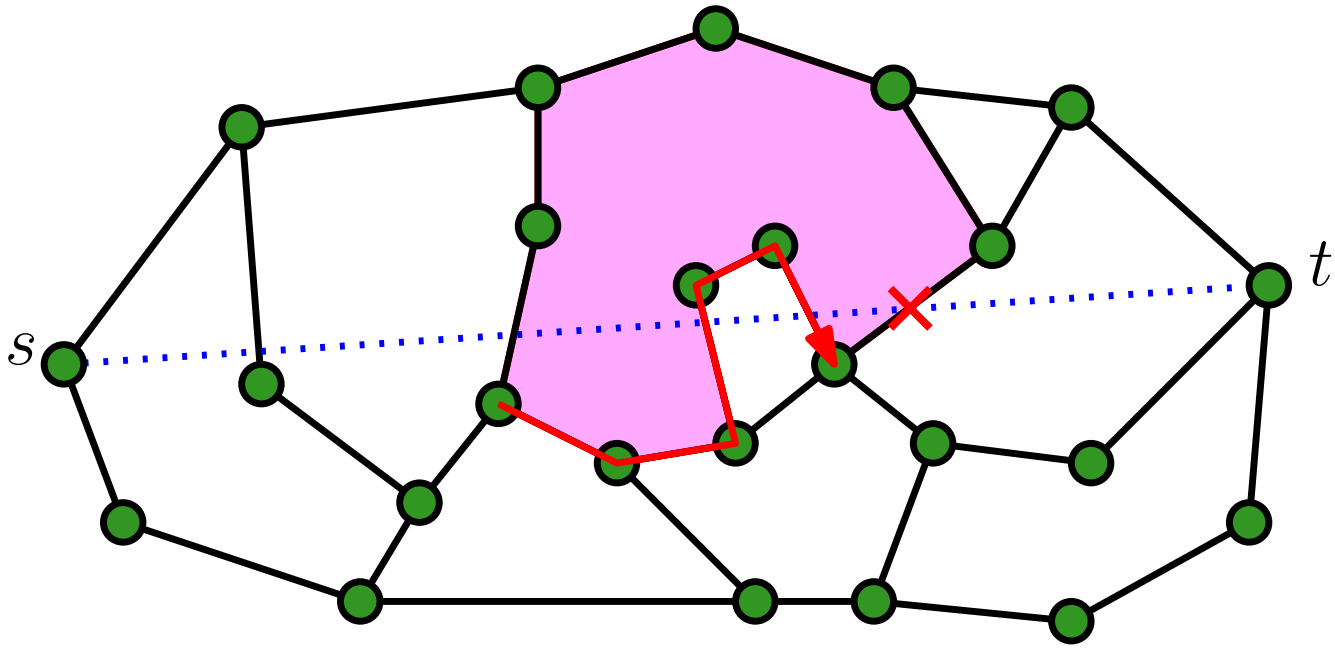
Facettenrouting (FR)



Facettenrouting [Kranakis et. al. ’99]

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

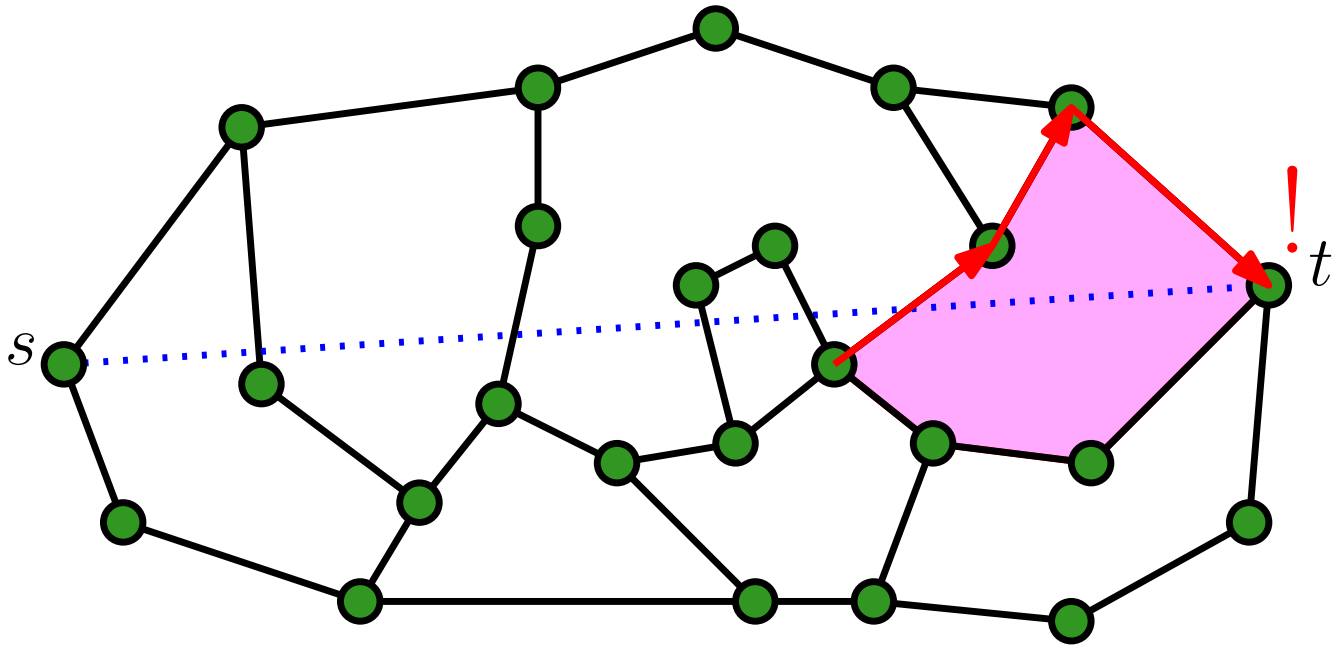
Facettenrouting (FR)



Facettenrouting [Kranakis et. al. ’99]

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

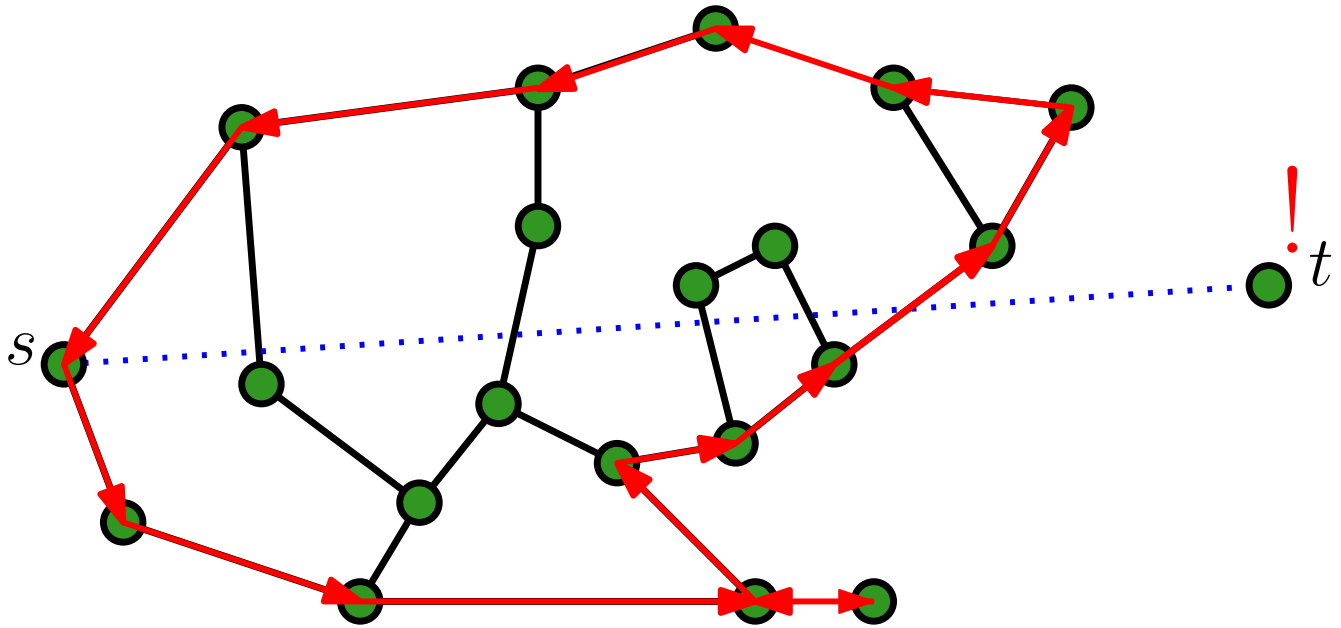
Facettenrouting (FR)



Facettenrouting [Kranakis et. al. ’99]

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

Facettenrouting (FR)



Fehlerbehandlung

- Was, wenn kein besserer Schnitt gefunden wird?
- Schicke Paket mit Fehlermeldung an *s* zurück
- einfach wieder per Facettenrouting

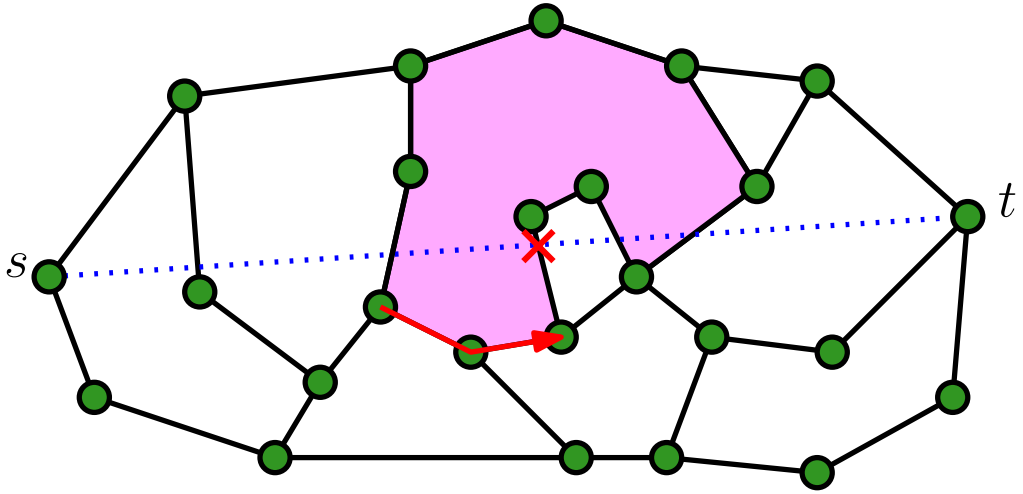
Korrektheitsbeweis und Laufzeit

Satz

Facettenrouting findet in $O(n)$ Schritten zum Ziel oder erkennt fehlenden Zusammenhang.

- Beweis:
  - Jede Facette wird maximal einmal behandelt
    - Facetten ohne Schnitt mit $\overrightarrow{st}$ fallen weg
    - sortiere andere Facetten nach letztem Schnitt mit $\overrightarrow{st}$
    - Facetten werden nur aufsteigend besucht!
  - jede Kante wird maximal viermal genutzt
    - zweimal in jeder anliegenden Facette (Erkundung, Rückkehr zum besten Schnitt)
  - Euler: Planare Graphen haben maximal $3n - 6 \in O(n)$ Kanten
  - Fehlermeldung kann Aufwand verdoppeln

Vereinfachung Facettenrouting

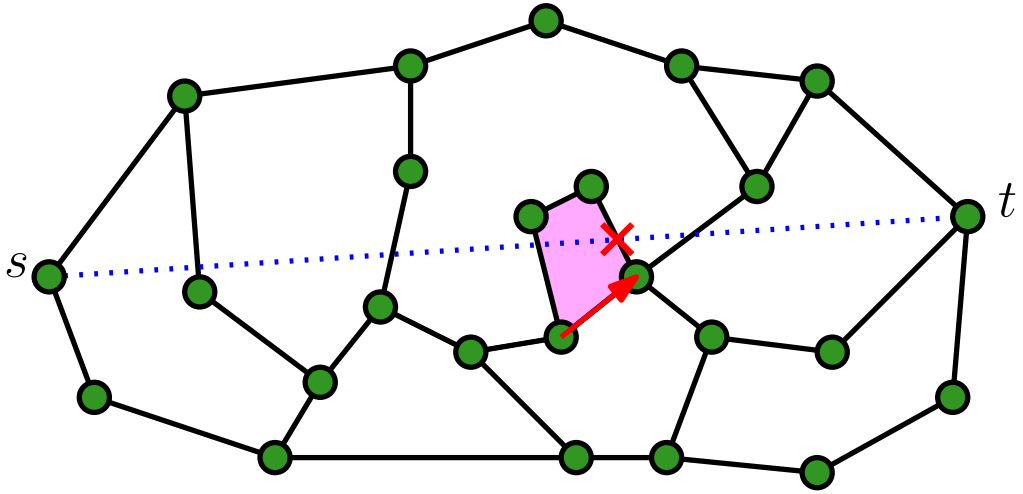


Vereinfachtes Facettenrouting

- Erkunde Facette, in die $\overrightarrow{st}$ zeigt
- Sobald besserer Schnitt mit $\overrightarrow{st}$ gefunden ist
  - Wechsle zur angrenzenden Facette
- beende, wenn Facette keinen besseren Schnitt enthält

- Korrekt, aber asymptotisch nicht besser! (ohne Beweis)

Vereinfachung Facettenrouting



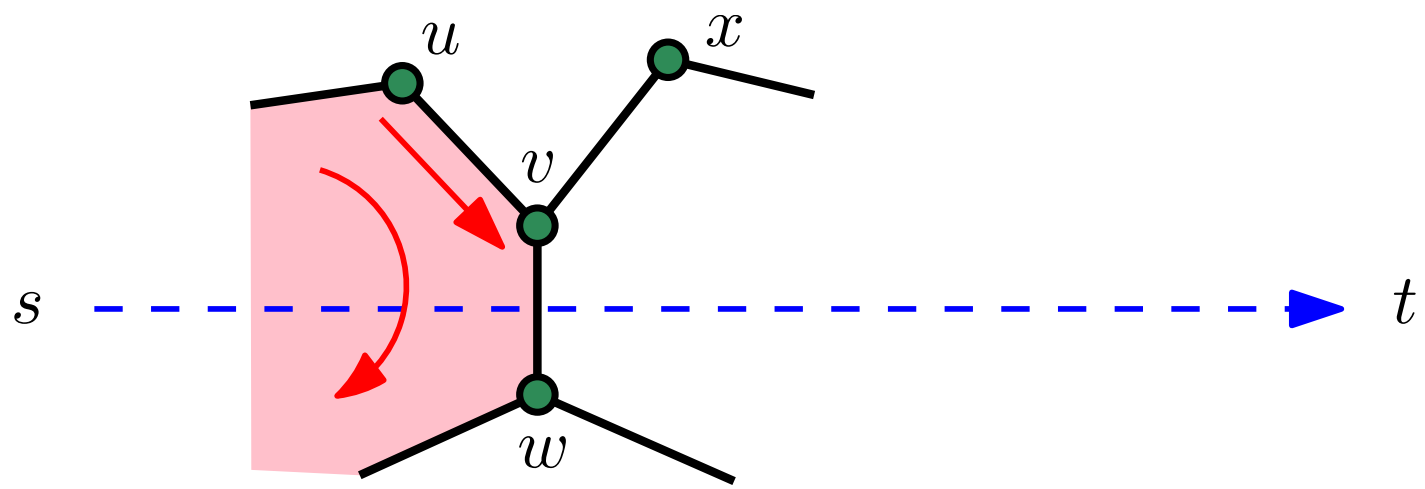
Vereinfachtes Facettenrouting

- Erkunde Facette, in die $\overrightarrow{st}$ zeigt
- Sobald besserer Schnitt mit $\overrightarrow{st}$ gefunden ist
  - Wechsle zur angrenzenden Facette
- beende, wenn Facette keinen besseren Schnitt enthält

- Korrekt, aber asymptotisch nicht besser! (ohne Beweis)

Regeln des Facettenroutings

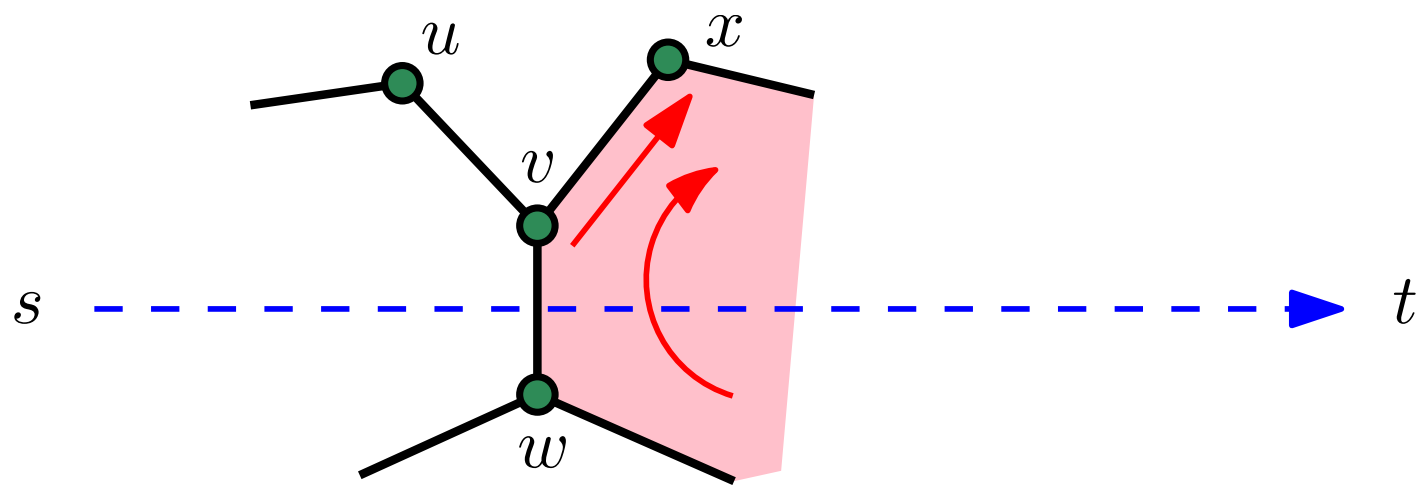
- Routinginformation komplett in Nachricht enthalten
  - Start- und Zielkoordinaten
  - bester Übergangspunkt zur nächsten Facette
- Vollständig lokal
  - Positionen der Nachbarknoten reichen aus
  - Facettenerkundung nur implizit durch Linke-Hand-Regel



u sendet an *v* eine Nachricht $\textbf{TRAVERSE}^{u}_{\text{CW}}(s, t, \text{bestKnownIntersection})$
v empfängt und sieht einen besseren Schnitt bei dem Nachfolger *w*

Regeln des Facettenroutings

- Routinginformation komplett in Nachricht enthalten
  - Start- und Zielkoordinaten
  - bester Übergangspunkt zur nächsten Facette
- Vollständig lokal
  - Positionen der Nachbarknoten reichen aus
  - Facettenerkundung nur implizit durch Linke-Hand-Regel



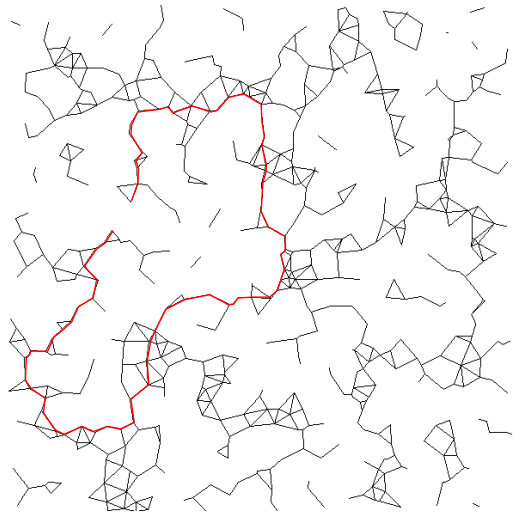
v bestimmt den Nachfolger *x* von *w* im Gegenuhrzeigersinn und sendet eine Nachricht $\textbf{TRAVERSE}^{v}_{\text{CW}}(s, t, \text{newBestKnownIntersection})$ an *x*

Bessere Analyse

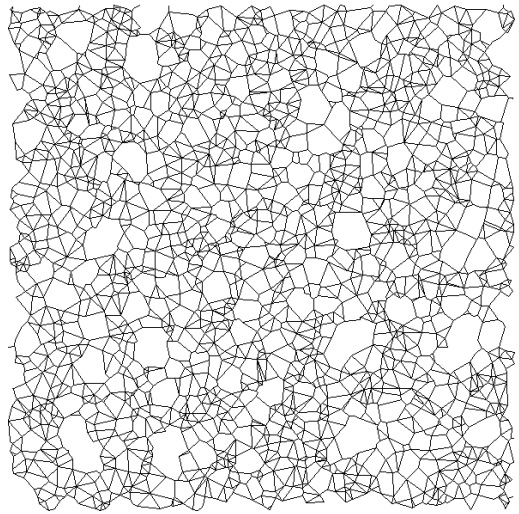
Facettenrouting findet in $O(n)$ Schritten zum Ziel oder erkennt fehlenden Zusammenhang. Was, wenn Start und Ziel dicht beieinander liegen?

- Nicht immer gibt es einen kurzen Weg!

kein Zusammenhang

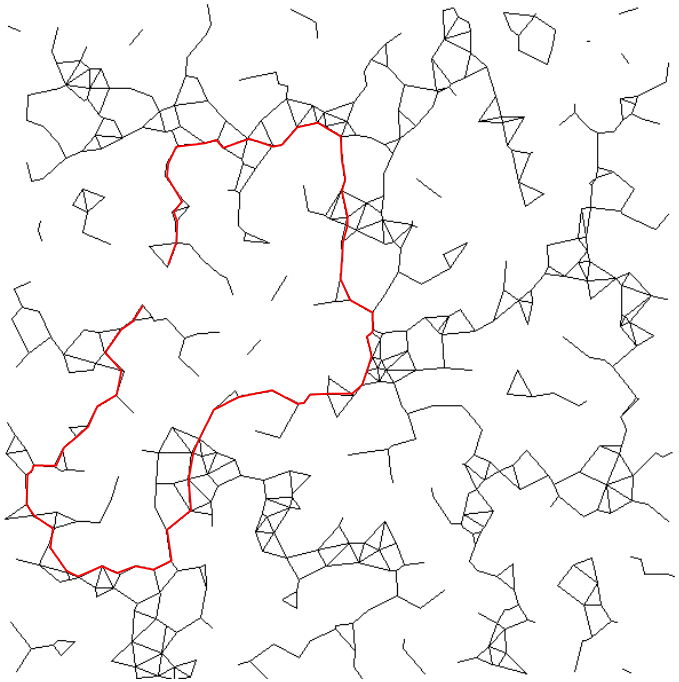


interessant



greedy ausreichend

Bessere Analyse



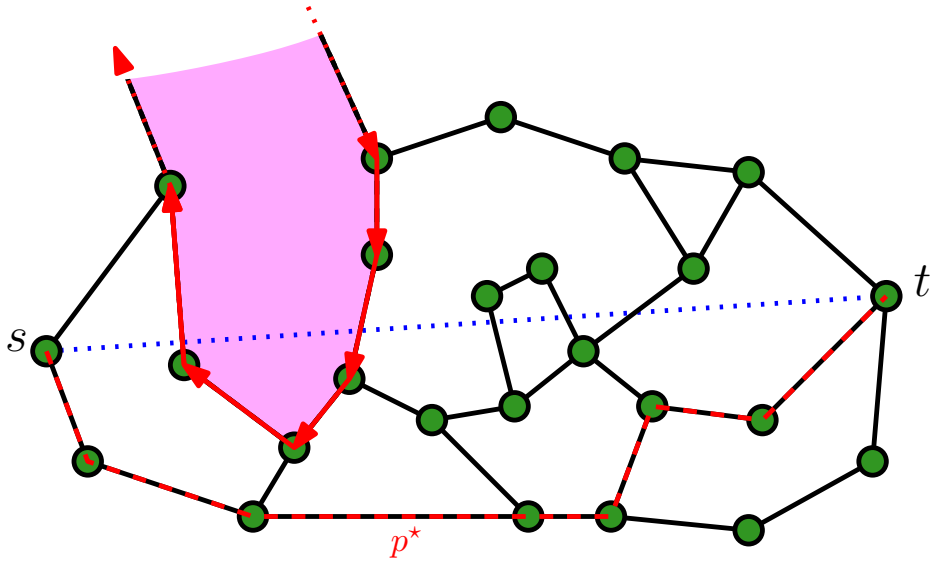
- Kosten des kürzesten Weges $c(p^\star)$ (Summe der Kantenlängen)
- Euklidischer Abstand $|\mathbf{p}(s) - \mathbf{p}(t)|$
- $c(p^\star) \gg |\mathbf{p}(s) - \mathbf{p}(t)|$

Kann man wenigstens so routen, dass man nicht beliebig schlechter als der kürzeste Weg ist?

Bessere Analyse? Unmöglich!

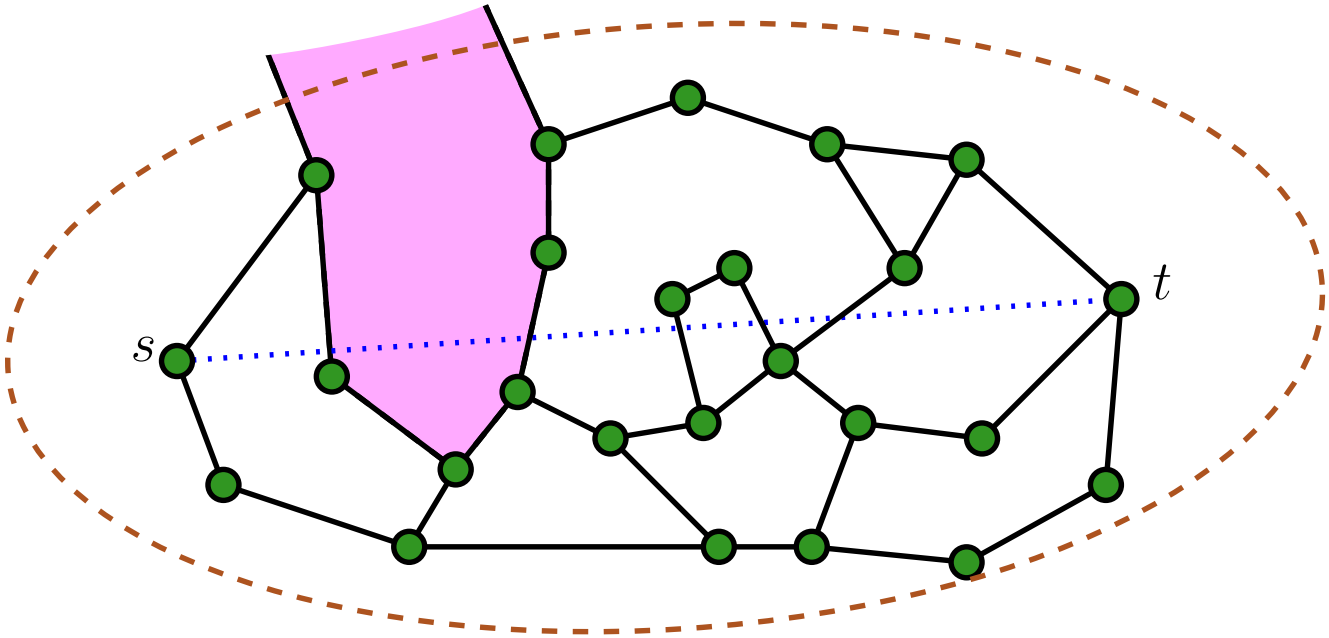
Problem

Facettenrouting lässt keine Abschätzung der Weglänge in Abhängigkeit vom kürzesten Weg zu!



Gibt es einen besseren Routingalgorithmus?

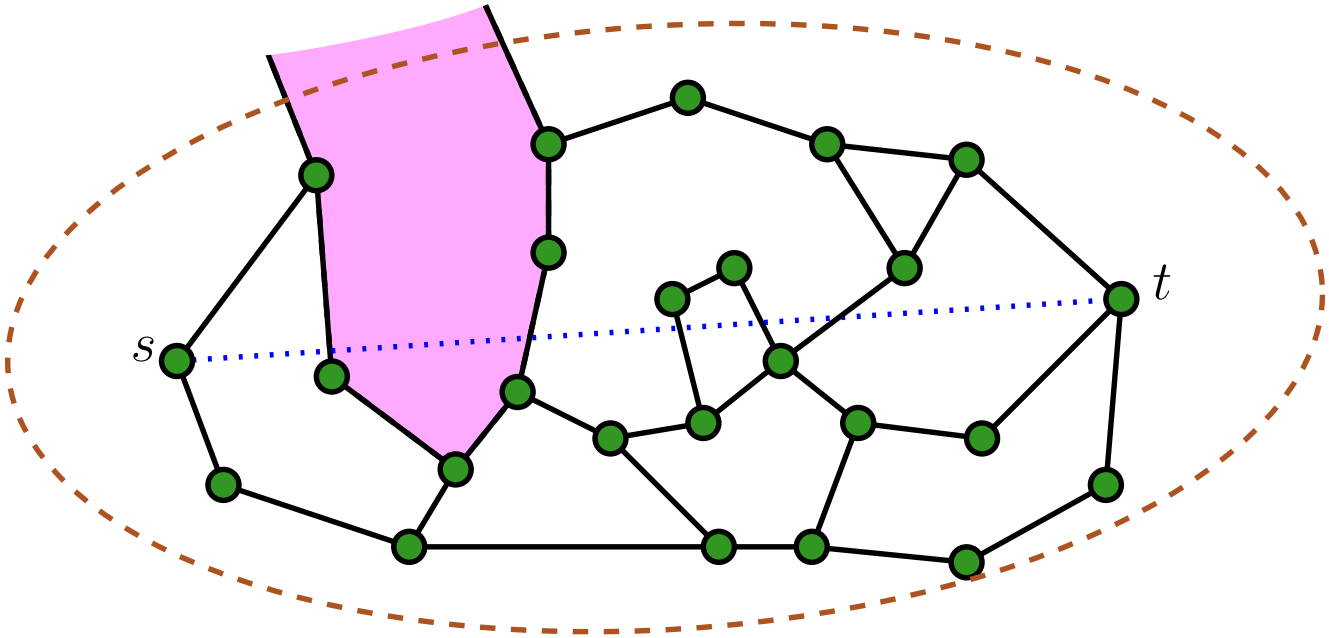
Beschränktes Facettenrouting (BFR)



Lemma

Ein Weg p zwischen s und t mit Länge $c(p)$ liegt vollständig innerhalb der Ellipse mit Foci s, t und Hauptachse $c(p)$. Diese Ellipse ist Menge aller Punkte x mit $|\mathbf{p}(x) - \mathbf{p}(s)| + |\mathbf{p}(x) - \mathbf{p}(t)| \leq c(p)$.

Beschränktes Facettenrouting (BFR)

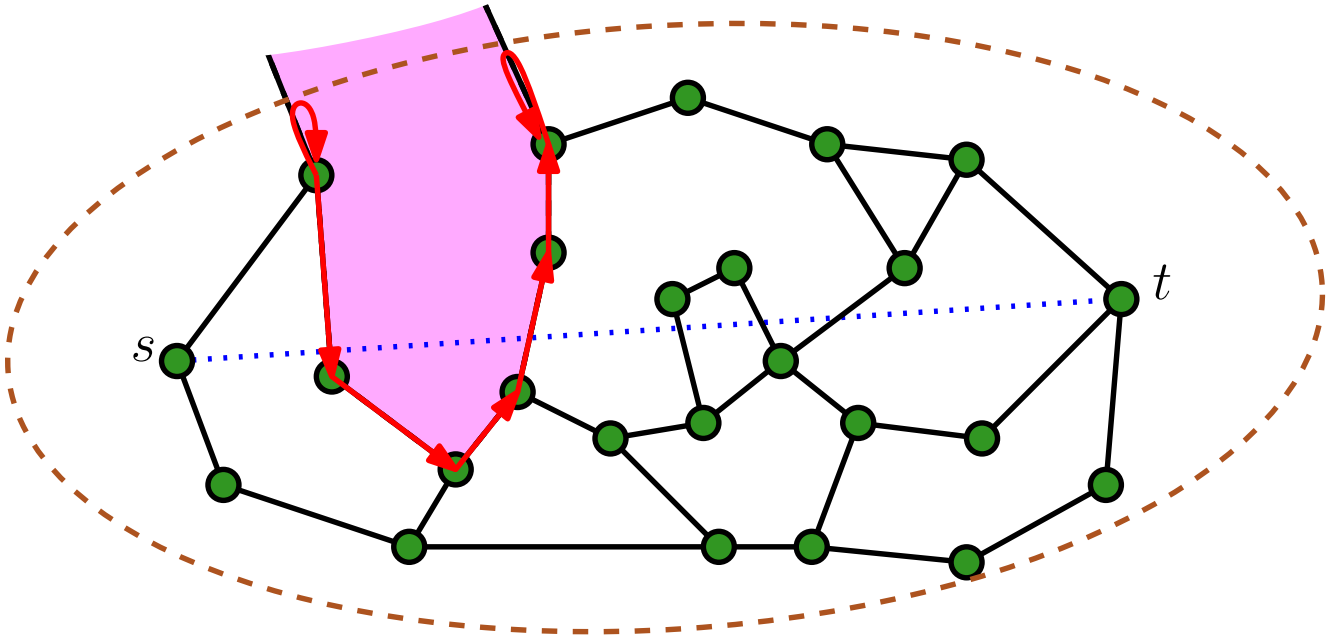


Lemma

Ein Weg p zwischen s und t mit Länge $c(p)$ liegt vollständig innerhalb der Ellipse mit Foci s, t und Hauptachse $c(p)$. Diese Ellipse ist Menge aller Punkte x mit $|\mathbf{p}(x) - \mathbf{p}(s)| + |\mathbf{p}(x) - \mathbf{p}(t)| \leq c(p)$.

Beweis?

Beschränktes Facettenrouting (BFR)



Beschränktes Facettenrouting (BFR)

Ist die Länge $c(p^\star)$ des kürzesten Weges bekannt, beschränke Facettenrouting durch Ellipse mit Foci s und t und Länge der Hauptachse $c(p^\star)$.

The secret ingredient: $\Omega(1)$-Modell

Definition: $\Omega(1)$-Modell

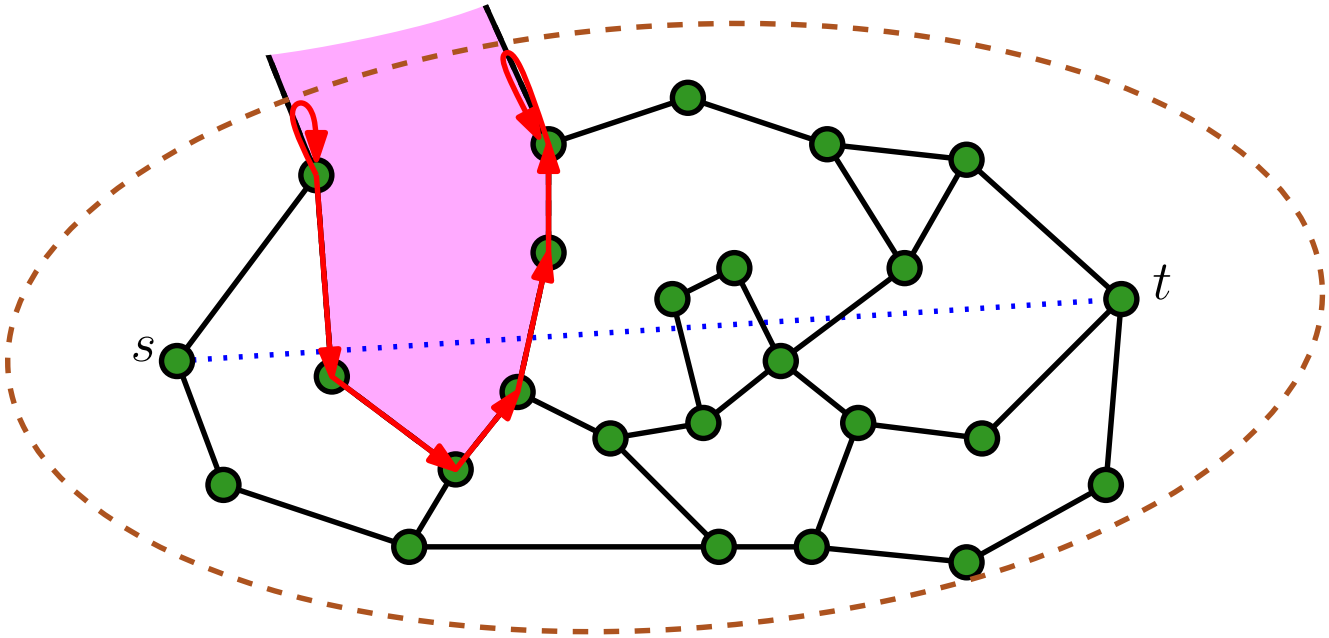
Im $\Omega(1)$-Modell ist der minimale Abstand zwischen zwei Knoten durch eine Konstante d_0 nach unten beschränkt.

Lemma

Im $\Omega(1)$-Modell besucht das beschränkte Facettenrouting nur $O(c(p^\star)^2)$ Knoten.

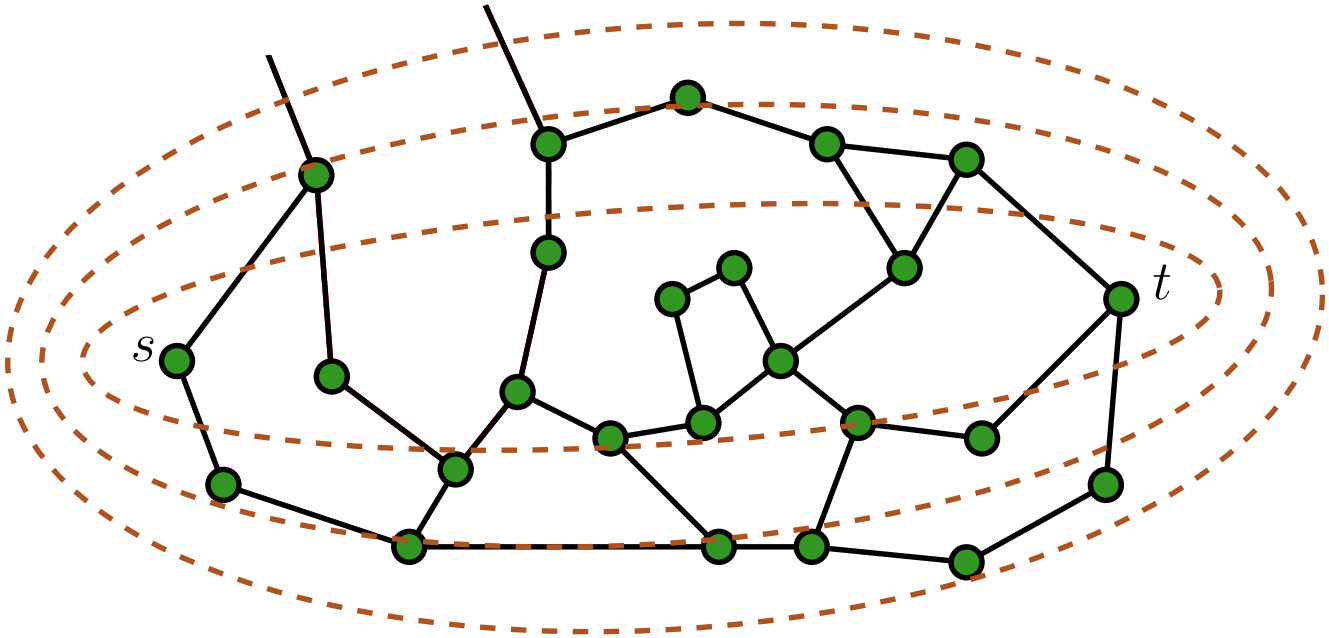
- Beweis:
  - Ellipse ist vollständig in Kreis mit Durchmesser $c(p^\star)$ um Mitte von s und t enthalten.
  - In einen Kreis mit Durchmesser d passen nur $O((d/d_0)^2)$ Punkte mit paarweisem Abstand d_0
  - d_0 konstant $\Rightarrow$ $O(c(p^\star)^2)$ Knoten im Kreis!

Beschränktes Facettenrouting (BFR)



Woher sollte man wissen, wie lang der kürzeste Weg zwischen Start- und Zielknoten ist? Was macht man, wenn man es nicht weiß?

Adaptives Facettenrouting (AFR)



Adaptives Facettenrouting (AFR)

Ist die Weglänge $c(p^\star)$ zum Ziel nicht bekannt, führe BFR mit $c_0 = 2 \cdot |st|$ durch. Schlägt das Routing fehl, starte BFR mit $c_1 = 2 \cdot c_0$, $c_2 = 2 \cdot c_1 \ldots$ $(c_i = 2^i \cdot c_0)$

Adaptives Facettenrouting – Analyse

Satz

Im $\Omega(1)$-Modell erreicht das Adaptive Facettenrouting das Ziel mit Kosten in $O\left(c^2(p^\star)\right)$, wenn die optimale Route Kosten $c(p^\star)$ hat.

- Beweis
  - spätestens in Runde k erfolgreich mit $c_{k-1} \leq c(p^\star) \leq c_k$
  - $\text{cost}_{\text{AFR}} = \sum_{i=0}^{k} c_i^2 = \sum_{i=0}^{k} (2^i c_0)^2 = \sum_{i=0}^{k} 4^i c_0^2$
  - $= \frac{4^{k+1}-1}{3} c_0^2 < \frac{16}{3}(2^{k-1} c_0)^2 = \frac{16}{3} c_{k-1}^2 \leq \frac{16}{3} c^2(p^\star)$

Adaptives Facettenrouting – Analyse

Satz

Im $\Omega(1)$-Modell erreicht das Adaptive Facettenrouting das Ziel mit Kosten in $O\left(c^2(p^\star)\right)$, wenn die optimale Route Kosten $c(p^\star)$ hat.

- Beweis
  - spätestens in Runde k erfolgreich mit $c_{k-1} \leq c(p^\star) \leq c_k$
  - $\text{cost}_{\text{AFR}} = \sum_{i=0}^{k} c_i^2 = \sum_{i=0}^{k} (2^i c_0)^2 = \sum_{i=0}^{k} 4^i c_0^2$
  - $= \frac{4^{k+1}-1}{3} c_0^2 < \frac{16}{3}(2^{k-1} c_0)^2 = \frac{16}{3} c_{k-1}^2 \leq \frac{16}{3} c^2(p^\star)$

Adaptives Facettenrouting – Analyse

Satz

Im $\Omega(1)$-Modell erreicht das Adaptive Facettenrouting das Ziel mit Kosten in $O\left(c^2(p^\star)\right)$, wenn die optimale Route Kosten $c(p^\star)$ hat.

- Beweis
  - spätestens in Runde k erfolgreich mit $c_{k-1} \leq c(p^\star) \leq c_k$
  - $\text{cost}_{\text{AFR}} = \sum_{i=0}^{k} c_i^2 = \sum_{i=0}^{k} (2^i c_0)^2 = \sum_{i=0}^{k} 4^i c_0^2$
  - $= \frac{4^{k+1}-1}{3} c_0^2 < \frac{16}{3}(2^{k-1} c_0)^2 = \frac{16}{3} c_{k-1}^2 \leq \frac{16}{3} c^2(p^\star)$

Adaptives Facettenrouting – Analyse

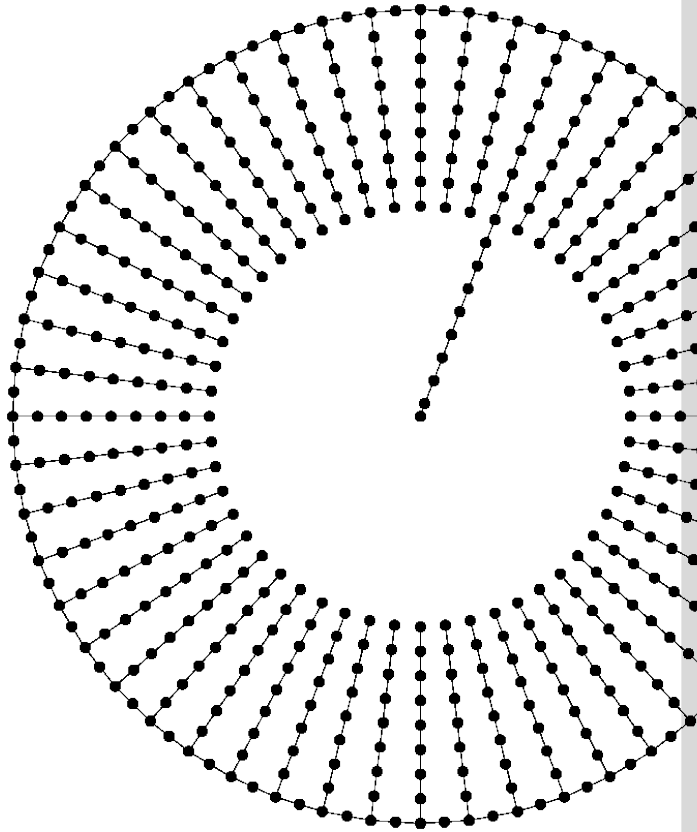
Bemerkung

Fehlender Zusammenhang wird in $O(n' \log n')$ Schritten erkannt, wenn n' die Anzahl der Knoten in der Zhgs.-Komponente von s ist.

- Entferntester Knoten ist maximal n' Funkradien von s entfernt
- Nach $O(\log n')$ Ellipsenverdoppelungen alle Knoten enthalten
  - Erkennung: Paket „stößt“ nicht mehr gegen die Ellipse
- $\Rightarrow O(\log n')$ BFR-Phasen mit je maximal $O(n')$ Schritten

$O(c(p^\star)^2)$ ist worst-case-optimal

- c Pfade vom Rand Richtung Mitte
  - jeder hat Länge $\Theta(c)$
- Ziel: Knoten in der Mitte
- Start: Beliebiger Knoten auf dem Ring
- Ohne Routinginformationen könnte jeder Pfad zur Mitte führen
- Verfahren muss $\Omega(c)$ Pfade testen.
- Bester Weg hat Länge $O(c)$
- Kosten: $\Omega(c^2)$ statt $O(c)$



Satz

AFR ist (asymptotisch) worst-case-optimal.

Überblick der bish. Algorithmen

Schön, aber
etwas langsam

FR

Ellipt.
Suchraum

Schnell, aber
sehr neugierig

BFR

Adaptiver
Suchraum

Praktischer
Jüngling

Greedy

AFR

Theoretische
Schönheit

Greedy AFR (GAFR)

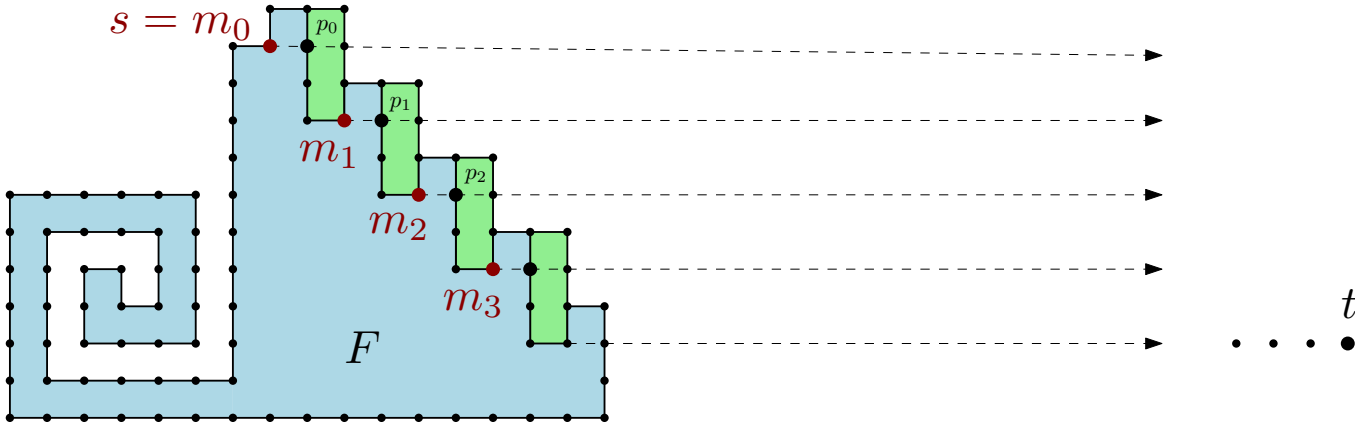
- Eigentlich ist Facettenrouting immer nur „Plan B“
  - Greedy Routing schneller, weniger fehleranfällig
  - Greedy Routing kann alle Verbindungen nutzen
- Verbinde Greedy Routing und Facettenrouting
  1. Route greedy, bis Paket an Knoten p steckenbleibt
  2. Starte adaptives Facettenrouting für Start p, Ziel t
  3. Stoppe, sobald *eine Facette* passiert ist
  4. Gehe zurück zu 1.

Satz

Greedy AFR ist nicht asymptotisch worst-case-optimal.

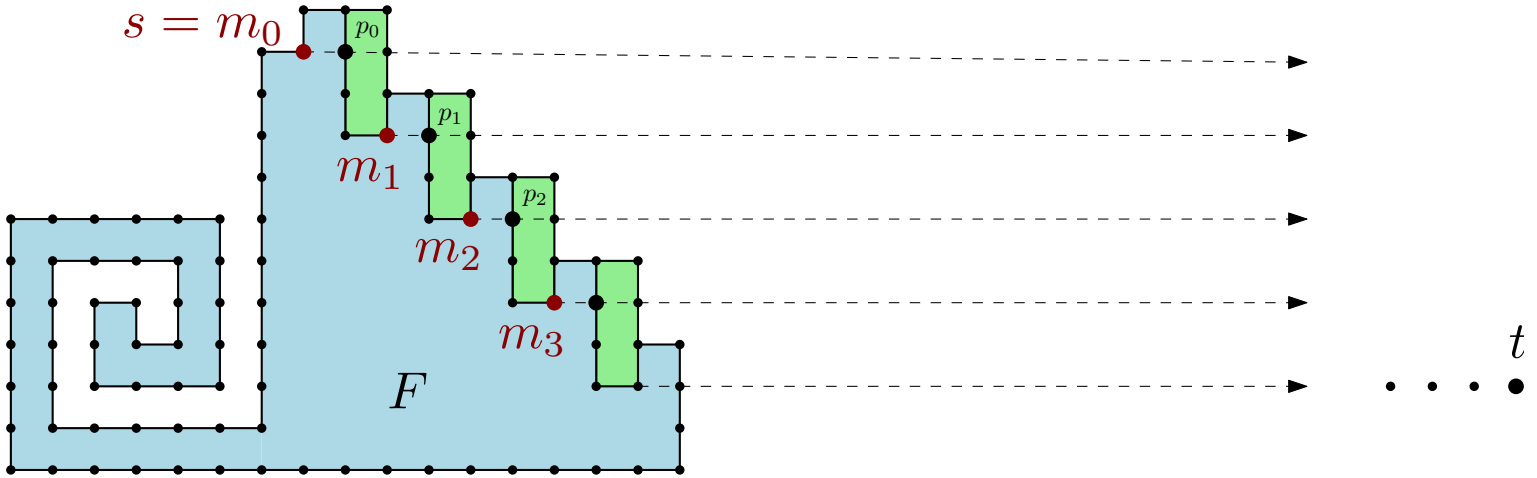
Suboptimalität Greedy+AFR

- Erinnerung:
  - AFR zerfällt in BFR-Schritte verschiedener Ellipsen.
  - In jedem BFR-Durchgang wird jede Facette nur einmal besucht
  - Aufwand linear in der Anzahl der Knoten (der Fläche der Ellipse)
- Greedy-Schritte zerstören diese Eigenschaft!
- Beweisskizze (betrachte „passende“ Ellipse $\mathcal{E}$):
  - Instanz mit Kosten $\in \Theta(c_k^3)$:



  - Facettenrouting startet in lokalen Minima $m_0, m_1, \ldots$: $\Theta(c_k)$ mal
  - Facette F enthält $\Theta(c_k^2)$ Knoten.

Zurück zum Facettenrouting

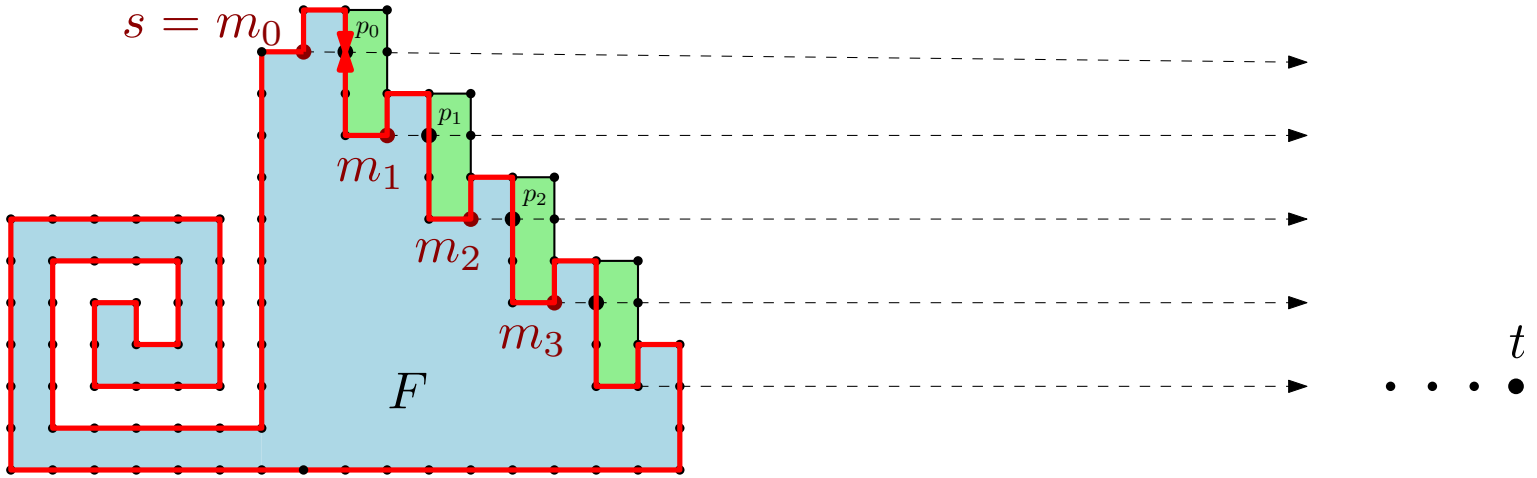


- **AFR** von m_0 auf F; gehe zu p_0; wechsle die Facette
- **Greedy** von p_0 zu m_1; $\overrightarrow{m_1 t}$ zeigt in F; ...

- Selbst „bester“ Schnittpunkt p_i von $\overrightarrow{m_i t}$ mit Facettenrand kein guter Startpunkt für Greedy-Routing.
- Greedy-Routing trifft immer wieder auf dieselbe Facette.
- Facettenrouting erkundet immer wieder dieselbe Facette.

Facettenrouting lässt sich so anpassen, dass das nicht passiert.

Zurück zum Facettenrouting

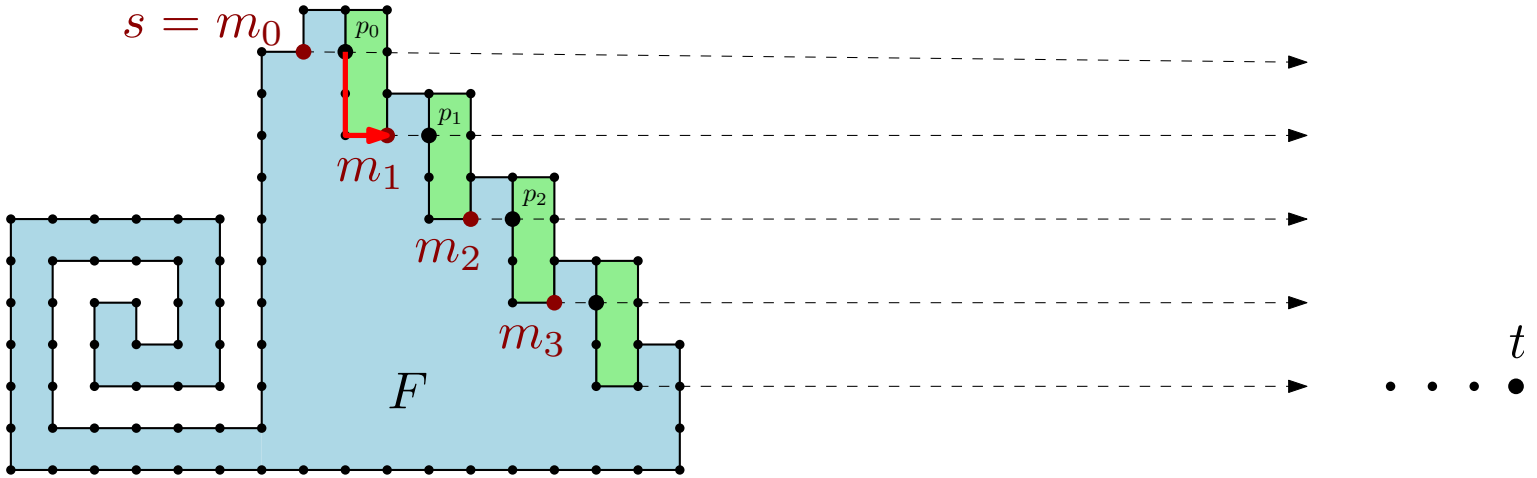


- **AFR** von m_0 auf F; gehe zu p_0; wechsle die Facette
- **Greedy** von p_0 zu m_1; $\overrightarrow{m_1 t}$ zeigt in F; ...

- Selbst „bester“ Schnittpunkt p_i von $\overrightarrow{m_i t}$ mit Facettenrand kein guter Startpunkt für Greedy-Routing.
- Greedy-Routing trifft immer wieder auf dieselbe Facette.
- Facettenrouting erkundet immer wieder dieselbe Facette.

Facettenrouting lässt sich so anpassen, dass das nicht passiert.

Zurück zum Facettenrouting

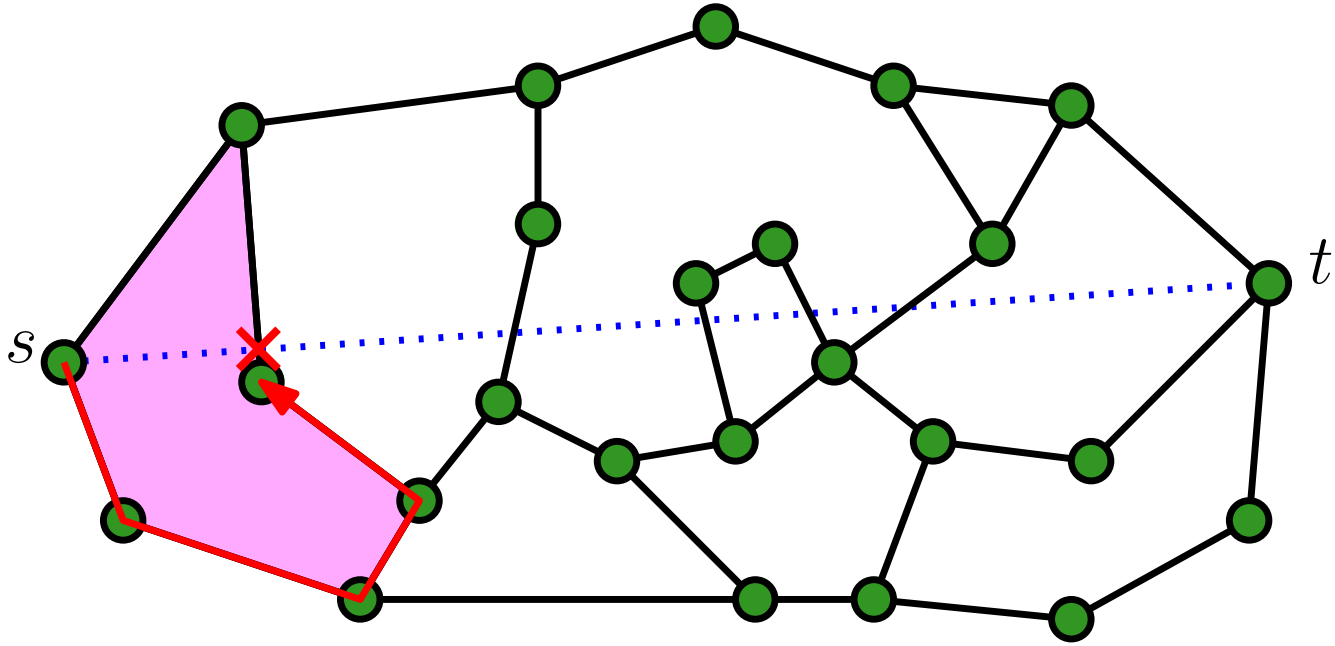


- **AFR** von m_0 auf F; gehe zu p_0; wechsle die Facette
- **Greedy** von p_0 zu m_1; $\overrightarrow{m_1 t}$ zeigt in F; ...

- Selbst „bester“ Schnittpunkt p_i von $\overrightarrow{m_i t}$ mit Facettenrand kein guter Startpunkt für Greedy-Routing.
- Greedy-Routing trifft immer wieder auf dieselbe Facette.
- Facettenrouting erkundet immer wieder dieselbe Facette.

Facettenrouting lässt sich so anpassen, dass das nicht passiert.

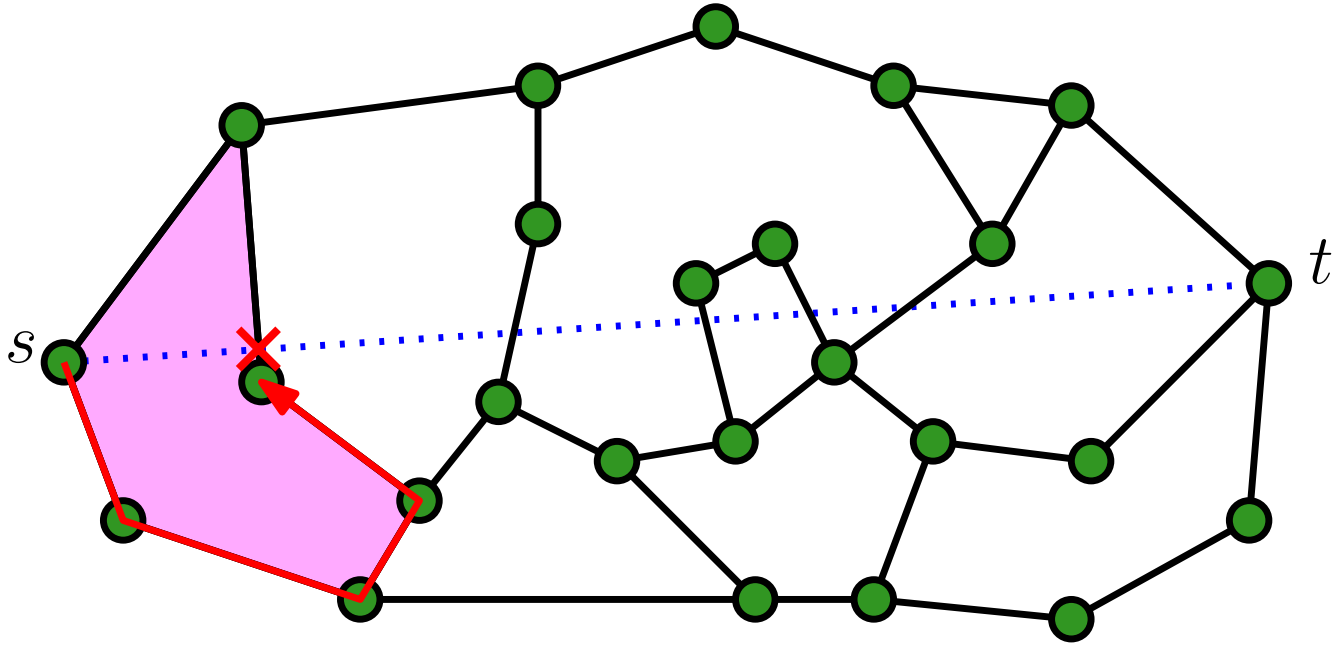
Optimiertes Facettenrouting (OFR)



Herkömmliches Facettenrouting

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

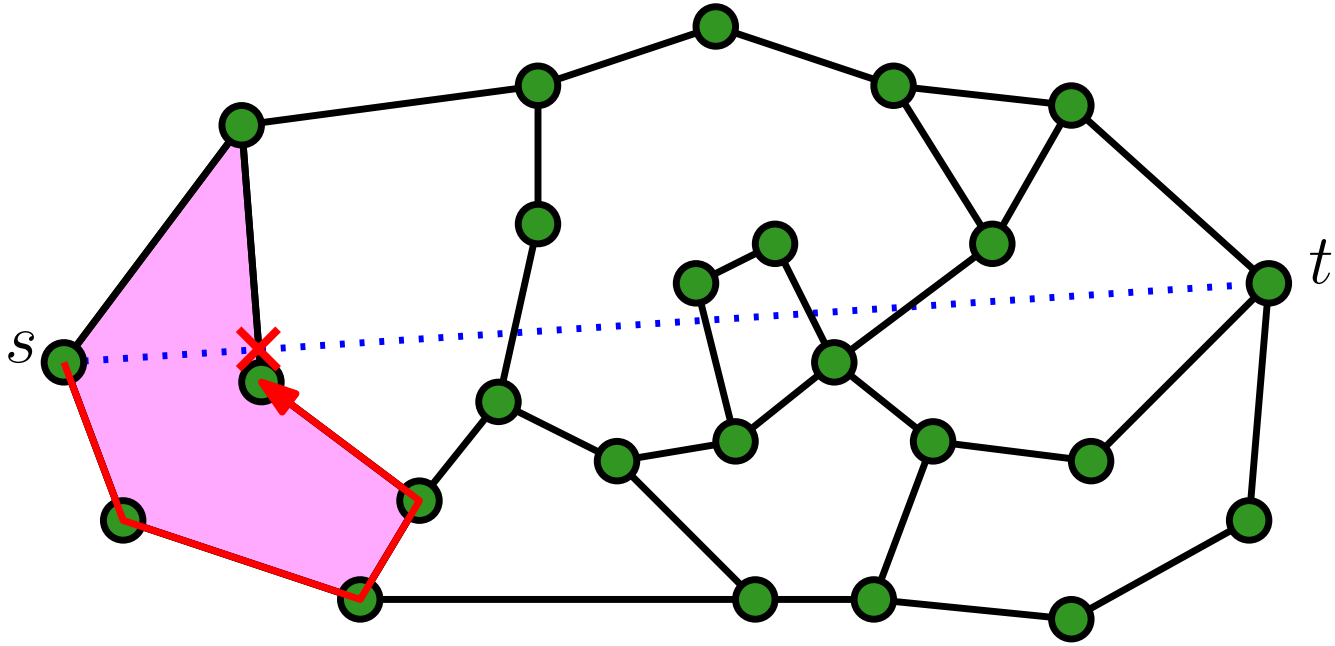
Optimiertes Facettenrouting (OFR)



Optimiertes Facettenrouting [Wattenhofer et al. ’03]

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

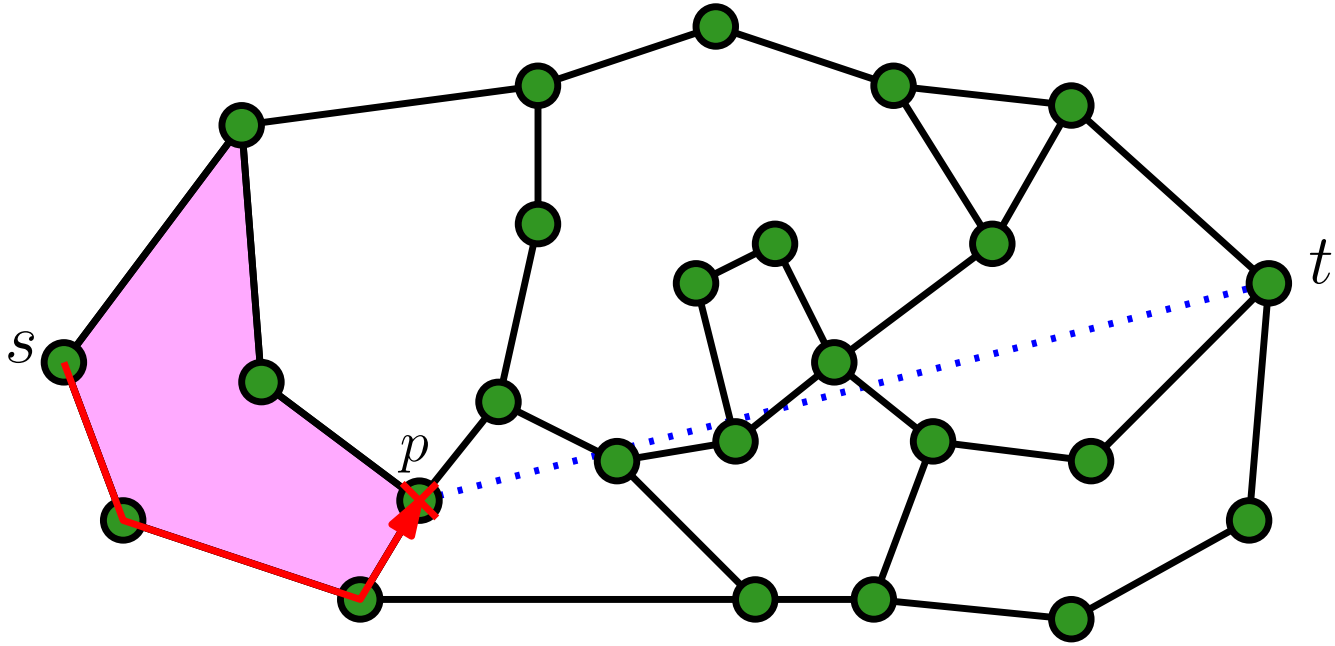
Optimiertes Facettenrouting (OFR)



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

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

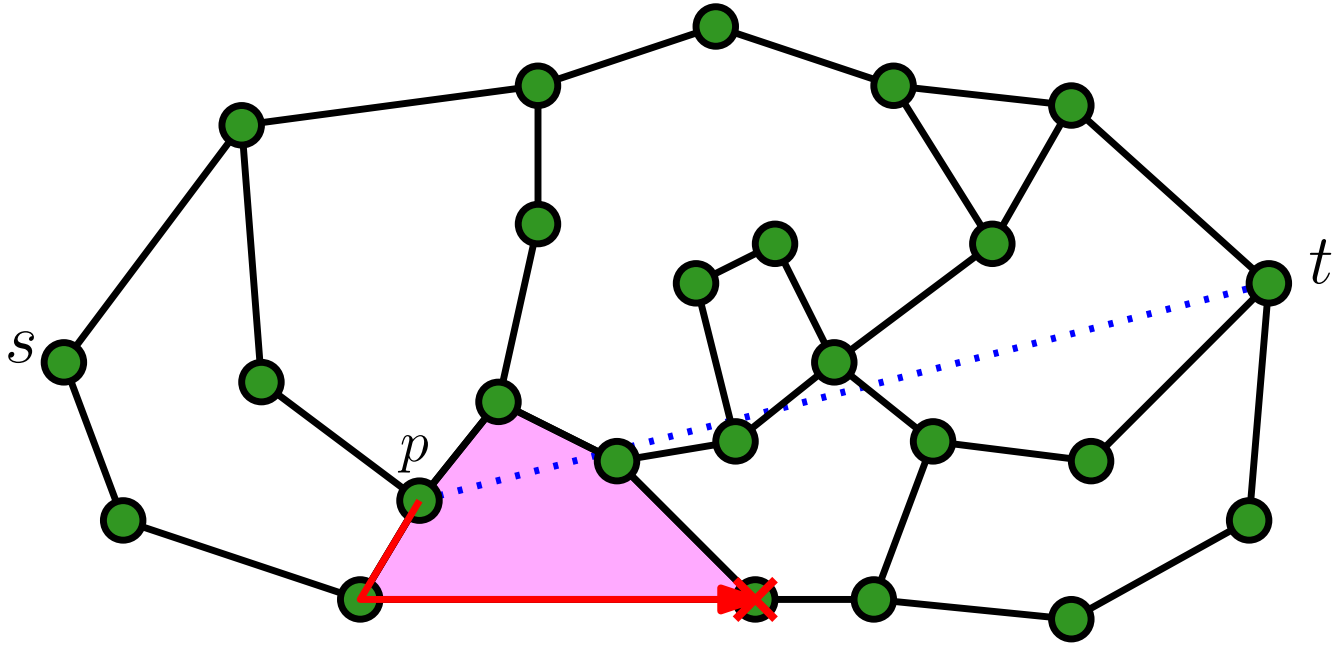
Optimiertes Facettenrouting (OFR)



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

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

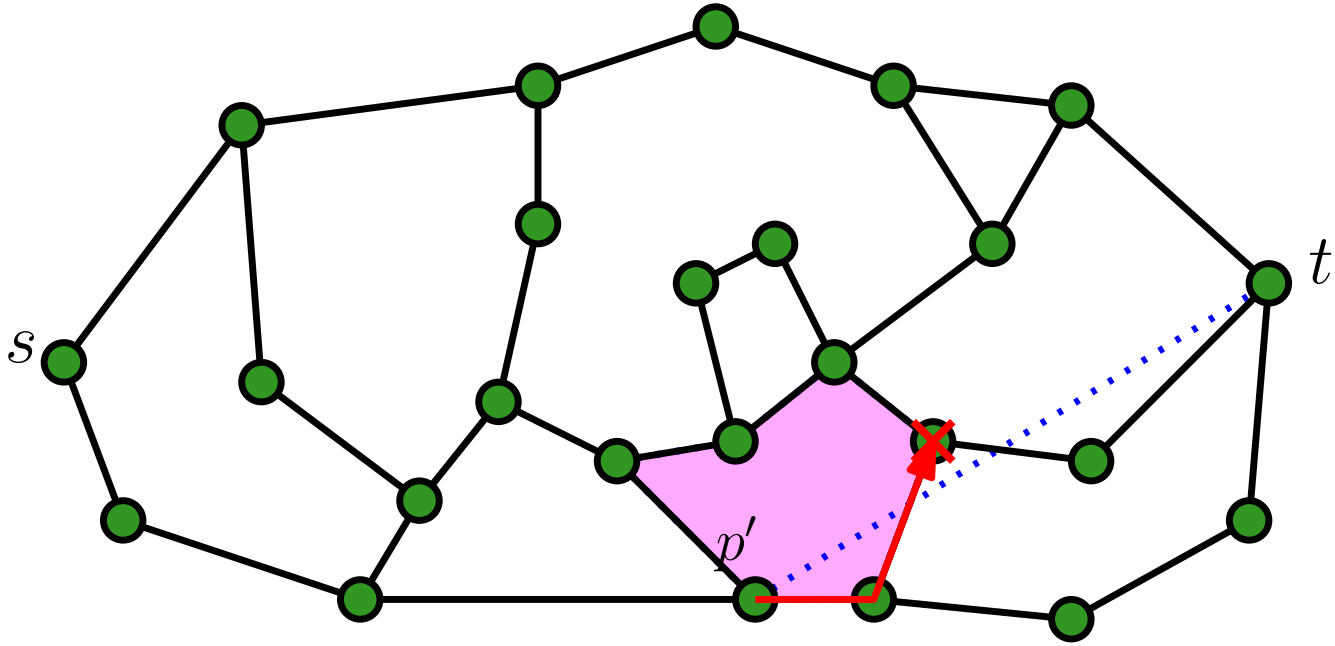
Optimiertes Facettenrouting (OFR)



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

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

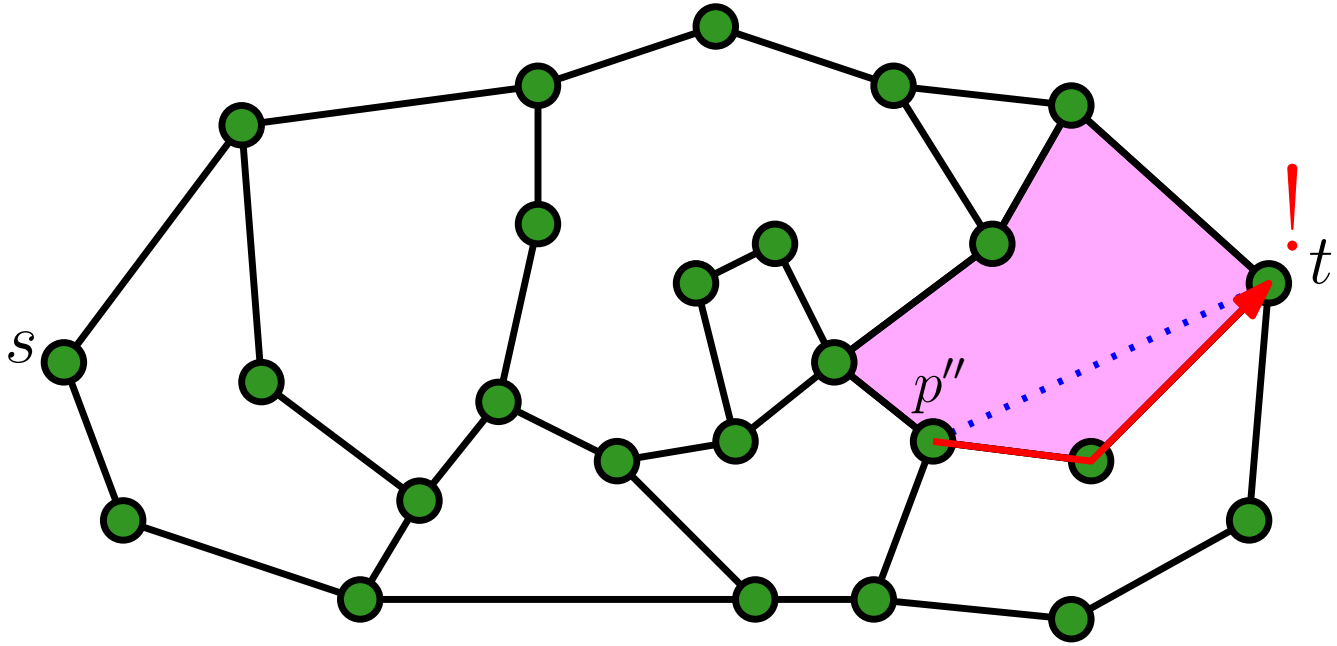
Optimiertes Facettenrouting (OFR)



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

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

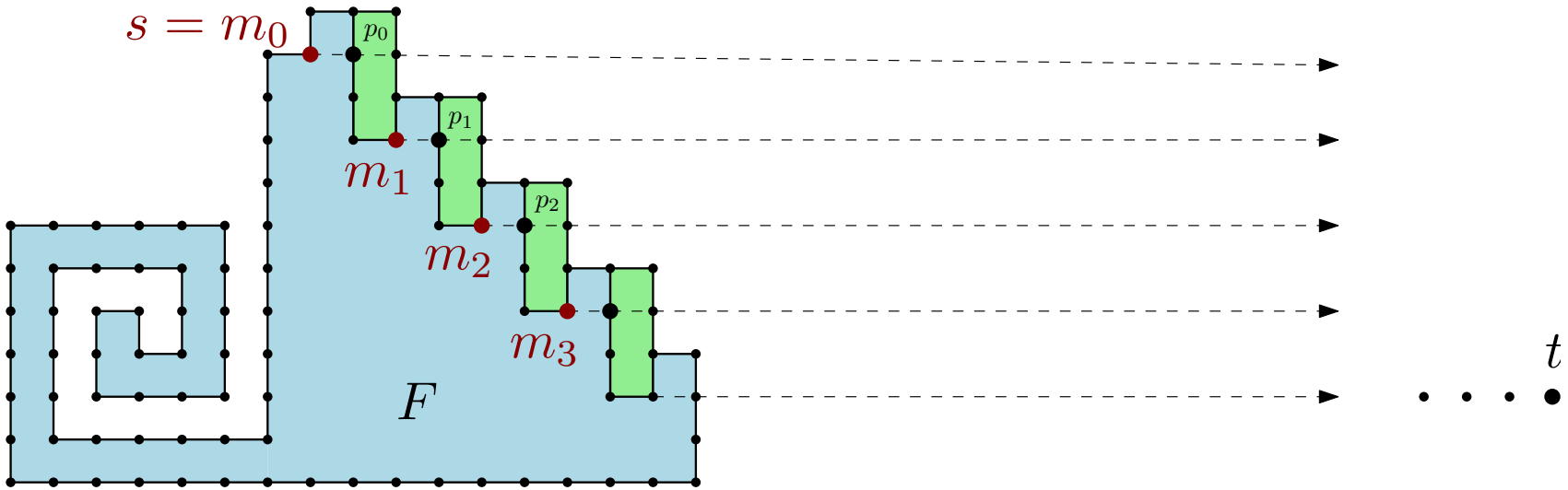
Optimiertes Facettenrouting (OFR)



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

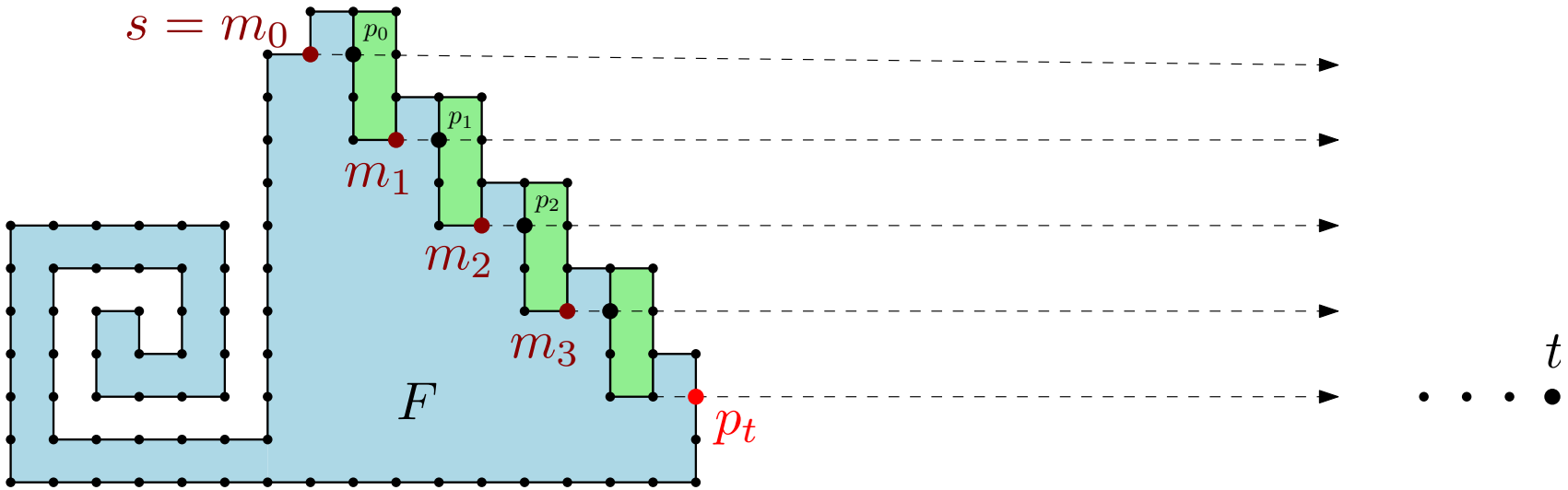
- Erkunde Facette, in die $\overrightarrow{st}$ zeigt
- Kehre zum **Knoten** p zurück, der t am nächsten ist (ohne Bew.)
- Wechsle zur angrenzenden Facette, in die $\overrightarrow{pt}$ zeigt.

Greedy+Optimiertes Facettenrouting (GOFR)



- Greedy Routing stockt bei m_0
- Facettenrouting endet nicht in Schnitt von $\overrightarrow{m_0 t}$ mit Facettenrand, sondern bei p_t
- Greedy Routing kommt nicht wieder an den Rand von F.

Greedy+Optimiertes Facettenrouting (GOFR)



- Greedy Routing stockt bei m_0
- Facettenrouting endet nicht in Schnitt von $\overrightarrow{m_0 t}$ mit Facettenrand, sondern bei p_t
- Greedy Routing kommt nicht wieder an den Rand von F.

Optimiertes adaptives FR (OAFR)

Satz

Für das optimierte Facettenrouting gelten alle Aussagen bis auf die Suboptimalität in Verbindung mit Greedy-Routing analog.

- Optimiertes FR findet in $O(n)$ Schritten zum Ziel.
- Beschränktes optimiertes Facettenrouting mit geschätzter Pfadlänge c findet in c^2 Schritten zum Ziel oder erkennt fehlenden Zusammenhang.
- Adaptives optimiertes FR findet in $c^2(p^\star)$ zum Ziel.
- Adaptives optimiertes FR erkennt fehlenden Zusammenhang in $O(n' \log n')$ Schritten.

Greedy + Optimiertes adaptives FR (GOAFR)

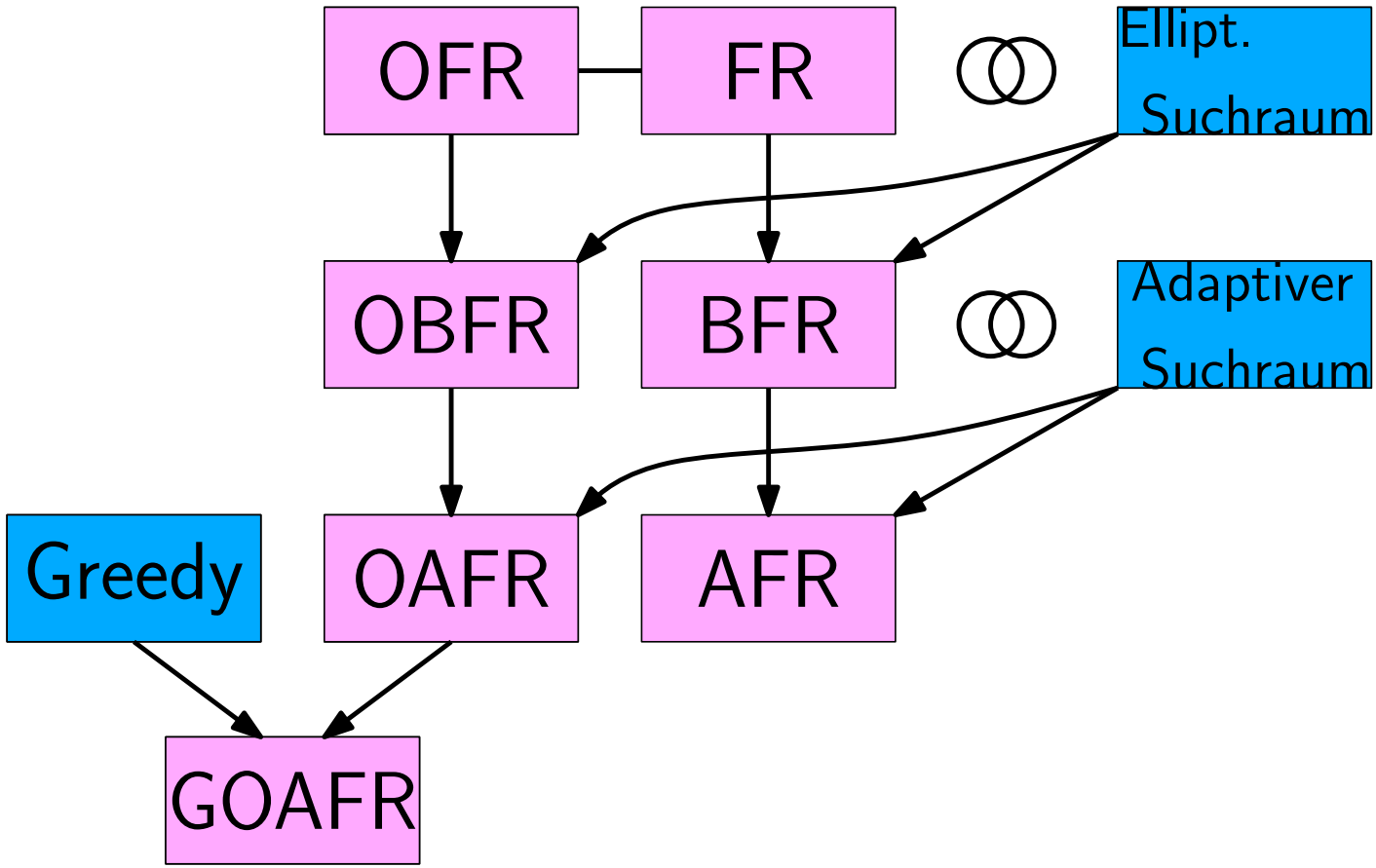
Greedy + Optimiertes adaptives Facettenrouting

1. Route greedy, bis Paket an Knoten p steckenbleibt
2. Starte optimiertes, adaptives FR für Start p, Ziel t
3. Stoppe, sobald *eine Facette* passiert ist
4. Gehe zurück zu 1.

Satz

GOAFR ist asymptotisch worst-case-optimal. (Ohne Beweis)

Überblick der Zusammenhänge



Denkanstoß

Können wir irgendwas davon noch in 3D verwenden?

- Modelle?
- Greedy Routing?
- Facettenrouting?
- Schranken?

Empfehlung weiterführende Literatur

F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger:
Geometric Ad-Hoc Routing: Of Theory and Practice.
In: Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC '03), 2003.

- GOAFR+ Algorithmus
  - asymptotisch worst-case optimal
  - kommt ohne $\Omega(1)$-Modell Annahme aus
- Behandlung verschiedener Kostenfunktionen

Was mitnehmen?

- Neue Modelle!
  - UDG: stark idealisiert, aber sehr mächtig
  - QUDG: etwas allgemeiner, immer noch idealisiert
  - Gabriel Graph: planar und lokal berechenbar
  - $\Omega(1)$-Modell: vernünftige Annahme zur Analyse!
- Zahlreiche Ansätze zum Georouting
  - Greedy Routing
  - Facettenrouting auf planaren Graphen in vielen Varianten
    - Kombinationen verschiedener Ansätze

Literatur zur Vorlesung

1. E. Kranakis, H. Singh, J. Urrutia: *Compass Routing on Geometric Networks*. In Proceedings of the 11th Canadian Conference on Computational Geometry, 1999

2. F. Kuhn, R. Wattenhofer, A. Zollinger: *Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing*. In: Proceedings of the 4th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc '03), 2003

3. P. Santi: *Topology Control in Wireless Ad Hoc and Sensor Networks*. Wiley, 2005