Karlsruher Institut für Technologie

Jahrestagung GRK1194 — Oktober 2010

Dennis Schieferdecker – *dennis.schieferdecker@kit.edu*

Institut für Theoretische Informatik - Algorithmik II

Randerkennung & Routing

Karlsruher Institut für Technologie

- I -

Randerkennung

in Sensornetzen

Dennis Schieferdecker:

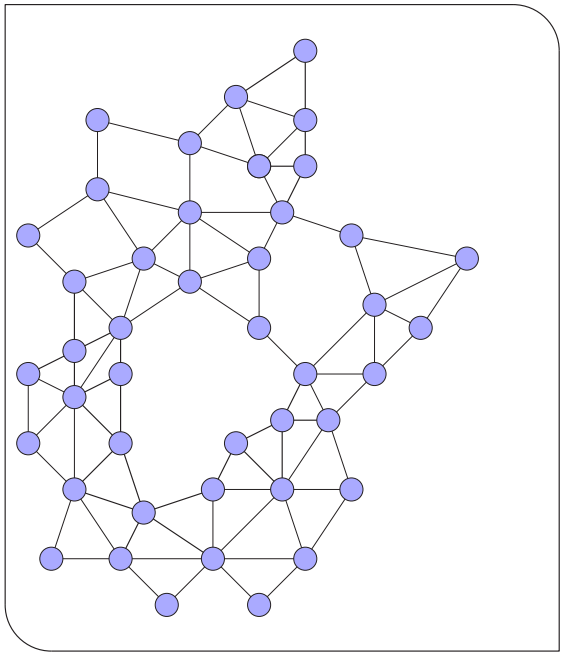
Institut für Theoretische Informatik
Algorithmik II

Einführung

Randerkennung

Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand $\rightarrow$ Ausdehnung
  innerer Rand $\rightarrow$ Loch



Anwendungsgebiete

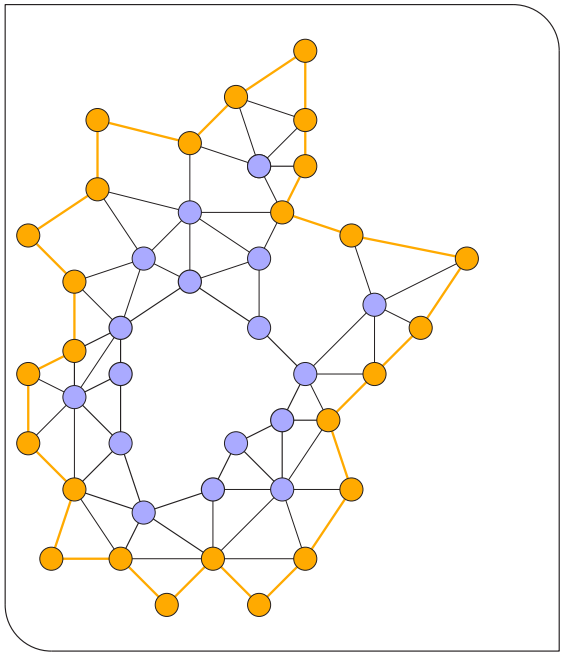
- Erkennen von Eindringlingen
- effizientes Routing
- Indikatoren für nicht ausreichende Überdeckung / Verbindungen

Einführung

Randerkennung

Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand $\rightarrow$ Ausdehnung
  innerer Rand $\rightarrow$ Loch



Anwendungsgebiete

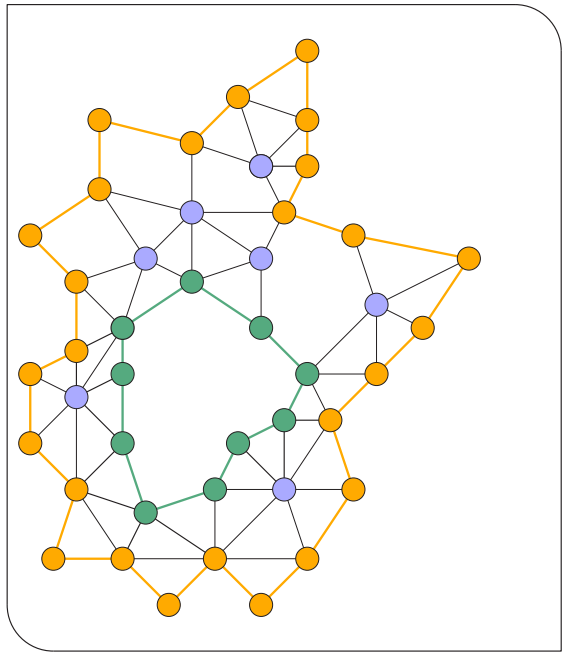
- Erkennen von Eindringlingen
- effizientes Routing
- Indikatoren für nicht ausreichende Überdeckung / Verbindungen

Einführung

Randerkennung

Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand $\rightarrow$ Ausdehnung
  innerer Rand $\rightarrow$ Loch



Anwendungsgebiete

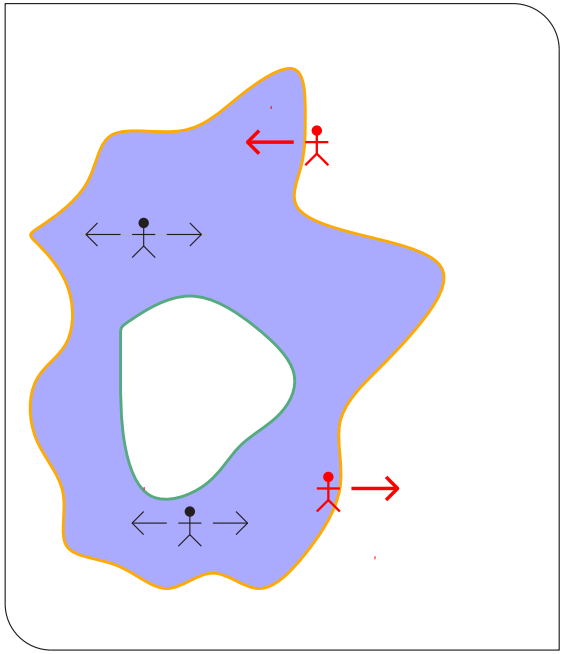
- Erkennen von Eindringlingen
- effizientes Routing
- Indikatoren für nicht ausreichende Überdeckung / Verbindungen

Einführung

Randerkennung

Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand $\rightarrow$ Ausdehnung
  innerer Rand $\rightarrow$ Loch



Anwendungsgebiete

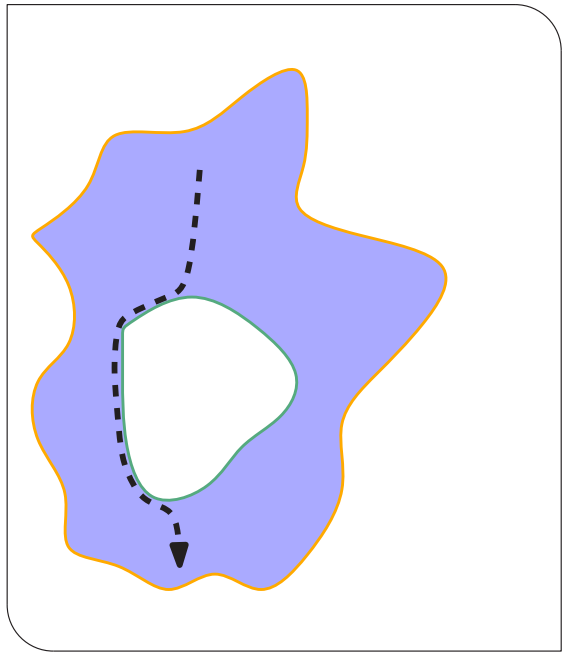
- Erkennen von Eindringlingen
- effizientes Routing
- Indikatoren für nicht ausreichende Überdeckung / Verbindungen

Einführung

Randerkennung

Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand $\rightarrow$ Ausdehnung
  innerer Rand $\rightarrow$ Loch



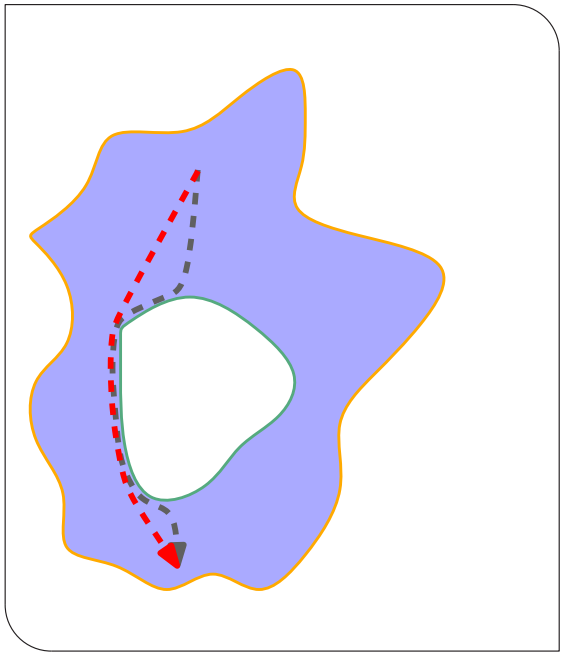
Anwendungsgebiete

- Erkennen von Eindringlingen
- effizientes Routing
- Indikatoren für nicht ausreichende Überdeckung / Verbindungen

Einführung
Randerkennung

Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand $\rightarrow$ Ausdehnung
  innerer Rand $\rightarrow$ Loch



Anwendungsgebiete

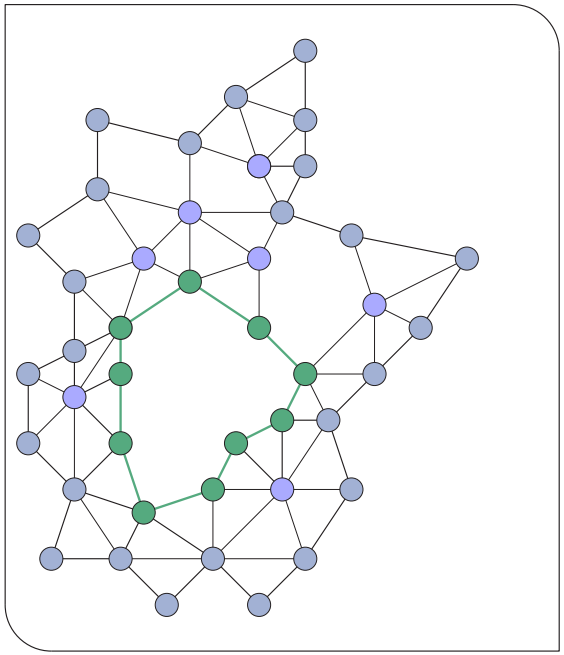
- Erkennen von Eindringlingen
- effizientes Routing
- Indikatoren für nicht ausreichende Überdeckung / Verbindungen

Einführung

Randerkennung

Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand $\rightarrow$ Ausdehnung
  innerer Rand $\rightarrow$ Loch



Anwendungsgebiete

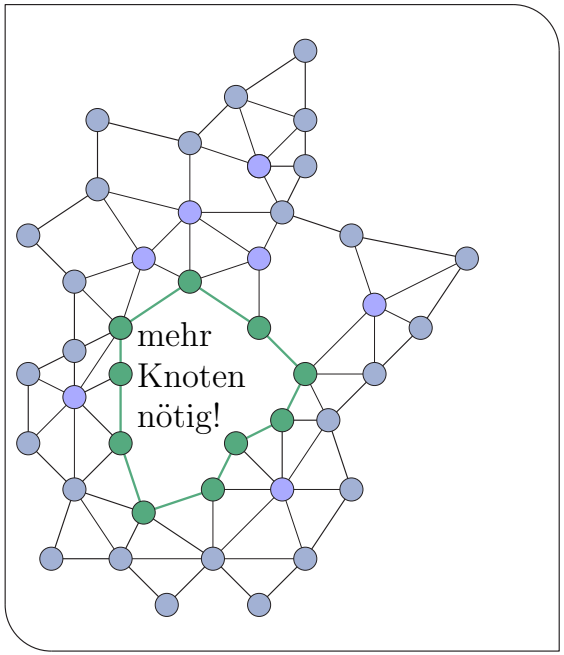
- Erkennen von Eindringlingen
- effizientes Routing
- Indikatoren für nicht ausreichende Überdeckung / Verbindungen

Einführung

Randerkennung

Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand $\rightarrow$ Ausdehnung
  innerer Rand $\rightarrow$ Loch



Anwendungsgebiete

- Erkennen von Eindringlingen
- effizientes Routing
- Indikatoren für nicht ausreichende Überdeckung / Verbindungen

Einführung

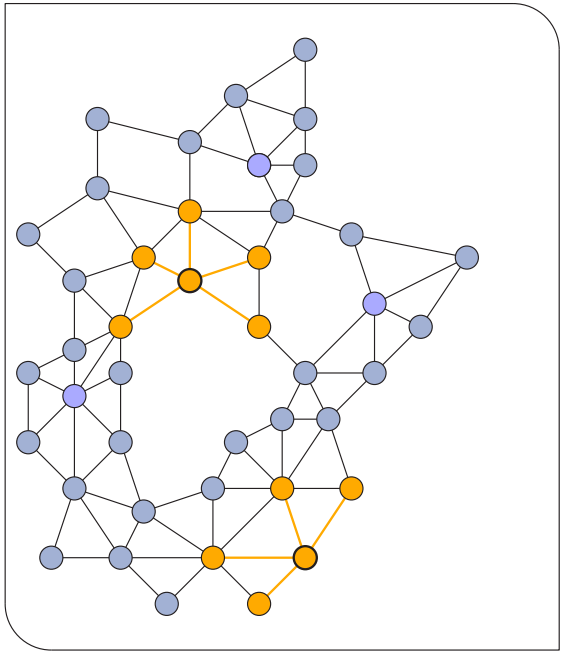
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad

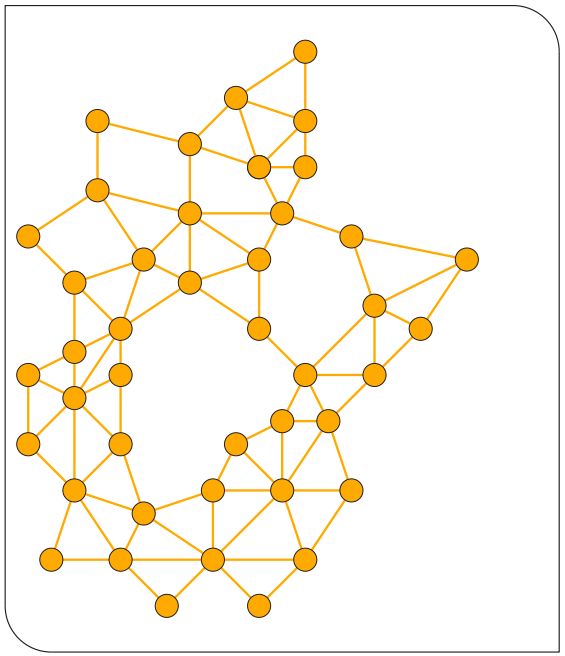


Einführung

bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)



verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad

Einführung

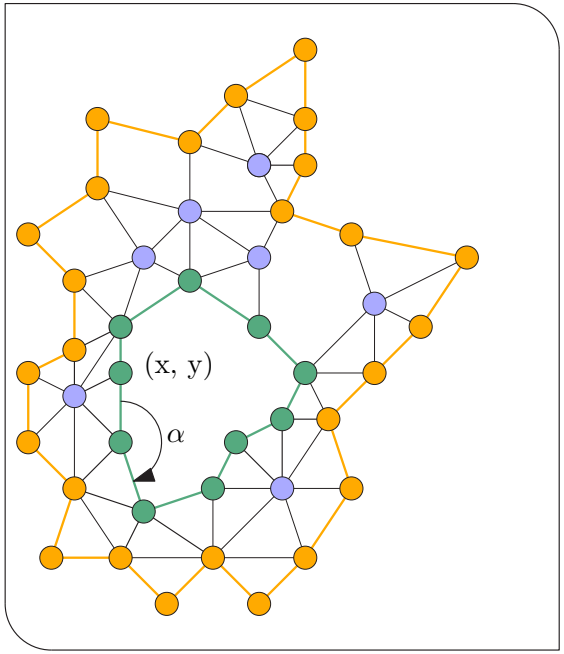
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

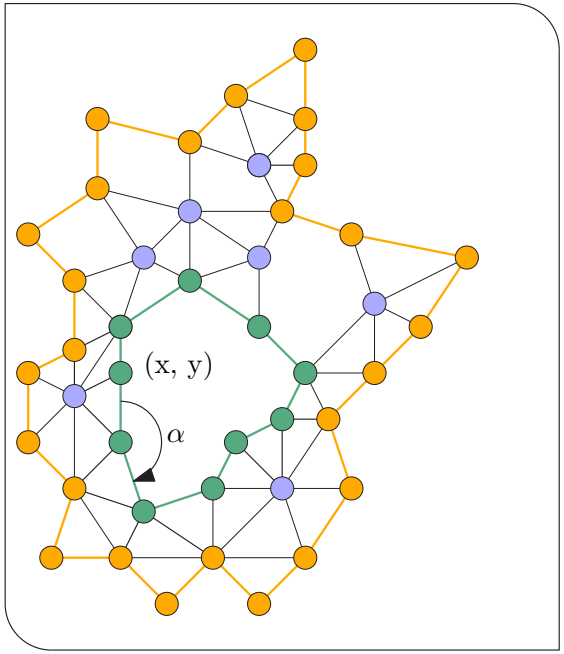
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

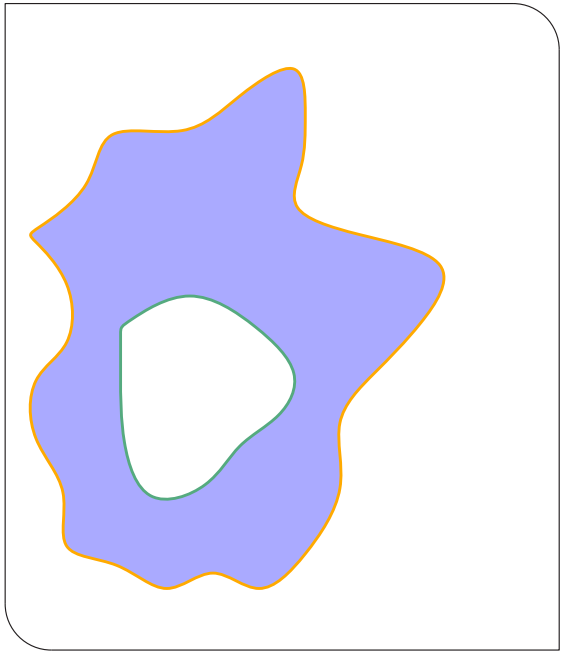
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

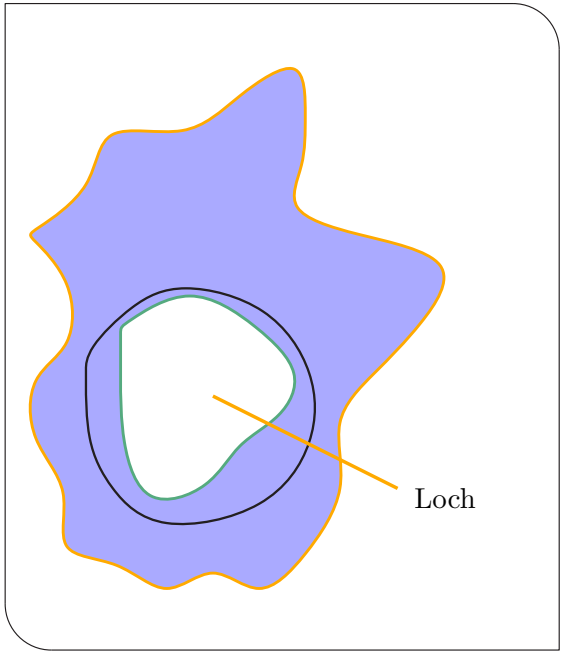
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

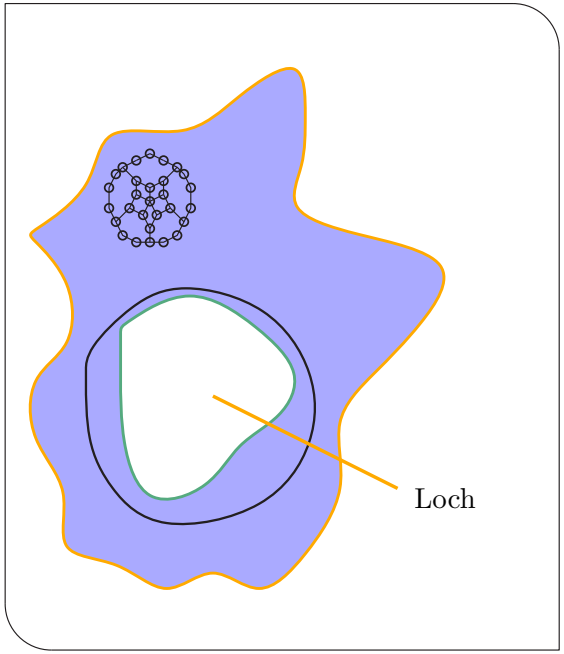
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

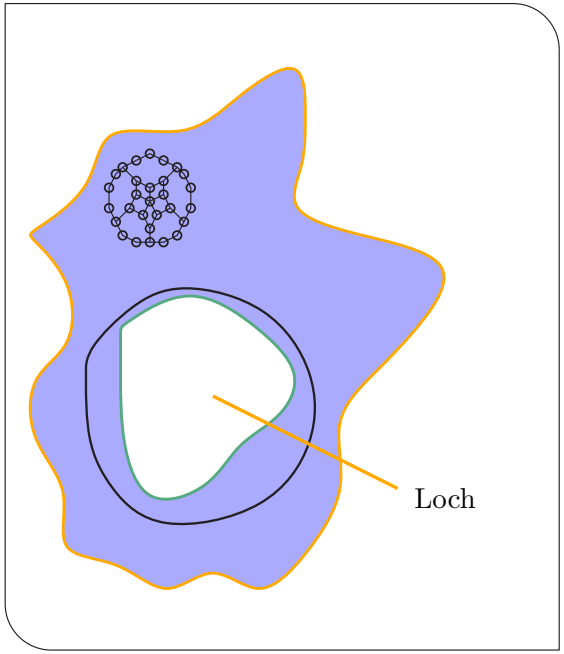
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

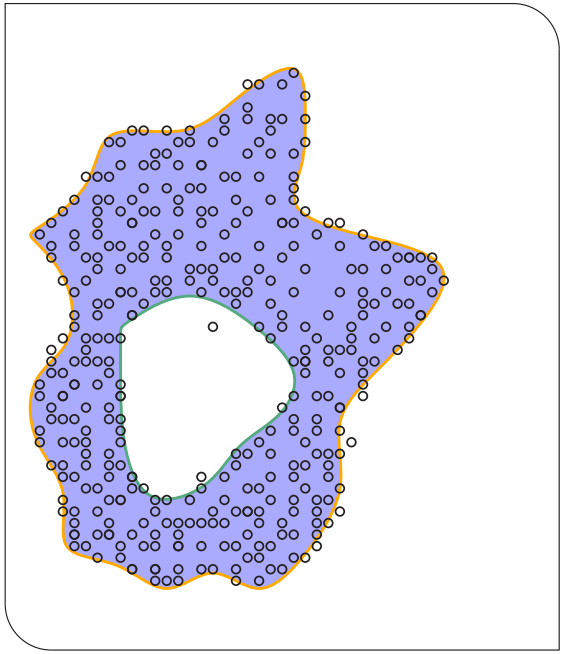
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

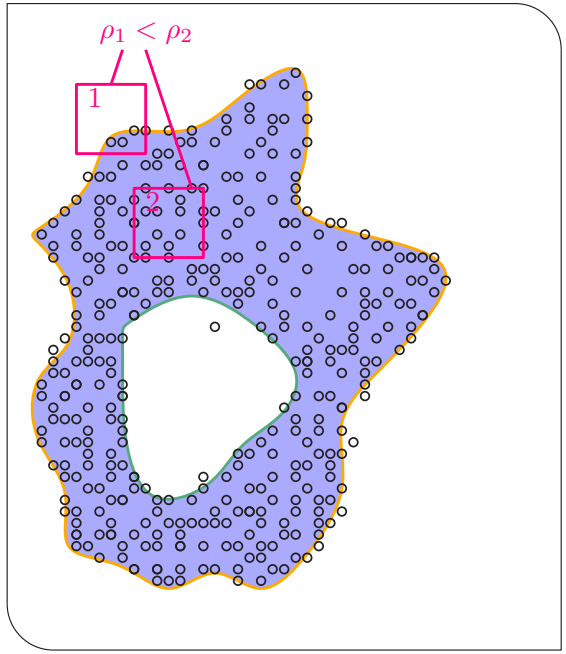
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

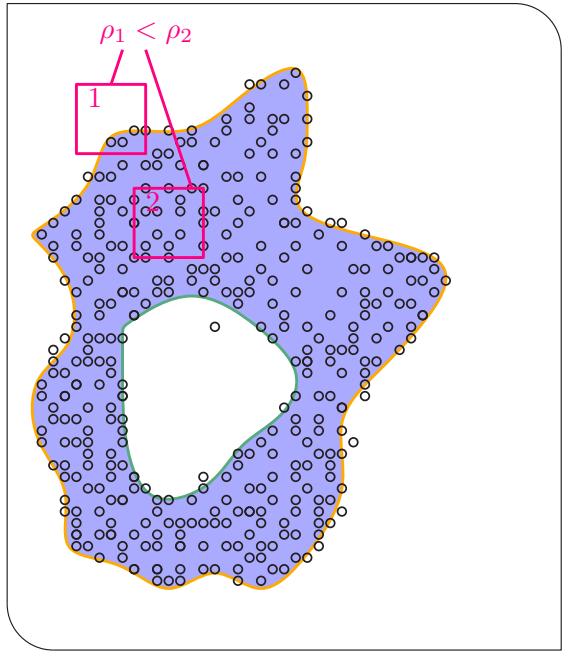
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad



Einführung

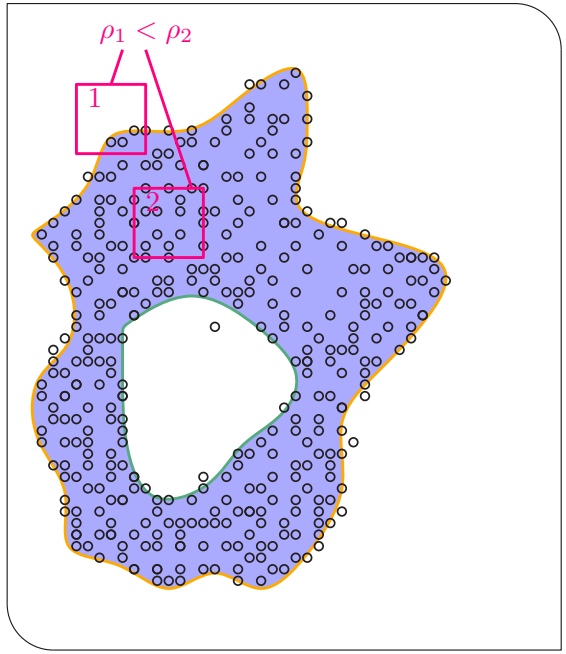
bestehende Lösungsansätze

grundlegende Unterteilung

- lokale / globale Verfahren (verteilt / zentral)

verschiedene Strategien

- geometrisch
  - Positions- oder Winkelinformationen
  - Daten selten vorhanden
- topologisch
  - Verbindungsinformation bzw. Graphstruktur
  - aufwändige Verfahren
- statistisch
  - Knoten- oder Verbindungsdichte
  - benötigt hohen, gleichmäßigen Grad

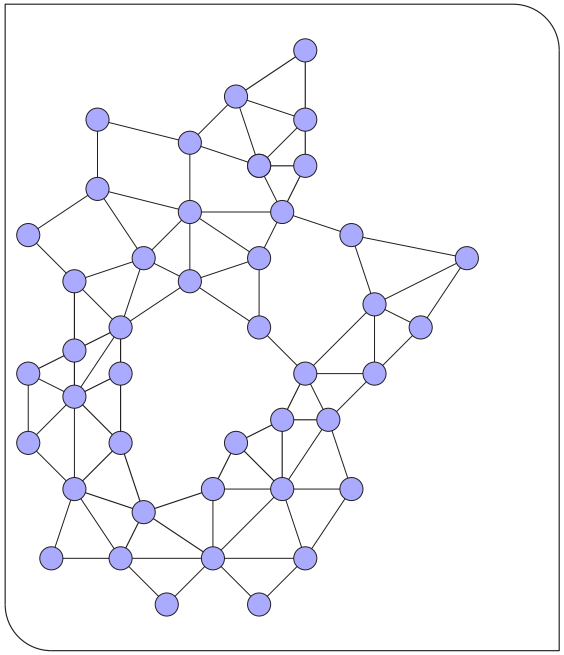


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

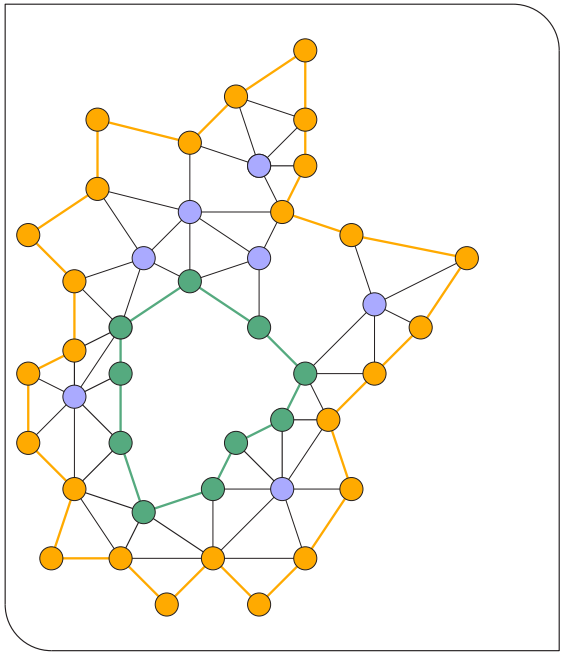


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

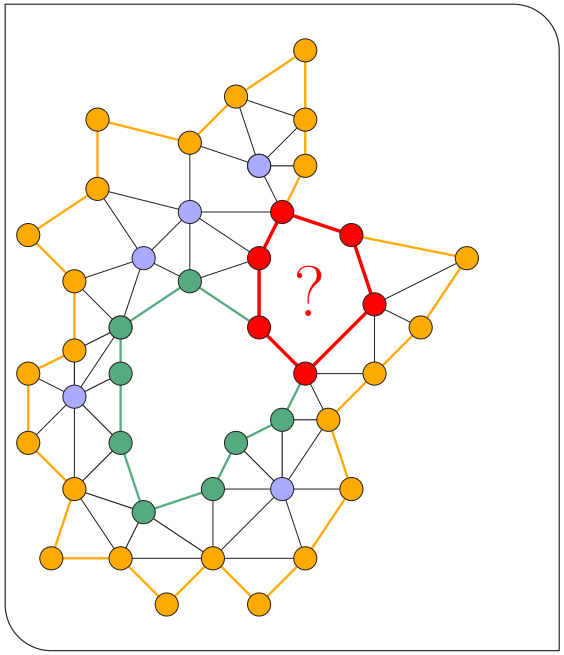


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

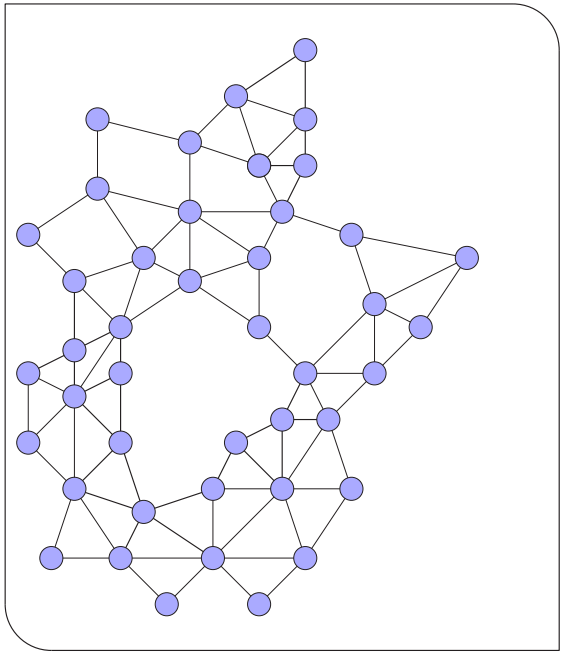


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

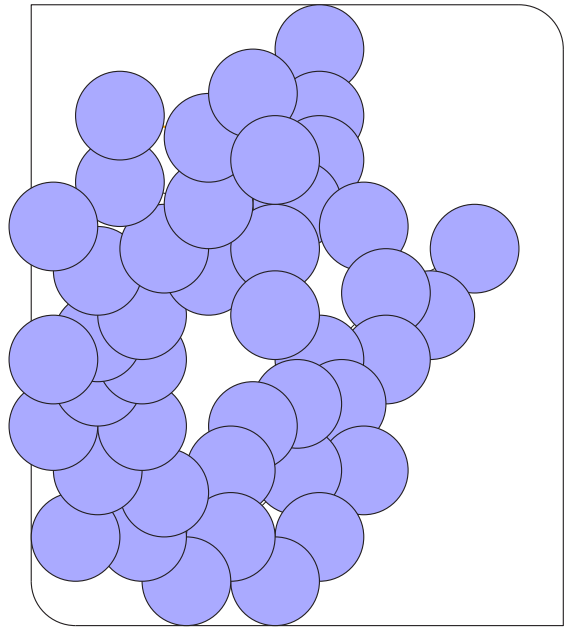


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

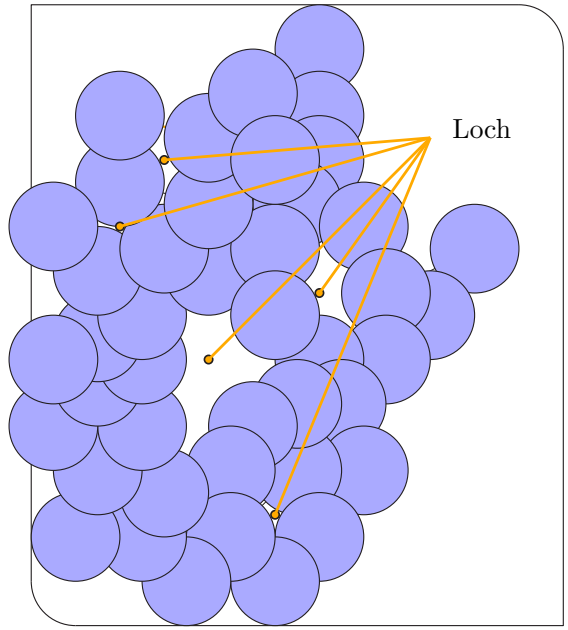


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

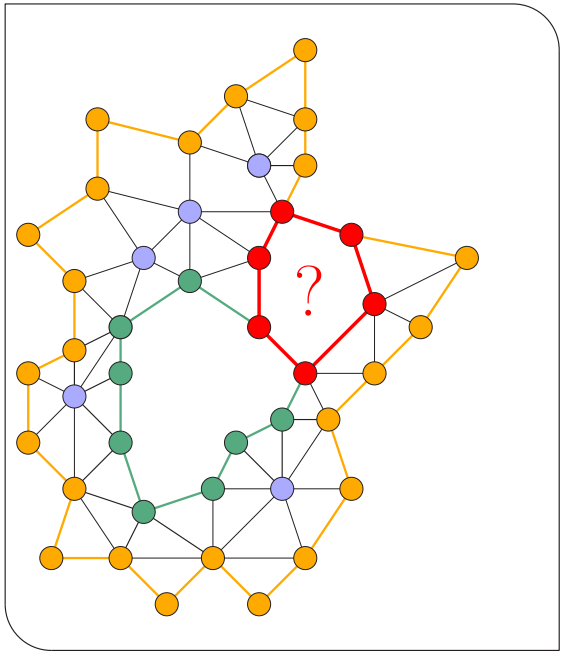


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

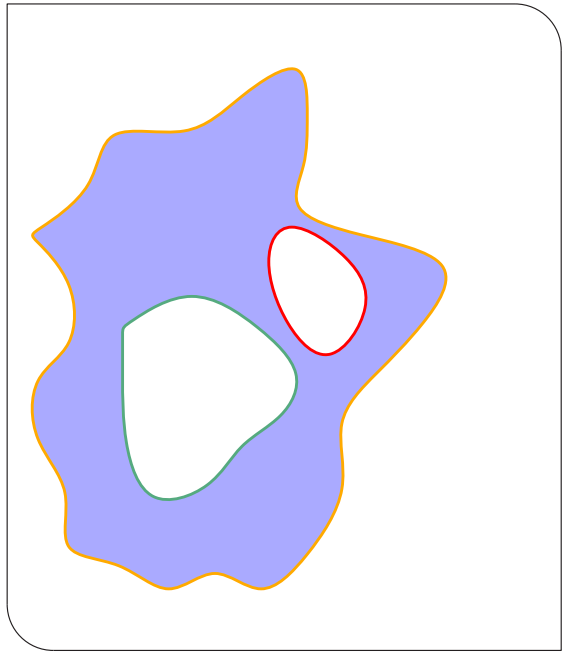


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

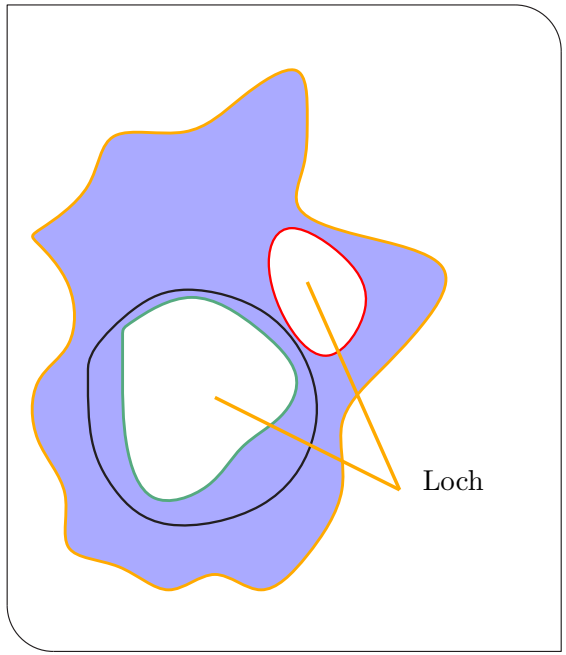


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

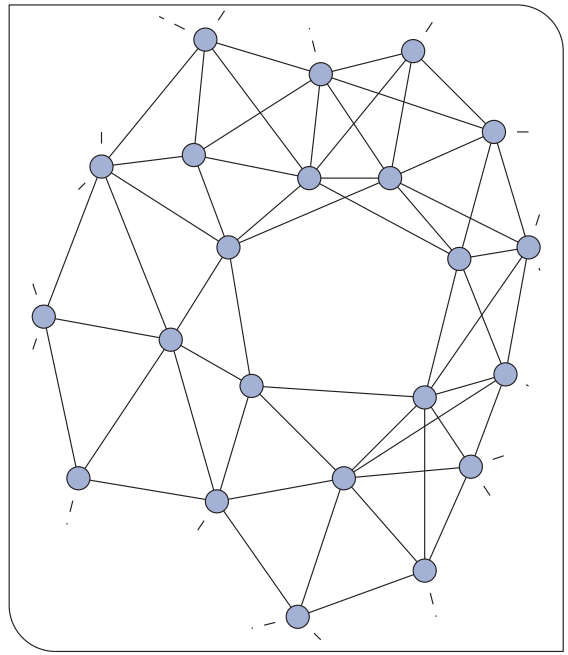


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen
    (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

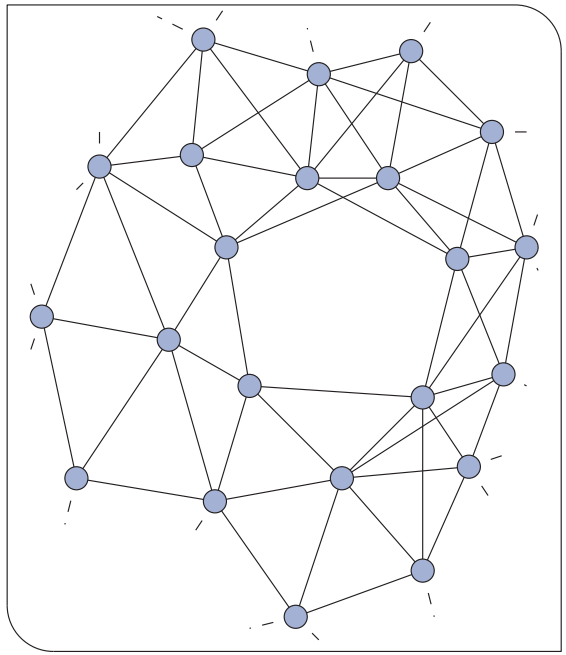


Einführung

Loch- und Randdefinitionen

Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen
- praktisch motiviert
  - ”weiße Flächen“
- topologisch motiviert
  - kontinuierliche Betrachtung
  - Einbettung des Verbindungsgraphen (gegebene / irgendeine korrekte Einbettung)
- wichtig für quantitative Analyse
- wichtig für quantitative Analyse

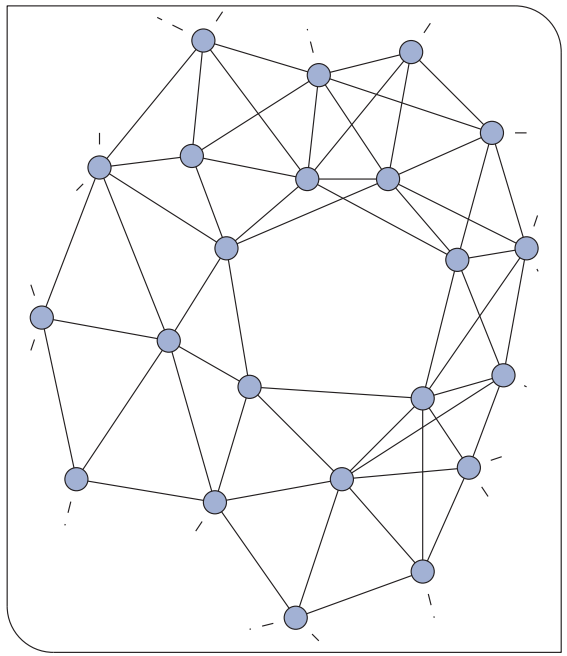


Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min} $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

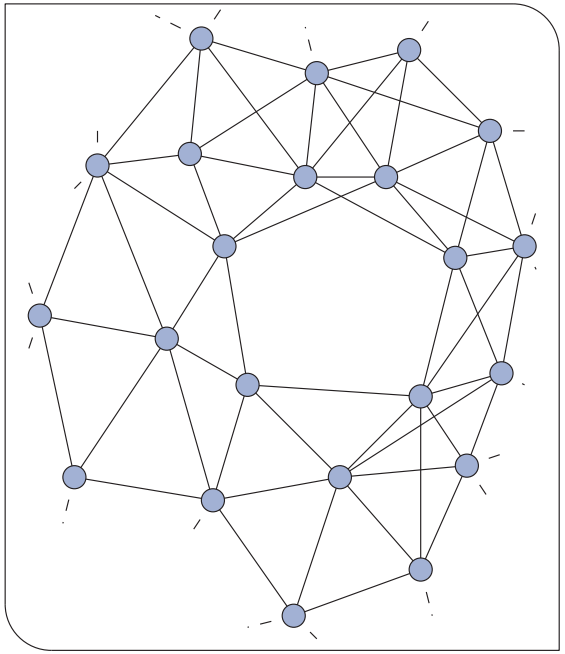
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min} $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

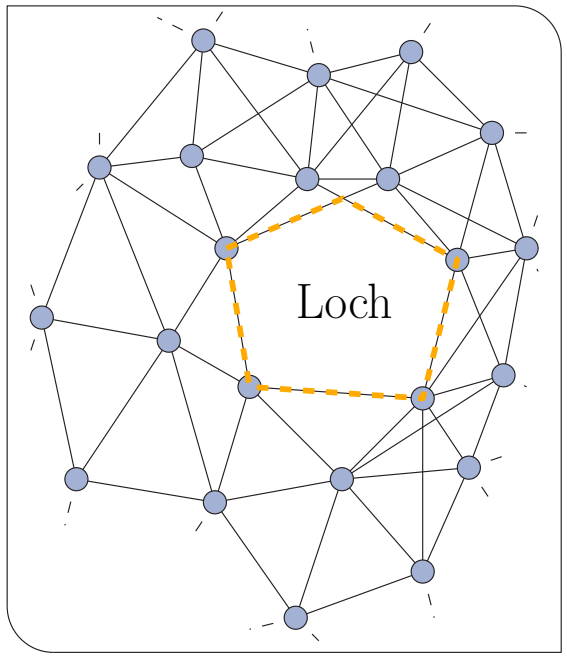
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min} $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

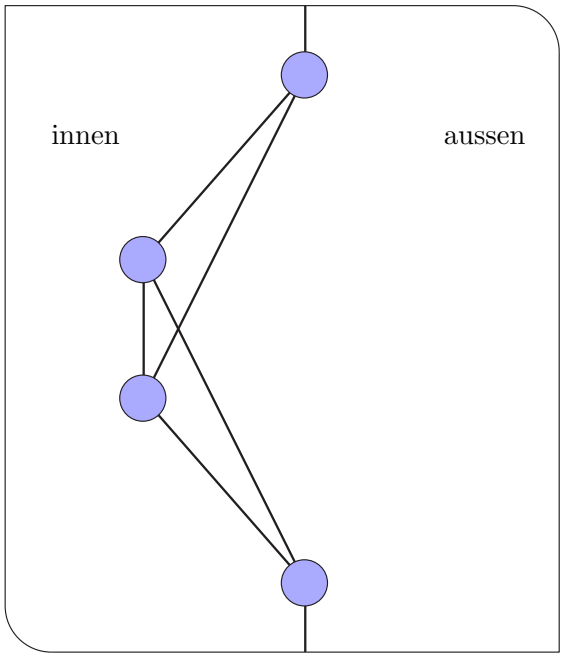
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min}
  $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

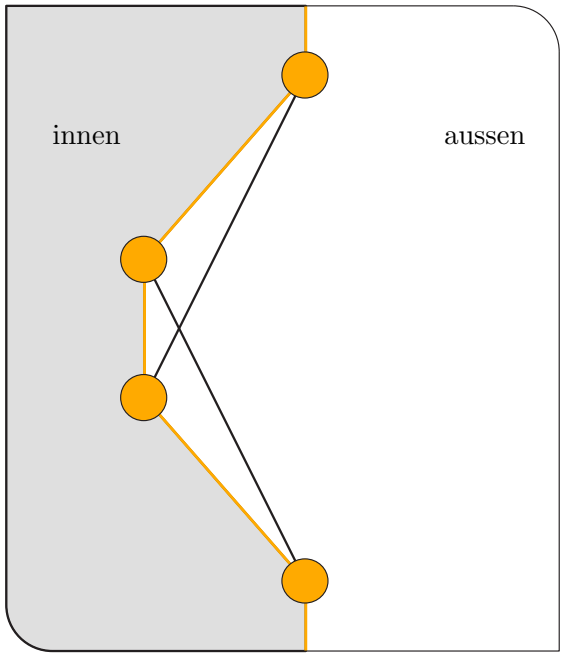
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min}
  $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

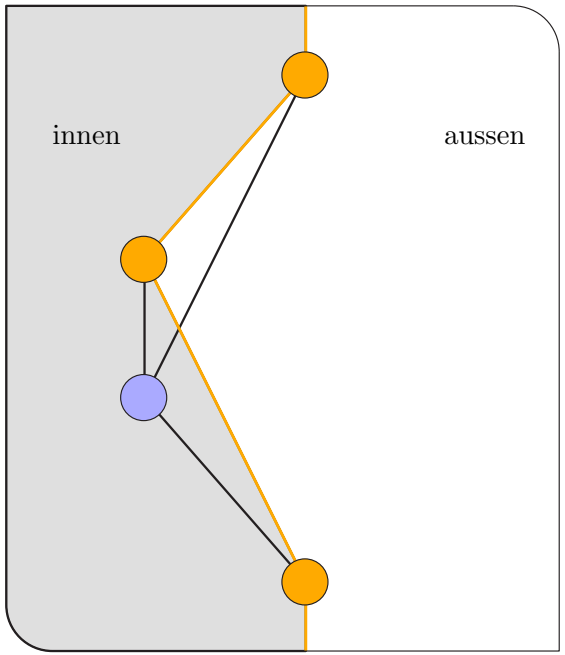
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min}
  $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

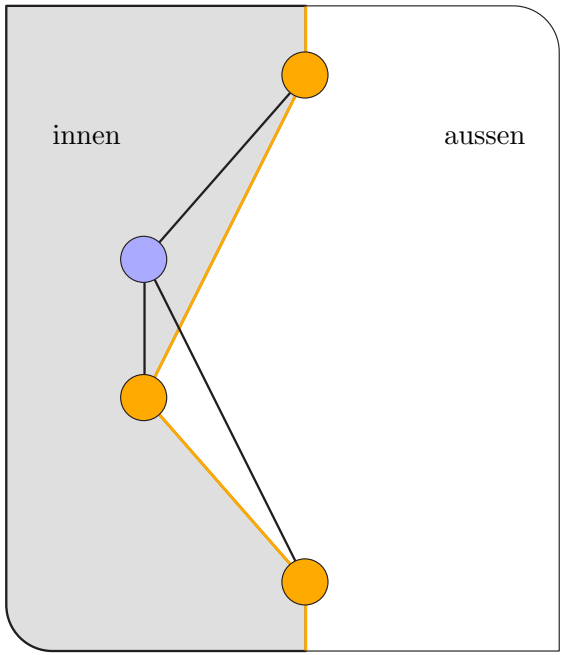
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min} $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

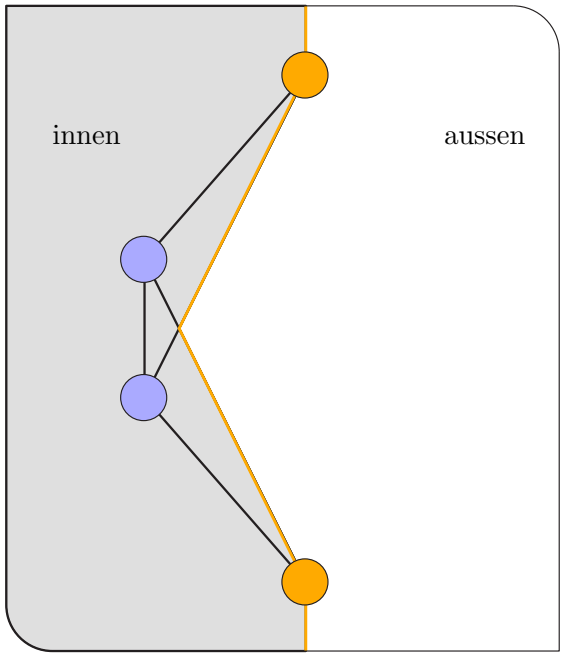
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min}
  $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

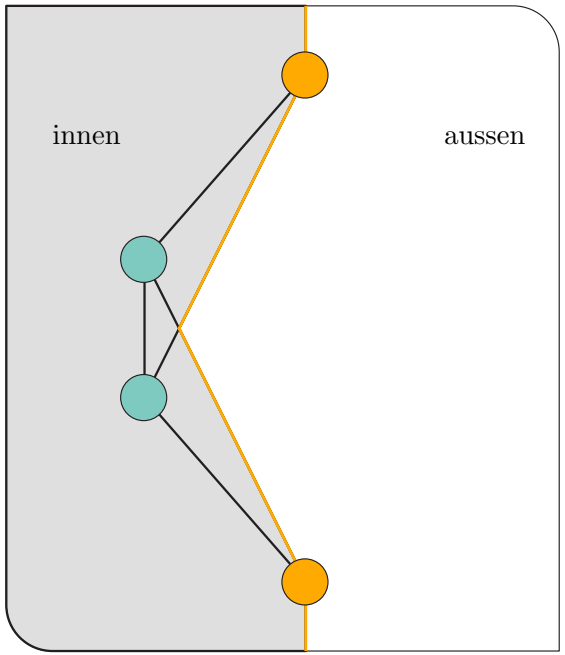
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min} $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

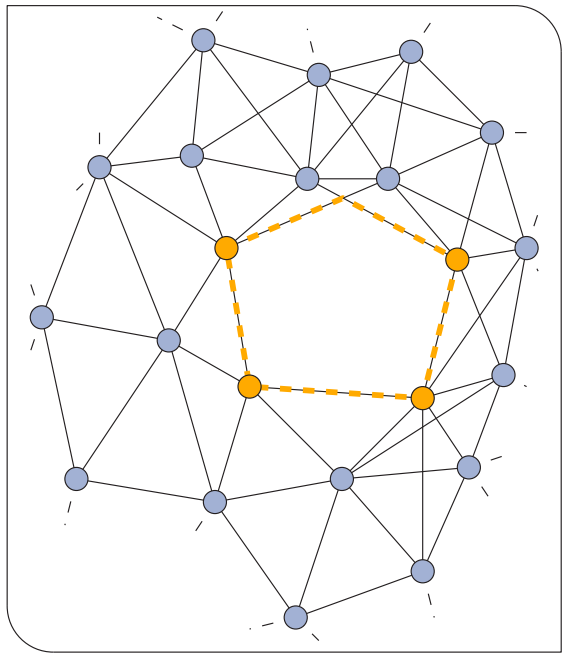
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min} $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

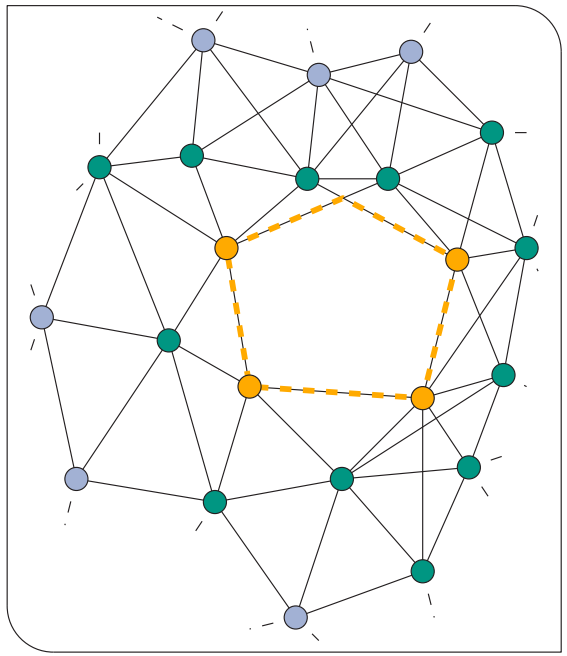
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min}
  $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

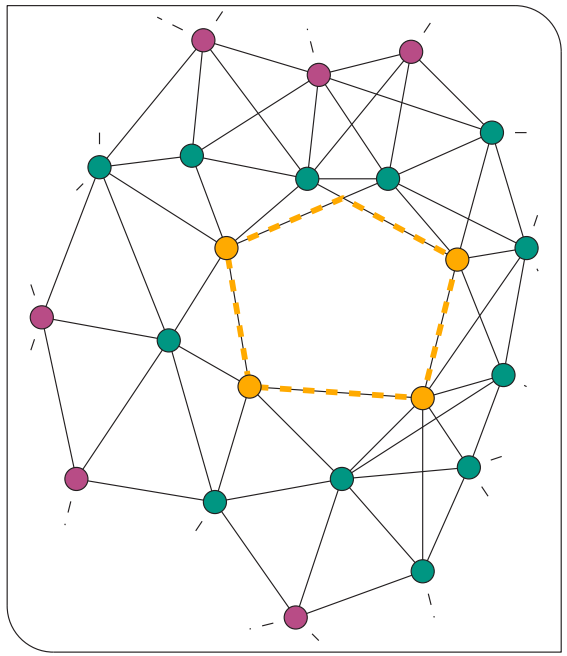
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min}
  $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

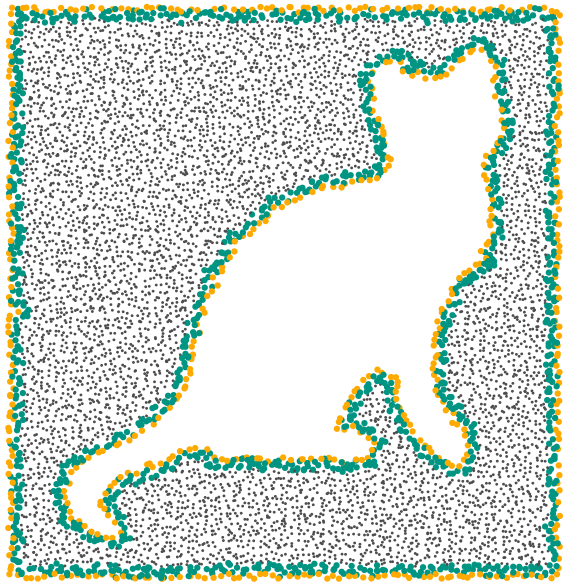
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min}
  $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

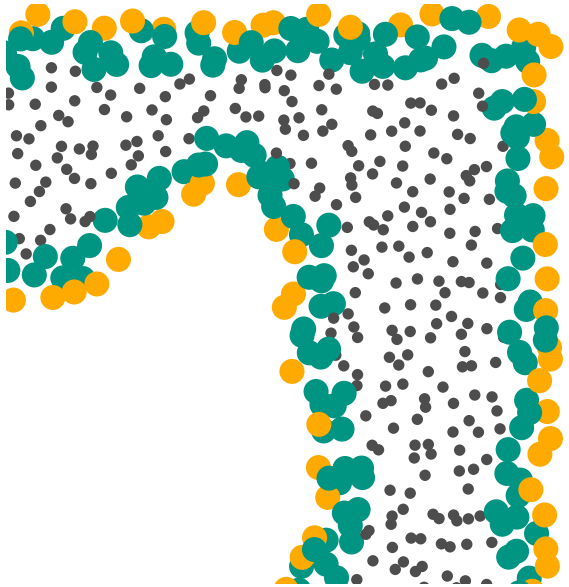
- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Loch- und Randmodell

Definitionen

Definition: Loch

- basierend auf gegebener, realer Einbettung des Verbindungsgraphen
- Facetten mit Umfang größer h_{min}
  $\Rightarrow$ Löcher



Definition: Randknoten

Unterteilung in 3 Kategorien:

- verbindliche Randknoten
- optionale Randknoten
- innere Knoten

Randerkennungsverfahren

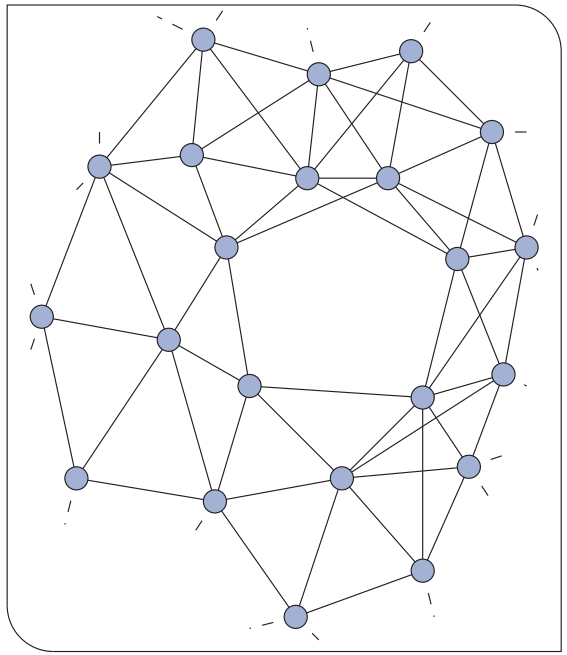
Multi-Dimensional Scaling - Boundary Recognition (MDS-BR)

Eigenschaften

- verteilter Algorithmus
- verwendet nur lokale Informationen

- benötigt nur Verbindungsinformation
  - nominell topologisches Verfahren
- im Kern aber geometrisches Verfahren
  - Koordinaten werden geschätzt

- einfache Beschreibung



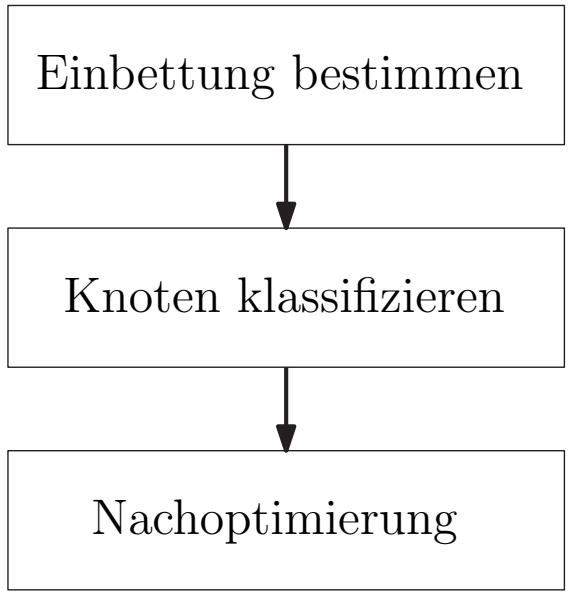
Randerkennungsverfahren

Multi-Dimensional Scaling - Boundary Recognition (MDS-BR)

Ablauf

- Schätzung von Knotenpositionen
  - Multi-Dimensional Scaling (MDS)
- Einteilung als Rand- oder innerer Knoten
  - Bedingungen an Winkel und Verbindungen
- Nachoptimierung
  - Entfernung von ”Rauschen“

(Ablauf unabhängig in jedem Knoten)



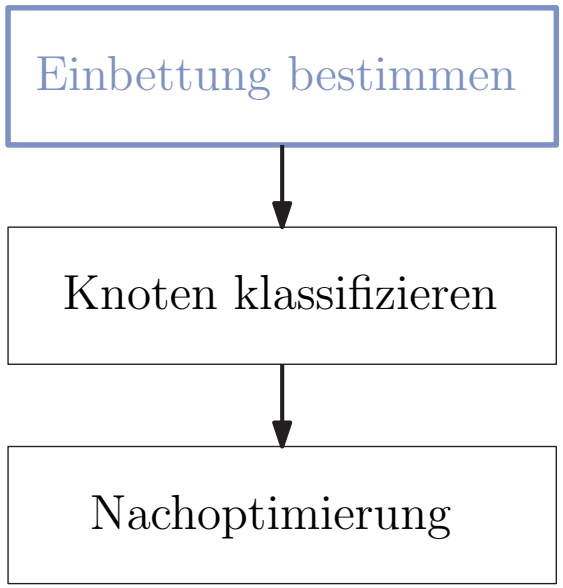
Randerkennungsverfahren

Multi-Dimensional Scaling - Boundary Recognition (MDS-BR)

Ablauf

- Schätzung von Knotenpositionen
  - Multi-Dimensional Scaling (MDS)
- Einteilung als Rand- oder innerer Knoten
  - Bedingungen an Winkel und Verbindungen
- Nachoptimierung
  - Entfernung von ”Rauschen“

(Ablauf unabhängig in jedem Knoten)



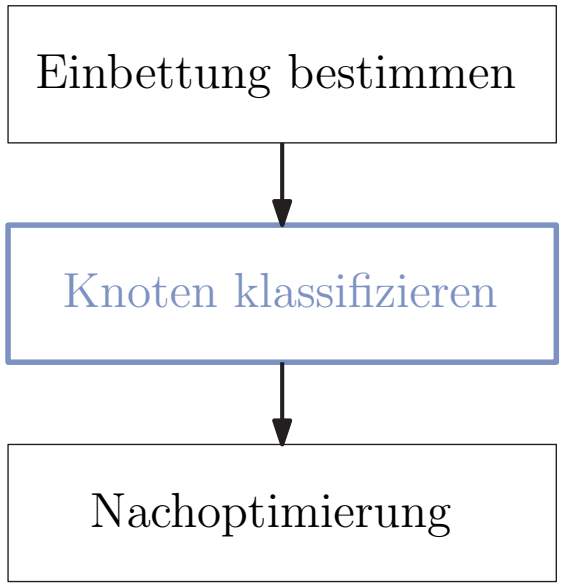
Randerkennungsverfahren

Multi-Dimensional Scaling - Boundary Recognition (MDS-BR)

Ablauf

- Schätzung von Knotenpositionen
  - Multi-Dimensional Scaling (MDS)
- Einteilung als Rand- oder innerer Knoten
  - Bedingungen an Winkel und Verbindungen
- Nachoptimierung
  - Entfernung von ”Rauschen“

(Ablauf unabhängig in jedem Knoten)



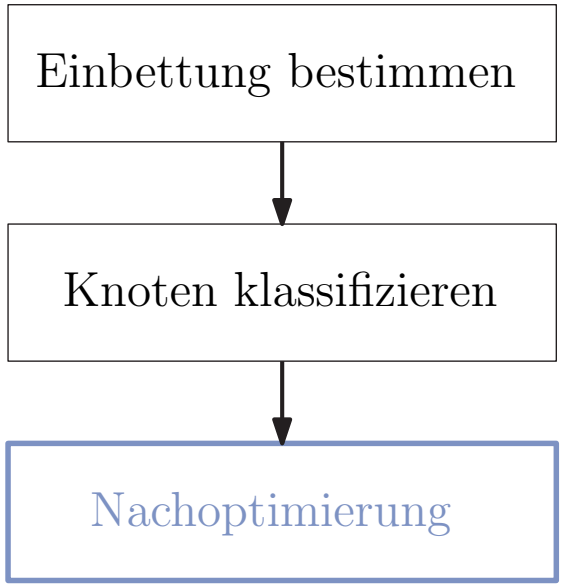
Randerkennungsverfahren

Multi-Dimensional Scaling - Boundary Recognition (MDS-BR)

Ablauf

- Schätzung von Knotenpositionen
  - Multi-Dimensional Scaling (MDS)
- Einteilung als Rand- oder innerer Knoten
  - Bedingungen an Winkel und Verbindungen
- Nachoptimierung
  - Entfernung von ”Rauschen“

(Ablauf unabhängig in jedem Knoten)



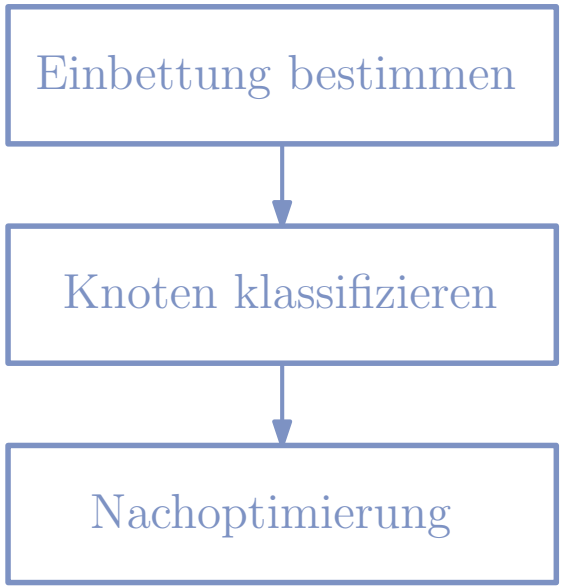
Randerkennungsverfahren

Multi-Dimensional Scaling - Boundary Recognition (MDS-BR)

Ablauf

- Schätzung von Knotenpositionen
  - Multi-Dimensional Scaling (MDS)
- Einteilung als Rand- oder innerer Knoten
  - Bedingungen an Winkel und Verbindungen
- Nachoptimierung
  - Entfernung von ”Rauschen“

(Ablauf unabhängig in jedem Knoten)



Randerkennungsverfahren

Multi-Dimensional Scaling (MDS)

gegeben

- Distanzen δ_{ij} zwischen allen Knoten i und j

gesucht

- Einbettung $\{x_1, \ldots, x_n\}$ aller Knoten in $\mathbb{R}^2$ mit $\sum_{i \neq j}(||x_i - x_j|| - \delta_{ij})^2$ minimal

Ablauf

- sei $\mathbf{X} = [x_1, \ldots, x_n]^T$, $\mathbf{B} = \mathbf{X}\mathbf{X}^T = (b_{ij})$ mit
  $b_{ij} = -\frac{1}{2}\left(\delta_{ij}^2 - \frac{1}{n}\sum \delta_{rj}^2 - \frac{1}{n}\sum \delta_{is}^2 + \frac{1}{n^2}\sum\sum \delta_{rs}^2\right)$ (*double-centering*)
- $\mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ diagonalisierbar
- Verwende größte 2 Eigenwerte, um $\mathbf{X} = \mathbf{V}\mathbf{D}^{1/2}$ zu nähern

Randerkennungsverfahren

Multi-Dimensional Scaling (MDS)

gegeben

- Distanzen δ_{ij} zwischen allen Knoten i und j

gesucht

- Einbettung $\{x_1, \ldots, x_n\}$ aller Knoten in $\mathbb{R}^2$ mit $\sum_{i \neq j} (||x_i - x_j|| - \delta_{ij})^2$ minimal

Ablauf

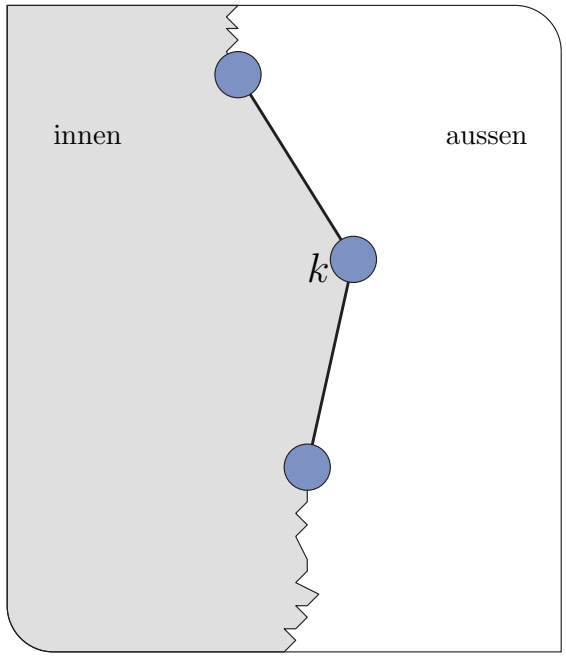
- sei $\mathbf{X} = [x_1, \ldots, x_n]^T$, $\mathbf{B} = \mathbf{X}\mathbf{X}^T = (b_{ij})$ mit
  $b_{ij} = -\frac{1}{2}\left(\delta_{ij}^2 - \frac{1}{n}\sum \delta_{rj}^2 - \frac{1}{n}\sum \delta_{is}^2 + \frac{1}{n^2}\sum\sum \delta_{rs}^2\right)$ (*double-centering*)
- $\mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ diagonalisierbar
- Verwende größte 2 Eigenwerte, um $\mathbf{X} = \mathbf{V}\mathbf{D}^{1/2}$ zu nähern

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

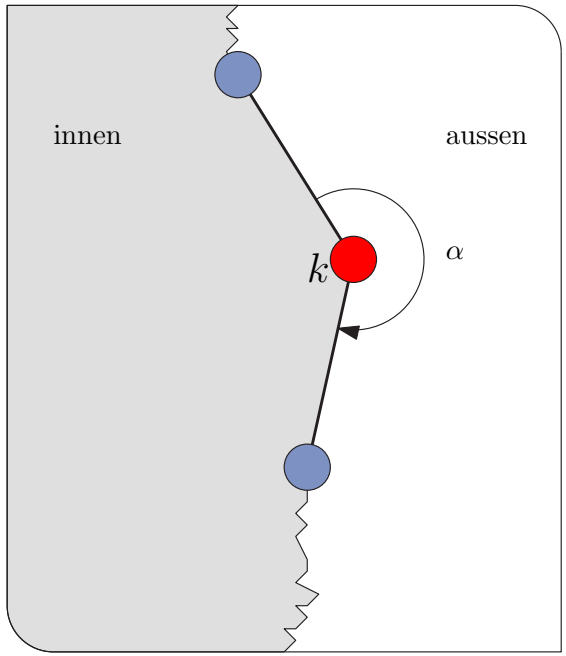
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

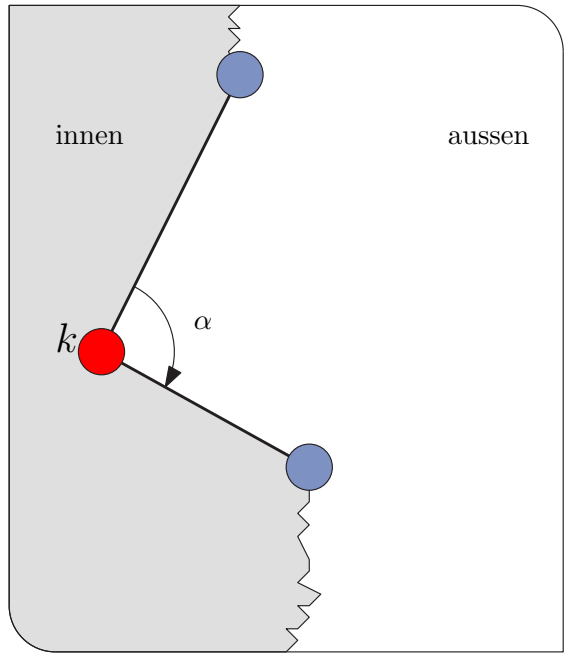
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

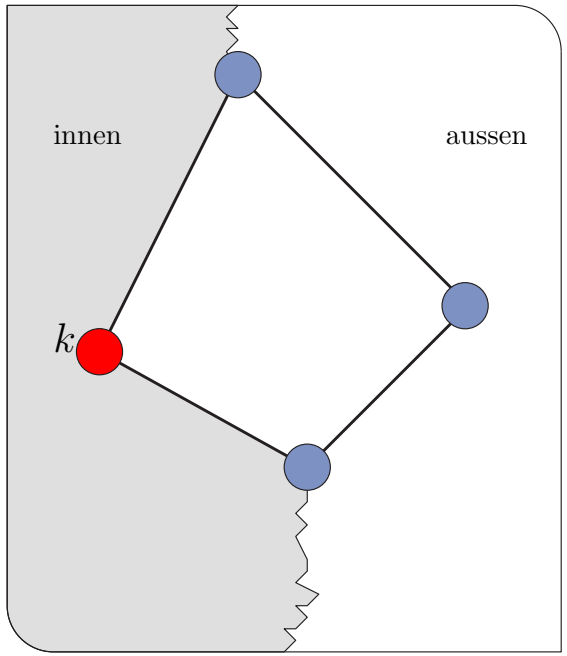
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

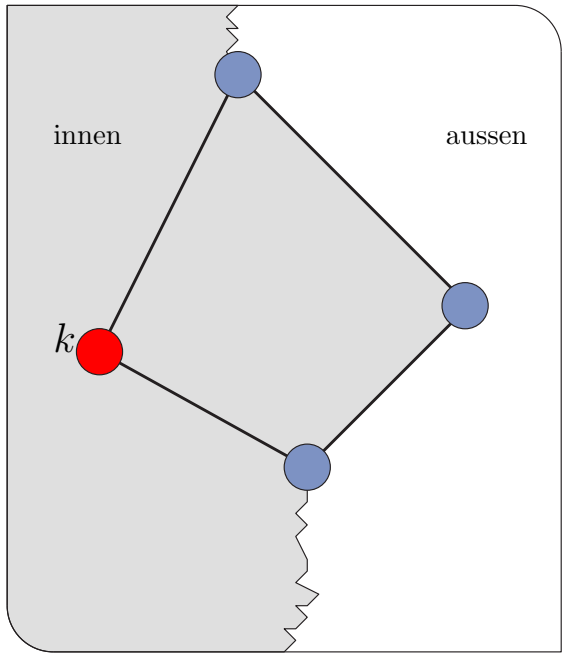
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

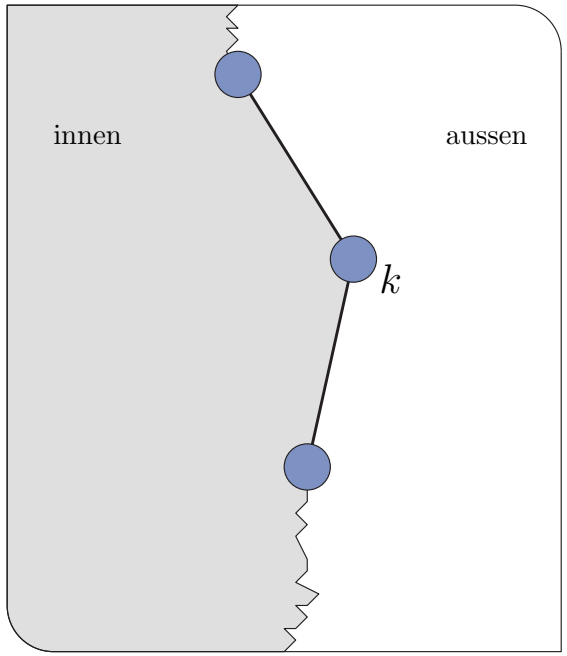
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

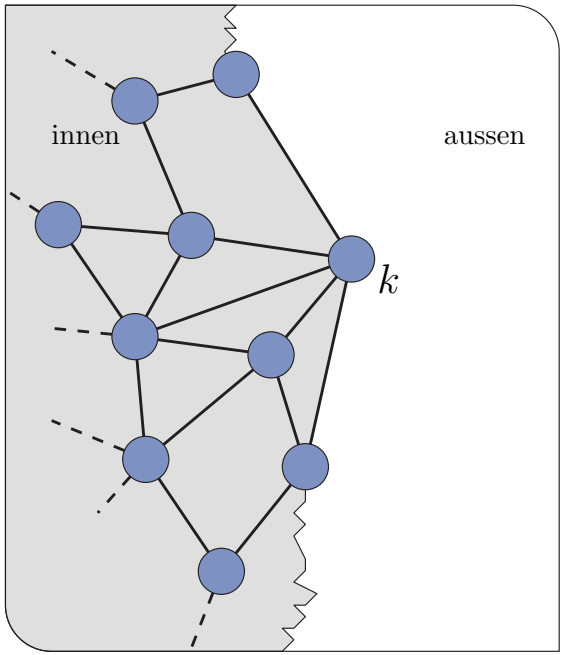
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

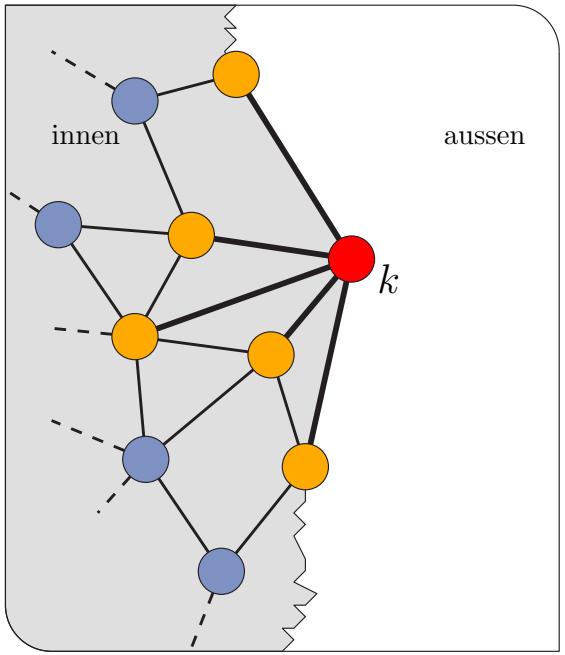
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

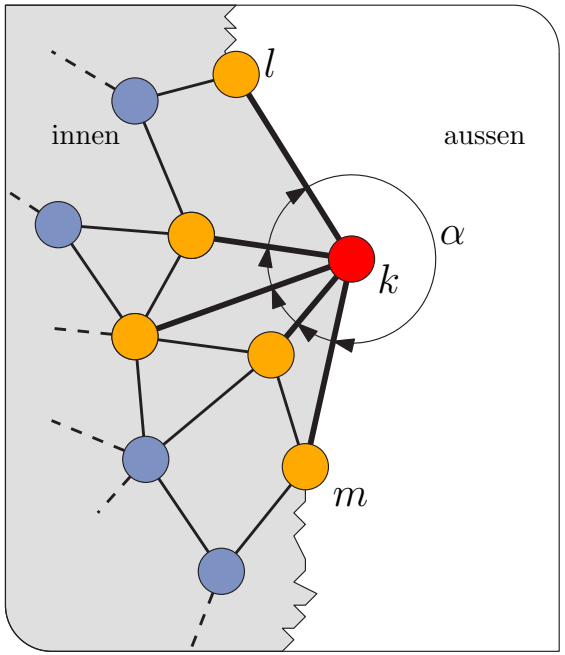
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

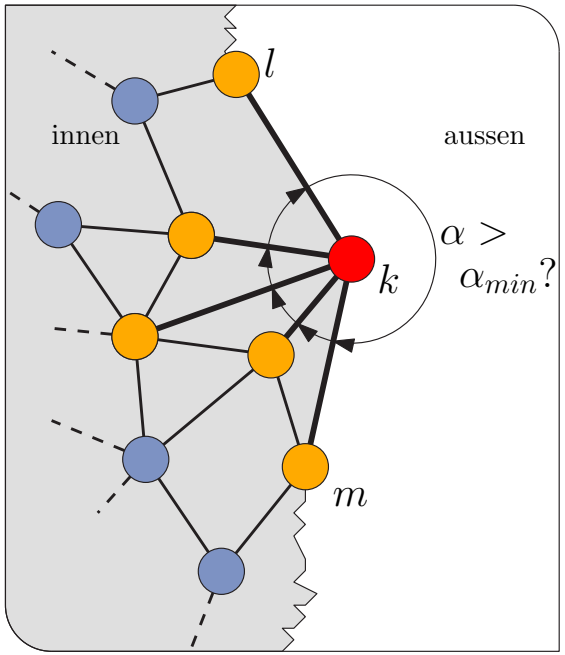
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

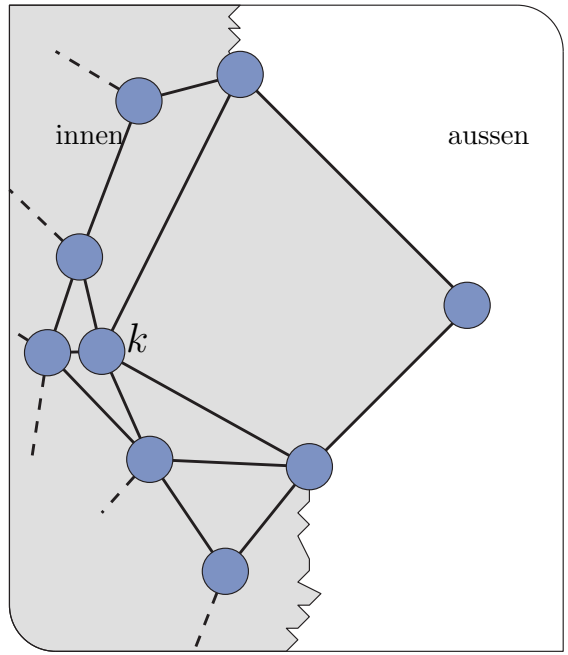
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

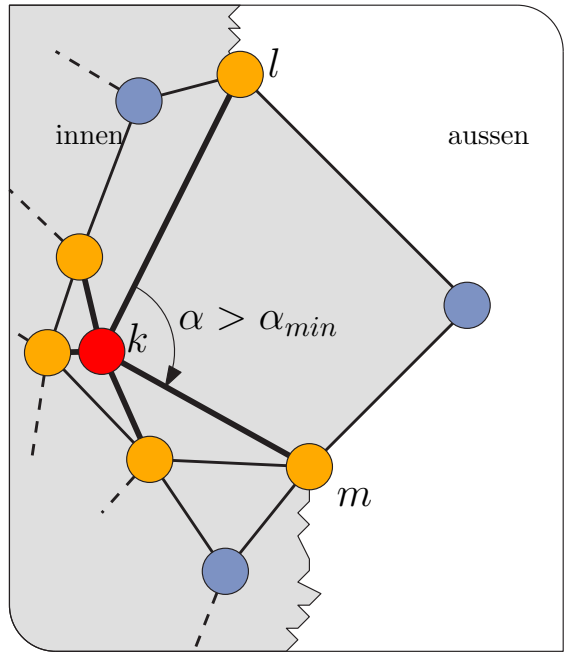
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

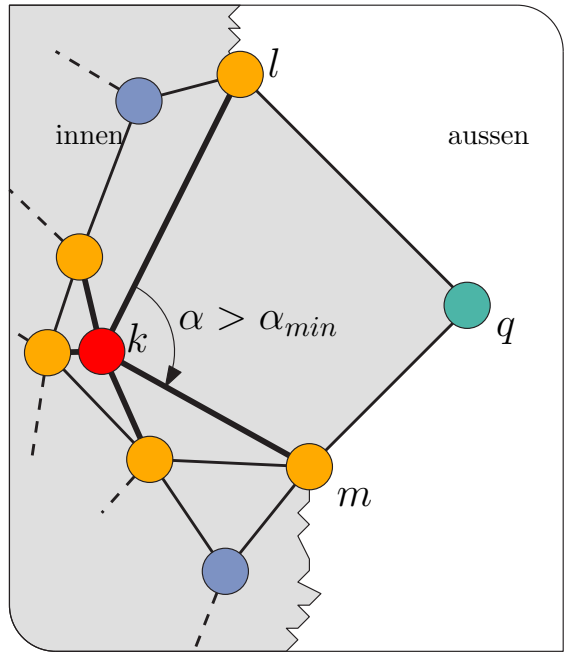
Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern

Ablauf

- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

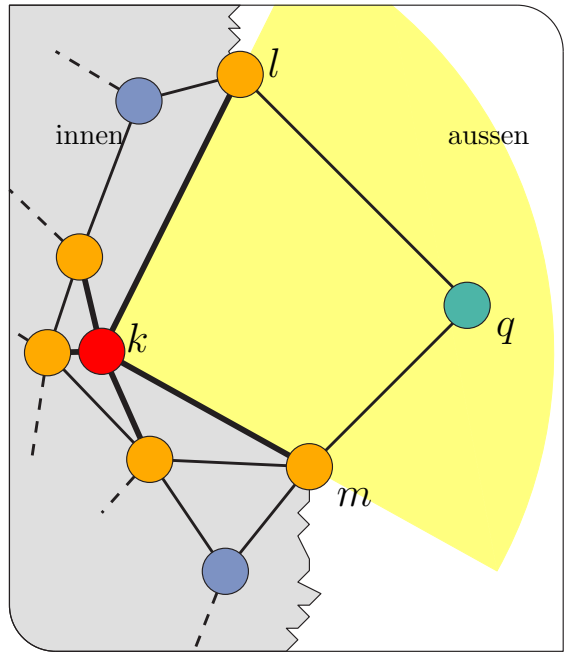


Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

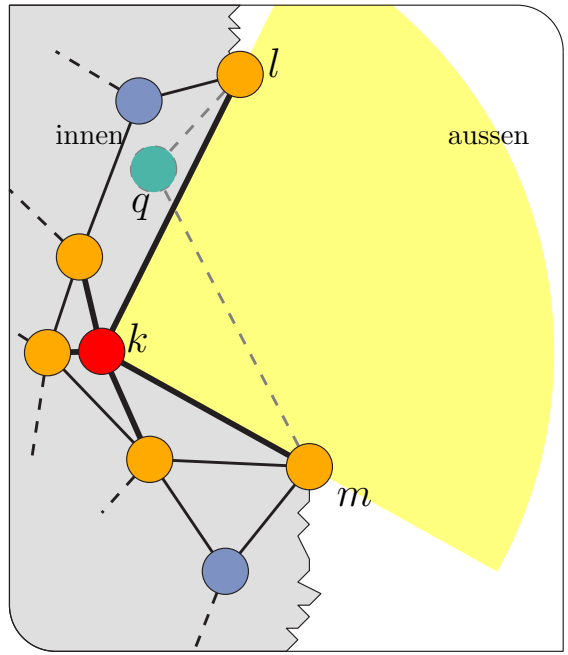
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Knotenklassifikation

Beobachtung

- typischerweise große Winkel am Rand
- Vermeidung von Mikrolöchern



Ablauf

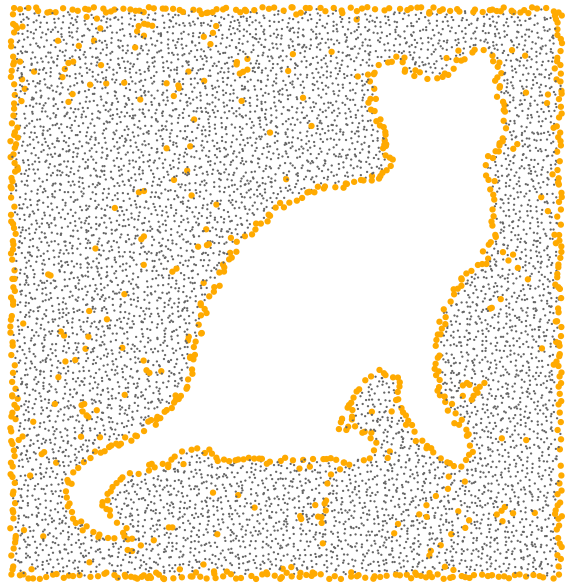
- betrachte 1-Hop Nachbarschaft von k
- bestimme größten Winkel $\alpha = \sphericalangle(l, k, m)$
  - falls $\alpha > \alpha_{min}$: k Kandidat für Randknoten
- betrachte Nachbarn q von l und m
  - falls sich q im Winkelbereich $\sphericalangle(l, k, m)$ befindet: k kein Randknoten

Randerkennungsverfahren

Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“



Ablauf

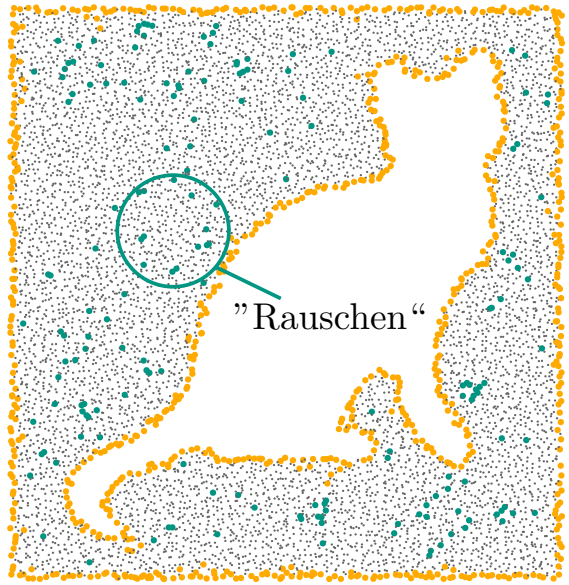
- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten

Randerkennungsverfahren

Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“



Ablauf

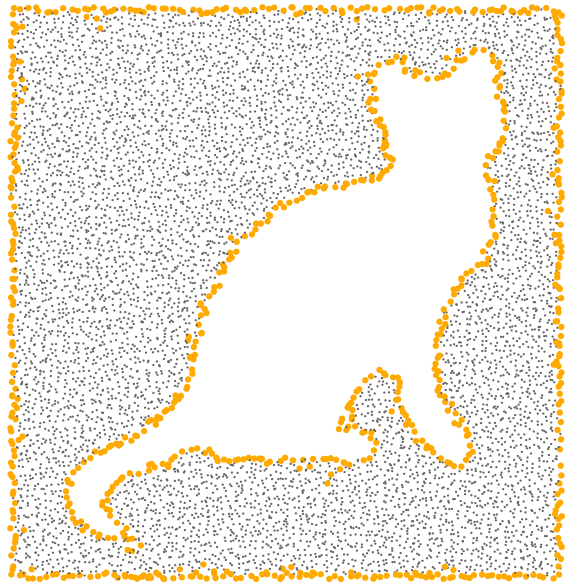
- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten

Randerkennungsverfahren

Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“



Ablauf

- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten

Randerkennungsverfahren

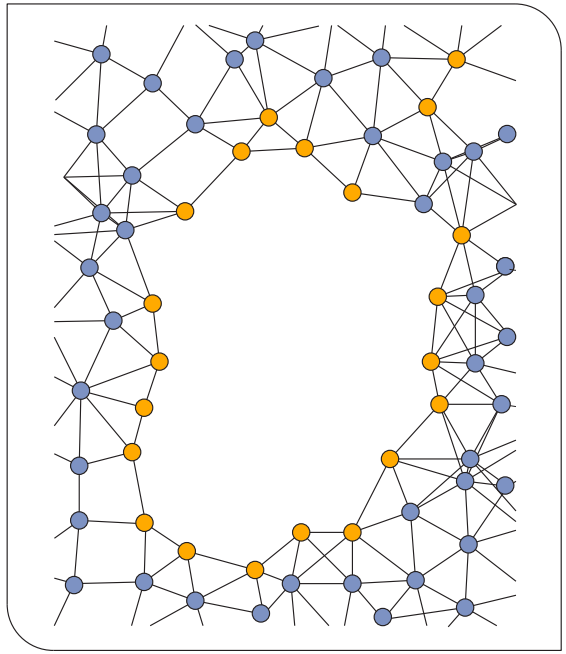
Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“

Ablauf

- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten



Randerkennungsverfahren

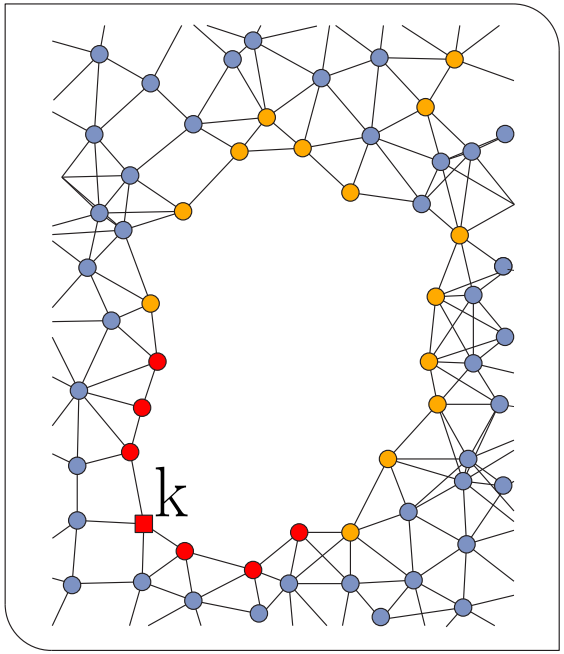
Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“

Ablauf

- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten



Randerkennungsverfahren

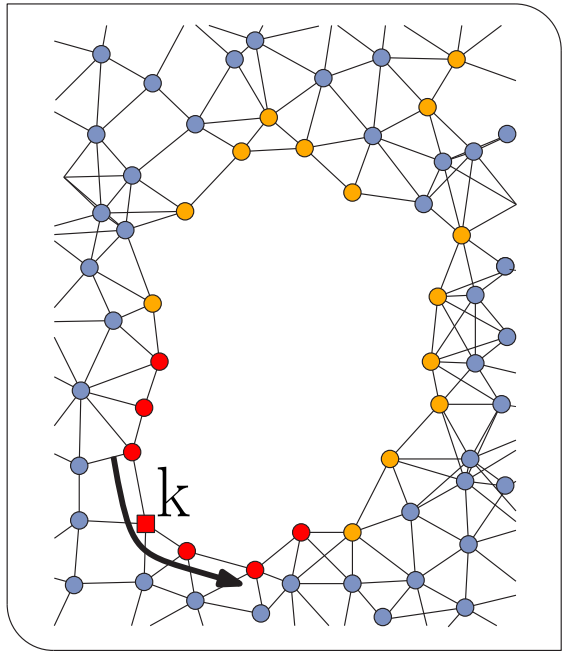
Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“

Ablauf

- Jeder Randknoten *k* betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der *k* enthält, bleibt *k* Randknoten



Randerkennungsverfahren

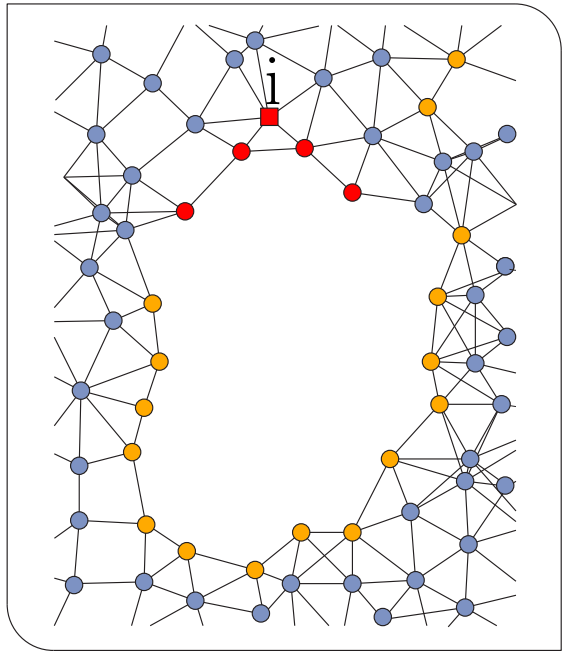
Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“

Ablauf

- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten



Randerkennungsverfahren

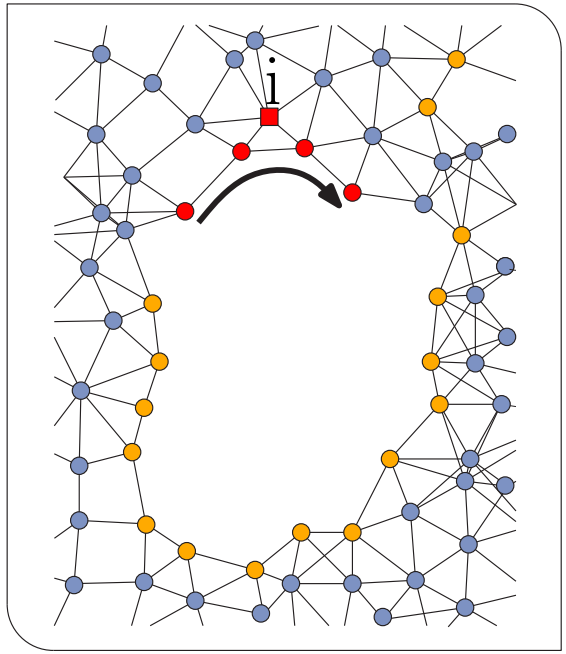
Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“

Ablauf

- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten



Randerkennungsverfahren

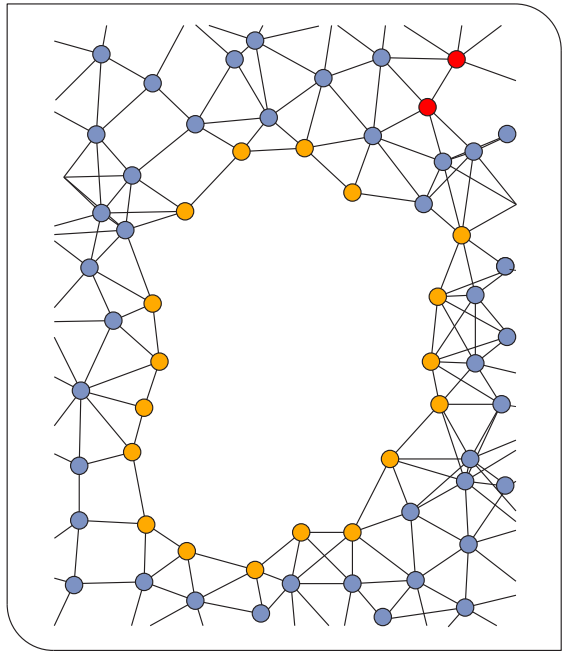
Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“

Ablauf

- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten



Randerkennungsverfahren

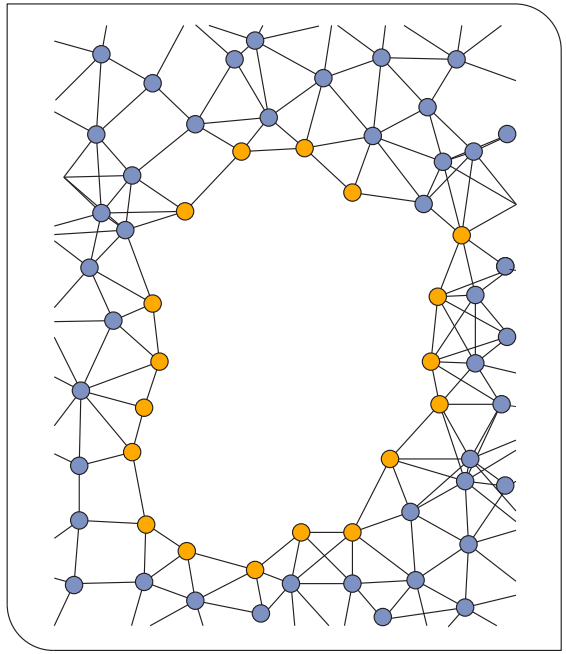
Nachoptimierung

Ziel

- Identifikation von Randstrukturen mit kleiner Ausdehnung
- Entfernung dieses ”Rauschens“

Ablauf

- Jeder Randknoten k betrachtet seine r_{min}-Hop Nachbarschaft aus Randknoten
- existiert ein kürzester Weg mit Länge r_{min} der k enthält, bleibt k Randknoten

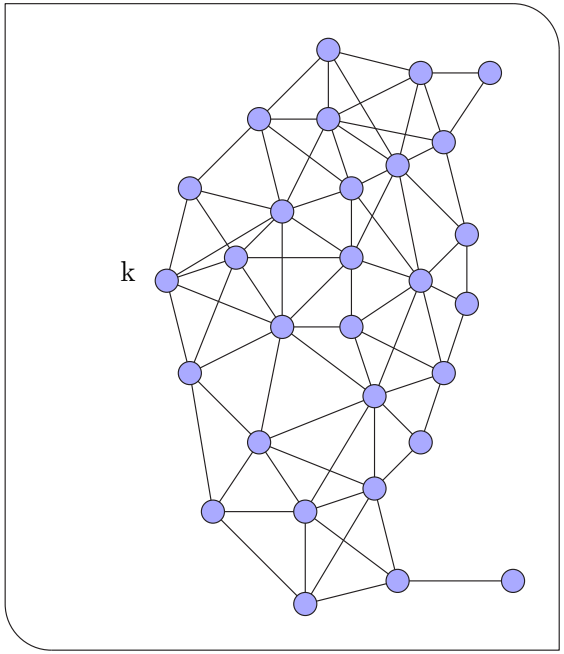


Randerkennungsverfahren

Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten *k*

- Knotenklassifikation
- Nachoptimierung



Randerkennungsverfahren

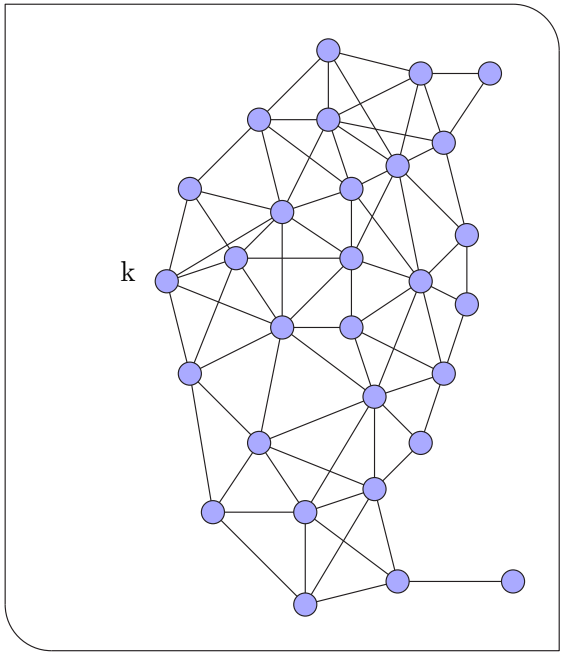
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Knotenklassifikation

- extrahiere 2-Hop Nachbarschaft von k
- bestimme Einbettung mit MDS
  (verwende Hop-Anzahl als Distanzmaß)
- teste auf Randknoten in Einbettung
  (1-Hop Nachbarschaft, $\alpha_{min} = 0.5\pi$)



Randerkennungsverfahren

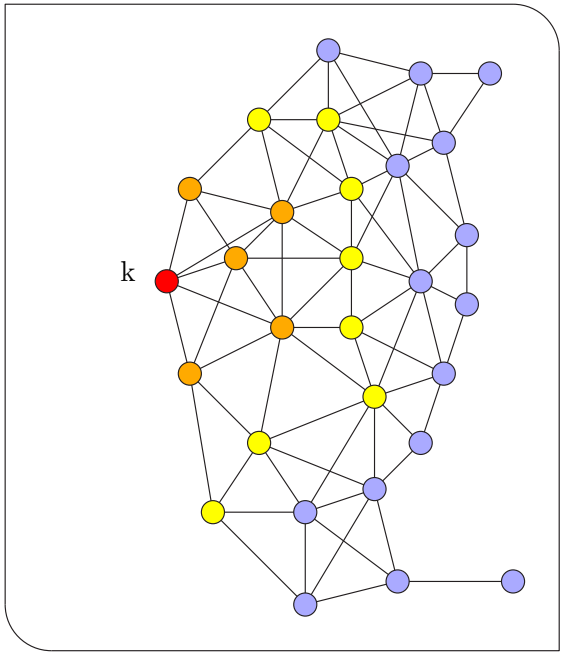
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Knotenklassifikation

- extrahiere 2-Hop Nachbarschaft von k
- bestimme Einbettung mit MDS
  (verwende Hop-Anzahl als Distanzmaß)
- teste auf Randknoten in Einbettung
  (1-Hop Nachbarschaft, $\alpha_{min} = 0.5\pi$)



Randerkennungsverfahren

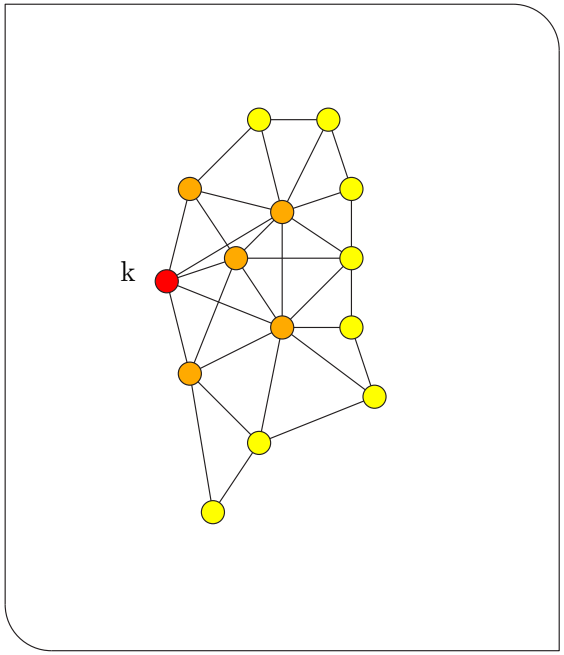
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Knotenklassifikation

- extrahiere 2-Hop Nachbarschaft von k
- bestimme Einbettung mit MDS
  (verwende Hop-Anzahl als Distanzmaß)
- teste auf Randknoten in Einbettung
  (1-Hop Nachbarschaft, $\alpha_{min} = 0.5\pi$)



Randerkennungsverfahren

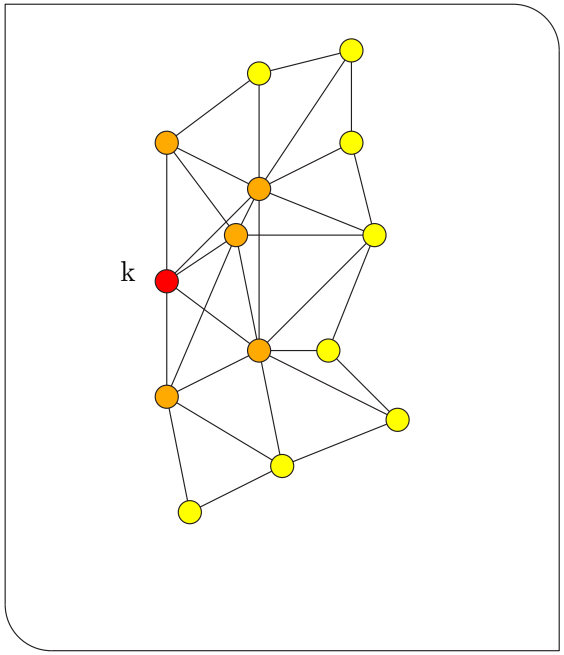
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Knotenklassifikation

- extrahiere 2-Hop Nachbarschaft von k
- bestimme Einbettung mit MDS (verwende Hop-Anzahl als Distanzmaß)
- teste auf Randknoten in Einbettung (1-Hop Nachbarschaft, $\alpha_{min} = 0.5\pi$)



Randerkennungsverfahren

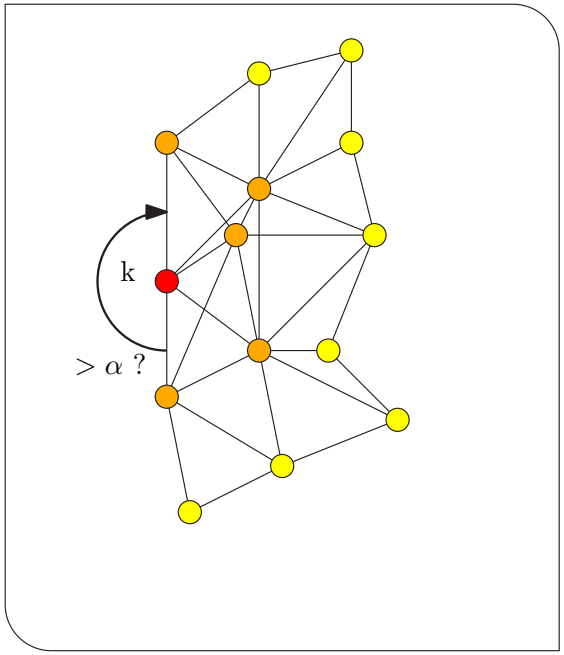
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Knotenklassifikation

- extrahiere 2-Hop Nachbarschaft von k
- bestimme Einbettung mit MDS
  (verwende Hop-Anzahl als Distanzmaß)
- teste auf Randknoten in Einbettung
  (1-Hop Nachbarschaft, $\alpha_{min} = 0.5\pi$)



Randerkennungsverfahren

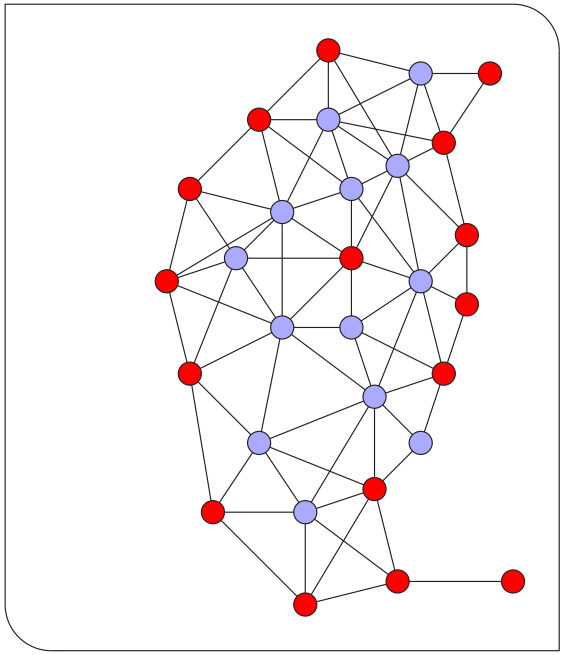
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Knotenklassifikation

- extrahiere 2-Hop Nachbarschaft von k
- bestimme Einbettung mit MDS
  (verwende Hop-Anzahl als Distanzmaß)
- teste auf Randknoten in Einbettung
  (1-Hop Nachbarschaft, $\alpha_{min} = 0.5\pi$)



Randerkennungsverfahren

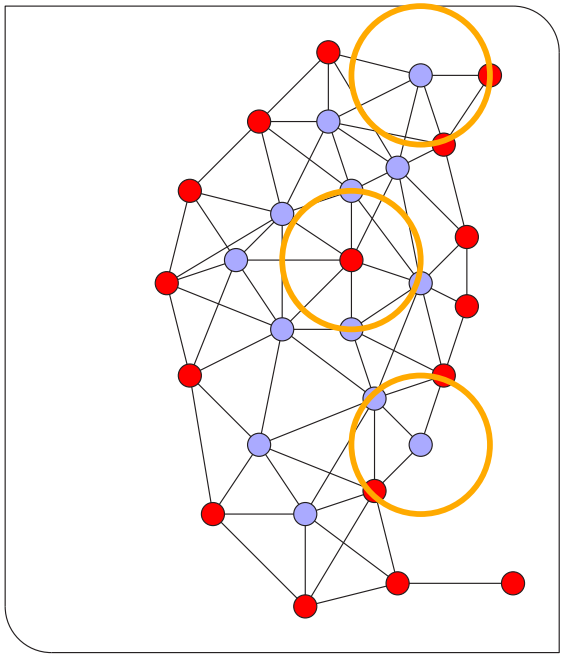
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Nachoptimierung

- suche nach Randstrukturen mit Ausdehnung kleiner als $r_{min} = 5$
- Manko: fehlende Randknoten können nicht ergänzt werden



Randerkennungsverfahren

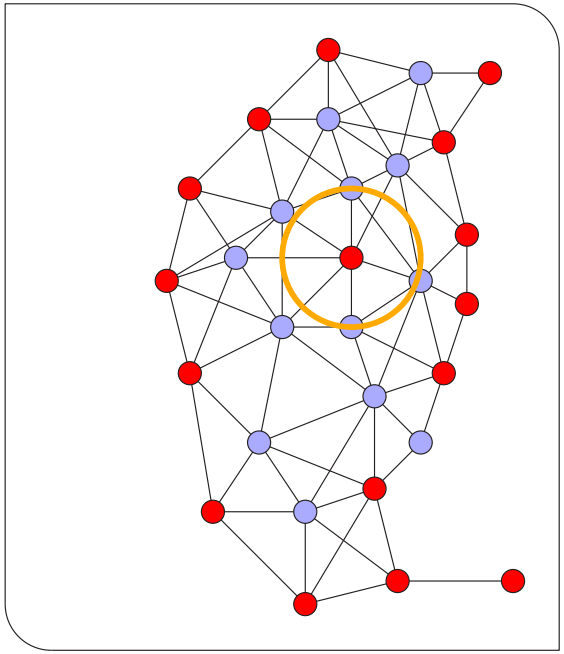
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Nachoptimierung

- suche nach Randstrukturen mit Ausdehnung kleiner als $r_{min} = 5$
- Manko: fehlende Randknoten können nicht ergänzt werden



Randerkennungsverfahren

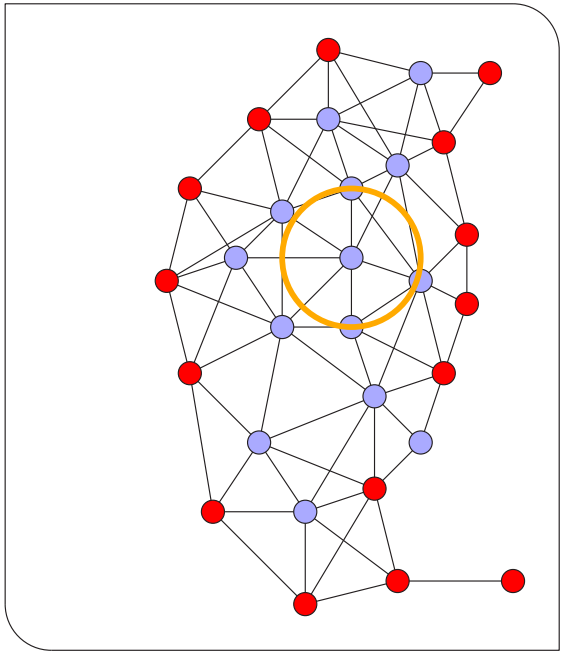
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Nachoptimierung

- suche nach Randstrukturen mit Ausdehnung kleiner als $r_{min} = 5$
- Manko: fehlende Randknoten können nicht ergänzt werden



Randerkennungsverfahren

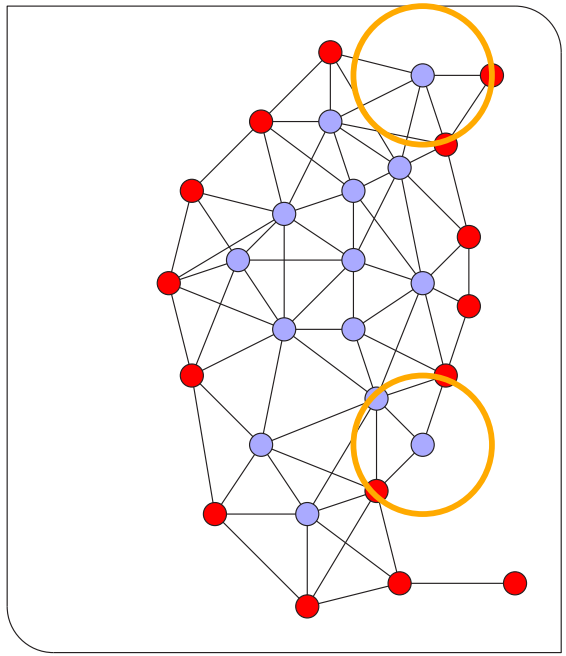
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Nachoptimierung

- suche nach Randstrukturen mit Ausdehnung kleiner als $r_{min} = 5$
- Manko: fehlende Randknoten können nicht ergänzt werden



Randerkennungsverfahren

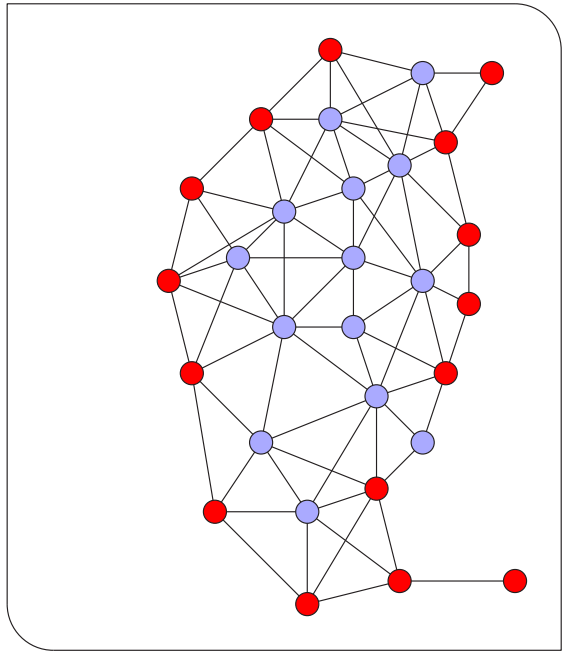
Exemplarischer Ablauf von MDS-BR

Ablauf in jedem Knoten k

- Knotenklassifikation
- Nachoptimierung

Nachoptimierung

- suche nach Randstrukturen mit Ausdehnung kleiner als $r_{min} = 5$
- Manko: fehlende Randknoten können nicht ergänzt werden



Randerkennungsverfahren

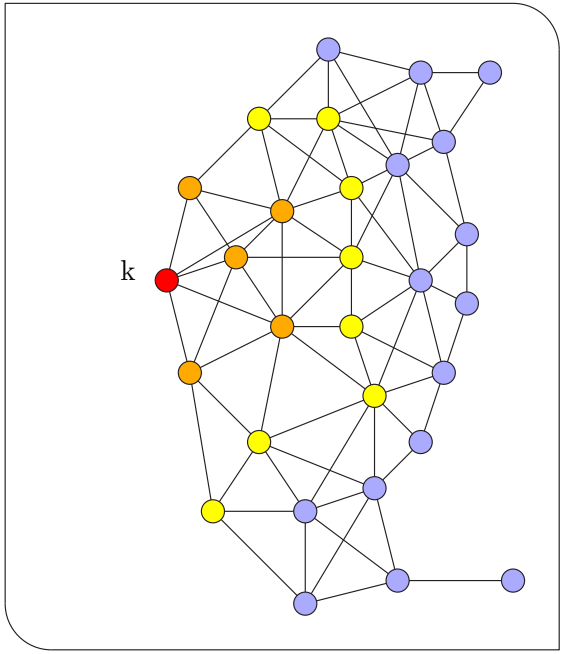
Aufwandsbetrachtung von MDS-BR

Knotenklassifikation

- 2 Nachrichten pro Knoten
- Rechenaufwand kubisch in Größe der 2-Hop Nachbarschaft (dominiert von MDS)

Nachoptimierung

- r_{min} Nachrichten pro Randknoten
- Rechenaufwand kubisch in Größe der r_{min}-Hop Nachbarschaft aus Randknoten



Randerkennungsverfahren

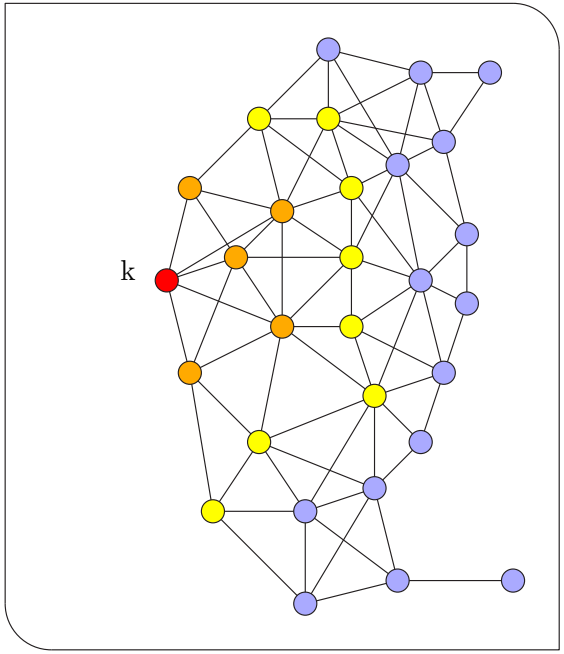
Aufwandsbetrachtung von MDS-BR

Knotenklassifikation

- 2 Nachrichten pro Knoten
- Rechenaufwand kubisch in Größe der 2-Hop Nachbarschaft (dominiert von MDS)

Nachoptimierung

- r_{min} Nachrichten pro Randknoten
- Rechenaufwand kubisch in Größe der r_{min}-Hop Nachbarschaft aus Randknoten



Randerkennungsverfahren

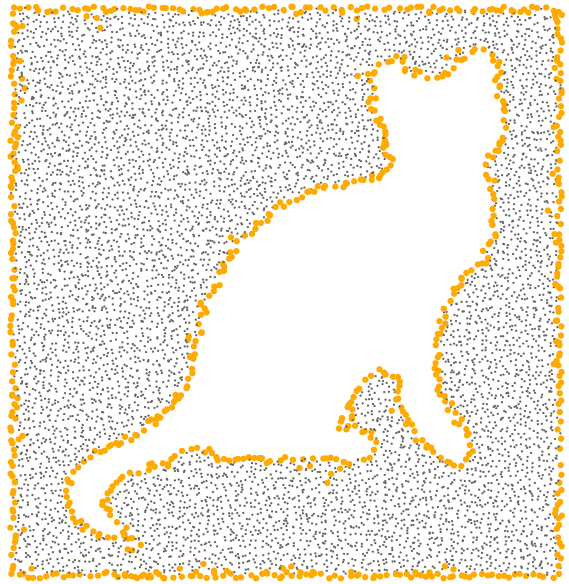
Aufwandsbetrachtung von MDS-BR

Knotenklassifikation

- 2 Nachrichten pro Knoten
- Rechenaufwand kubisch in Größe der 2-Hop Nachbarschaft (dominiert von MDS)

Nachoptimierung

- r_{min} Nachrichten pro Randknoten
- Rechenaufwand kubisch in Größe der r_{min}-Hop Nachbarschaft aus Randknoten



Randerkennungsverfahren

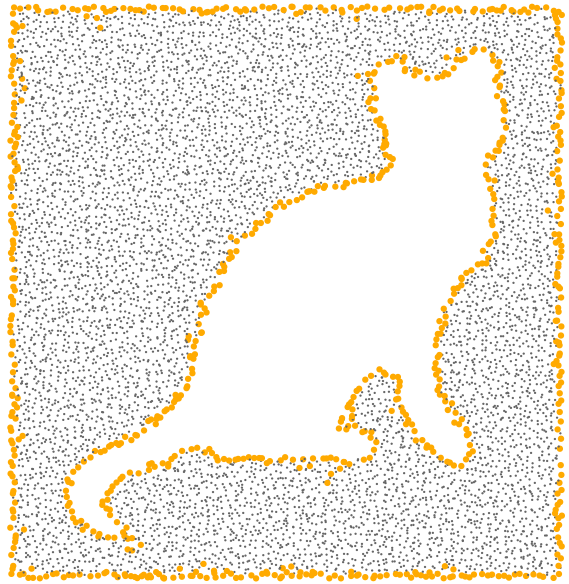
Aufwandsbetrachtung von MDS-BR

Knotenklassifikation

- 2 Nachrichten pro Knoten
- Rechenaufwand kubisch in Größe der 2-Hop Nachbarschaft (dominiert von MDS)

Nachoptimierung

- r_{min} Nachrichten pro Randknoten
- Rechenaufwand kubisch in Größe der r_{min}-Hop Nachbarschaft aus Randknoten



Simulation

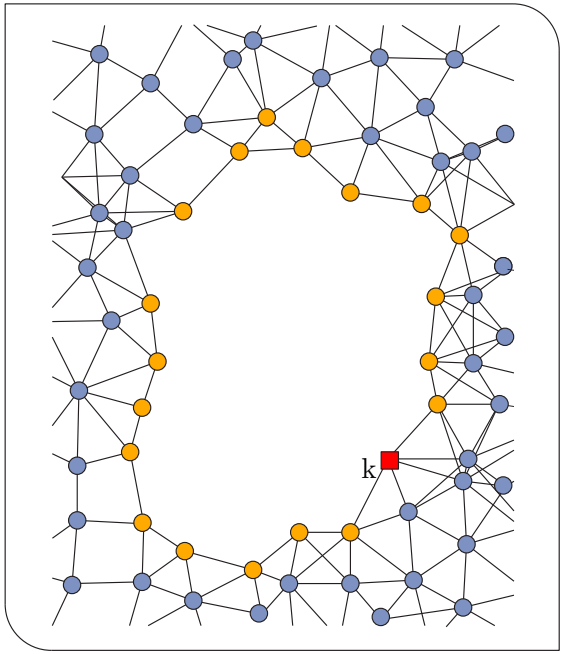
untersuchte Aspekte

freie Parameter des MDS-BR Verfahrens

- Öffnungswinkel α_{min}
- Ausdehnung r_{min}

Varianten des Verfahrens

- Einbettungsverfahren
  (OPT, MDS, MDS3, SSMDS)
- Winkelbedingungen
  (direkte Nachbarn, zweite Nachbarn)



Simulation

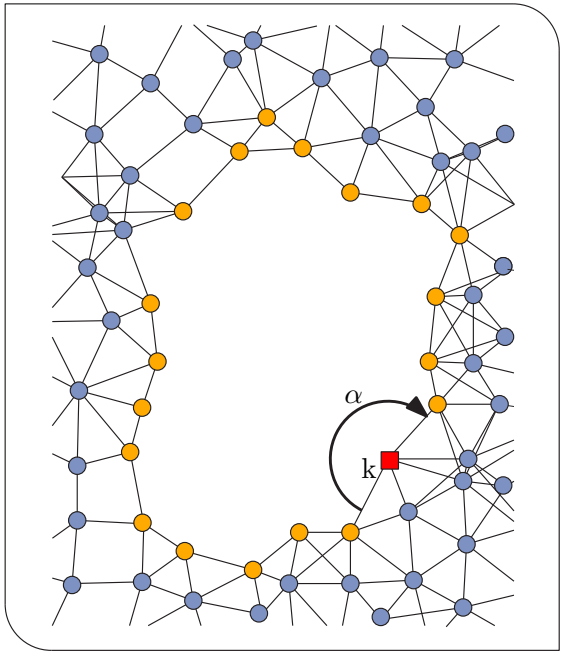
untersuchte Aspekte

freie Parameter des MDS-BR Verfahrens

- Öffnungswinkel α_{min}
- Ausdehnung r_{min}

Varianten des Verfahrens

- Einbettungsverfahren
  (OPT, MDS, MDS3, SSMDS)
- Winkelbedingungen
  (direkte Nachbarn, zweite Nachbarn)



Simulation

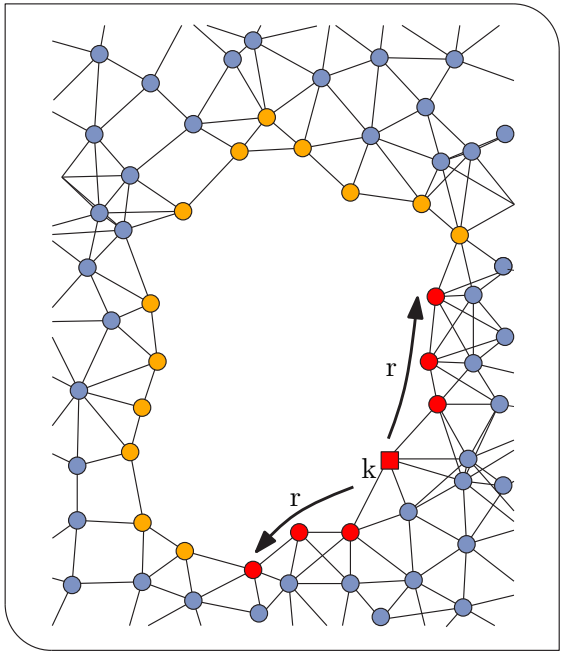
untersuchte Aspekte

freie Parameter des MDS-BR Verfahrens

- Öffnungswinkel α_{min}
- Ausdehnung r_{min}

Varianten des Verfahrens

- Einbettungsverfahren
  (OPT, MDS, MDS3, SSMDS)
- Winkelbedingungen
  (direkte Nachbarn, zweite Nachbarn)



Simulation

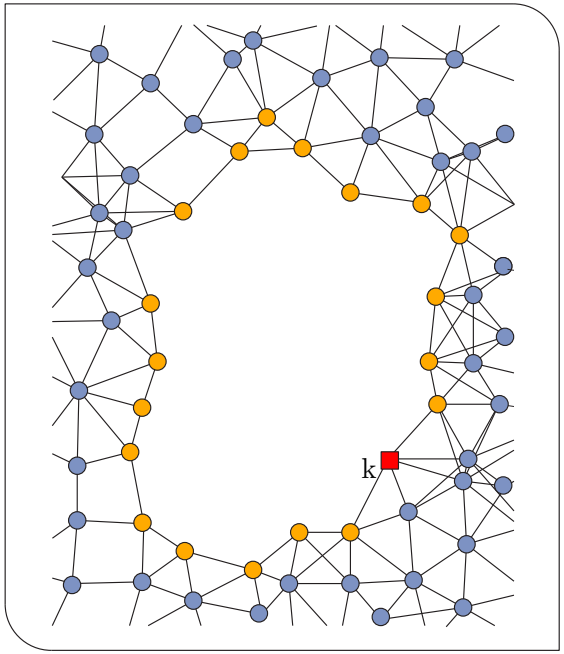
untersuchte Aspekte

freie Parameter des MDS-BR Verfahrens

- Öffnungswinkel $\alpha_{min} = 0.5\pi$
- Ausdehnung $r_{min} = 5$

Varianten des Verfahrens

- Einbettungsverfahren
  (OPT, MDS, MDS3, SSMDS)
- Winkelbedingungen
  (direkte Nachbarn, zweite Nachbarn)



Simulation

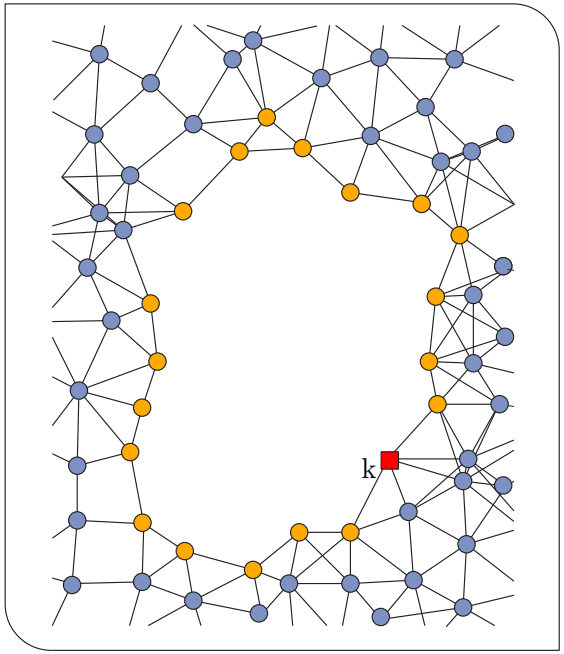
untersuchte Aspekte

freie Parameter des MDS-BR Verfahrens

- Öffnungswinkel $\alpha_{min} = 0.5\pi$
- Ausdehnung $r_{min} = 5$

Varianten des Verfahrens

- Einbettungsverfahren
  (OPT, MDS, MDS3, SSMDS)
- Winkelbedingungen
  (direkte Nachbarn, zweite Nachbarn)



Simulation

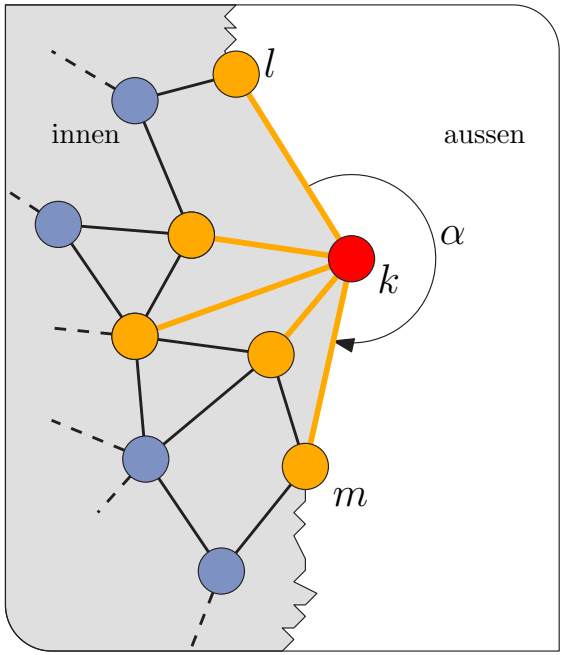
untersuchte Aspekte

freie Parameter des MDS-BR Verfahrens

- Öffnungswinkel $\alpha_{min} = 0.5\pi$
- Ausdehnung $r_{min} = 5$

Varianten des Verfahrens

- Einbettungsverfahren
  (OPT, MDS, MDS3, SSMDS)
- Winkelbedingungen
  (direkte Nachbarn, zweite Nachbarn)



Simulation

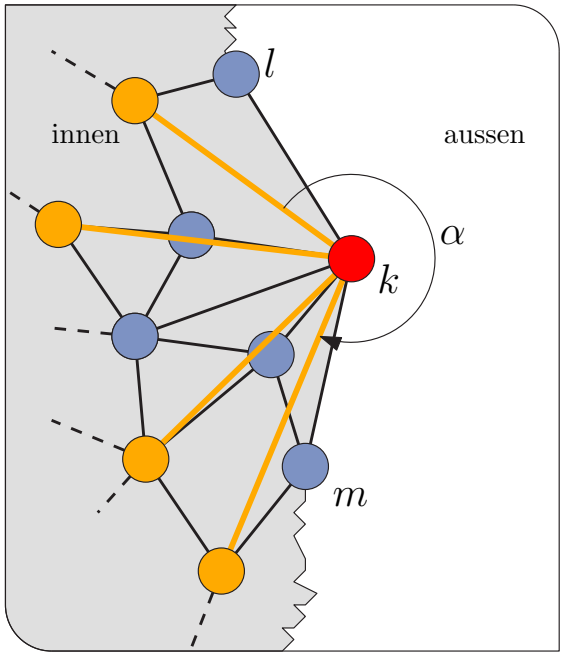
untersuchte Aspekte

freie Parameter des MDS-BR Verfahrens

- Öffnungswinkel $\alpha_{min} = 0.5\pi$
- Ausdehnung $r_{min} = 5$

Varianten des Verfahrens

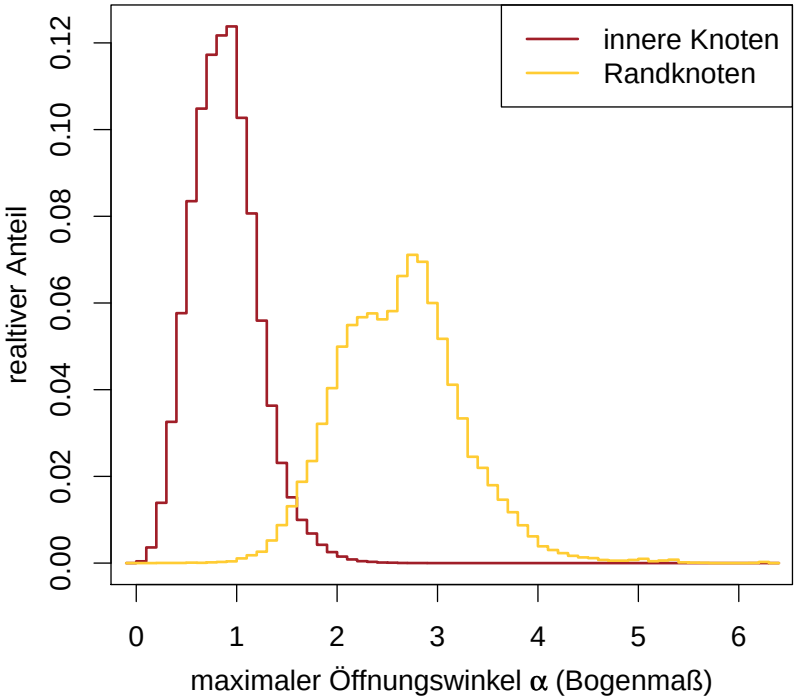
- Einbettungsverfahren
  (OPT, MDS, MDS3, SSMDS)
- Winkelbedingungen
  (direkte Nachbarn, zweite Nachbarn)



Simulation

Winkelverteilungen

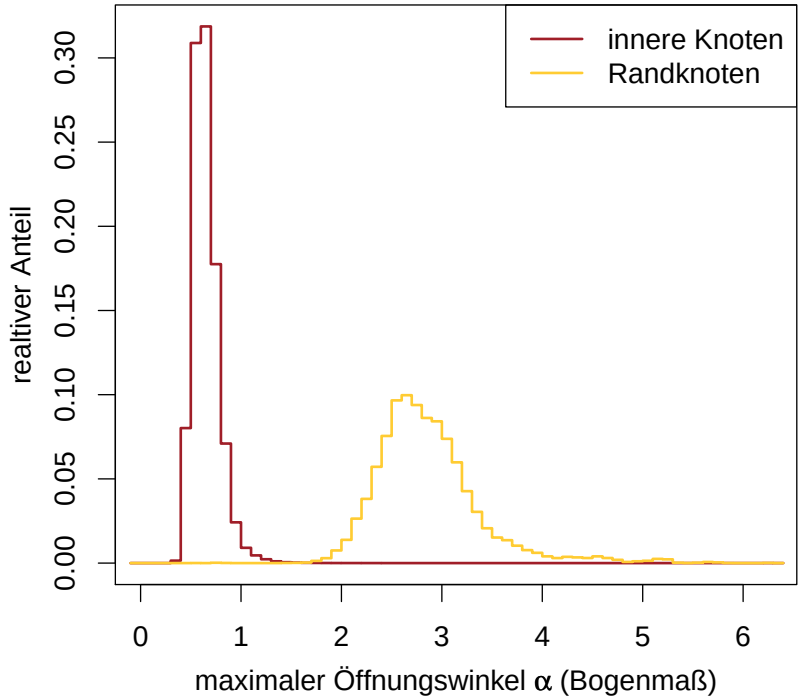
1-hop Nachbarschaft



Fehlklassifikationen:

- Randknoten: 2.88%
- innere Knoten: 3.07%

exkl. 2-hop Nachbarschaft



Fehlklassifikationen:

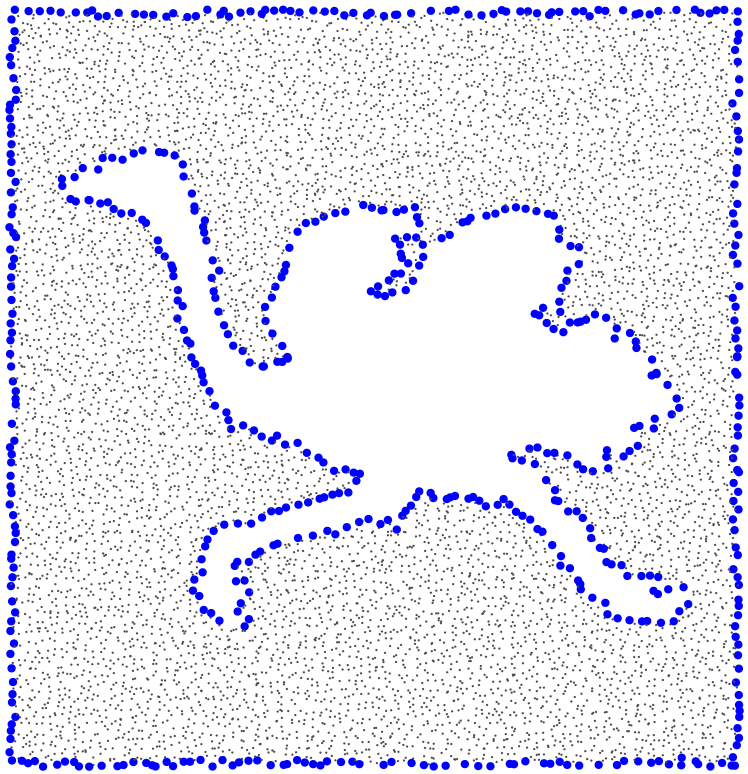
- Randknoten: 0.04%
- innere Knoten: 0.04%

Simulation

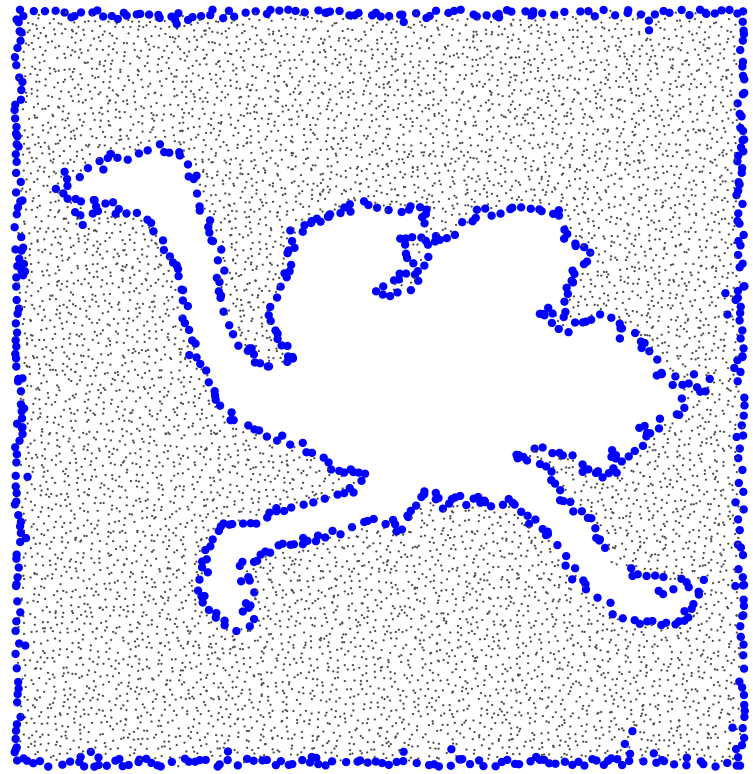
Ergebnisse

Zusammenfassung

- ausführliche Ergebnisse siehe Vortrag Markus Völker



Truth



MDS-BR

Karlsruher Institut für Technologie

- II -

Routing

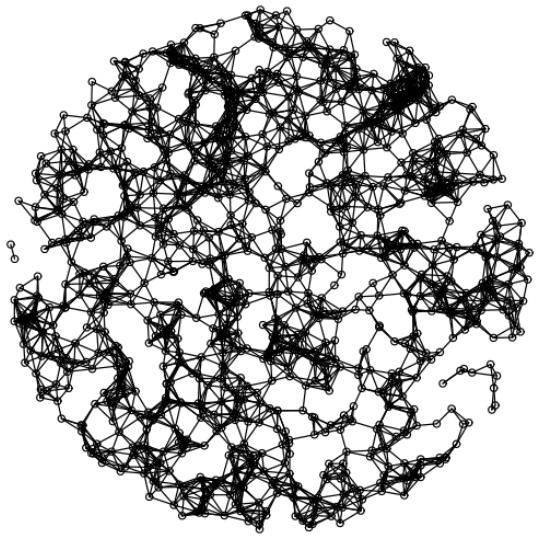
apxCHALT

Einführung

Routing

Problemstellung

- schnelle Bestimmung (fast) optimaler kürzester Wege in einem Verbindungsnetz
- statisches Netz, zentrale Berechnung



Motivation

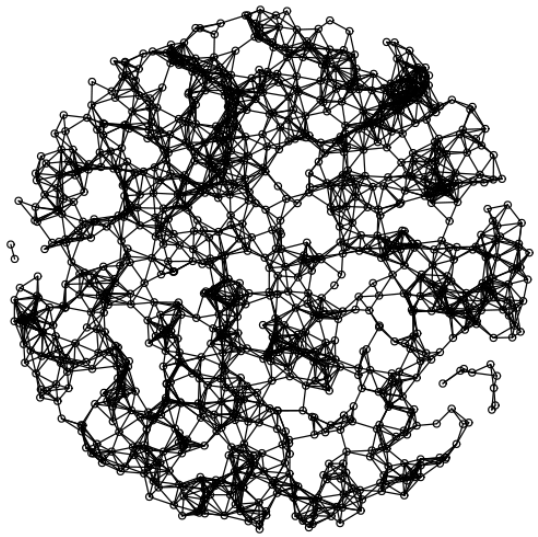
- theoretische Untersuchungen
  - maximale Kapazität des Netzes
  - Erkennung von ”hot spots“
  - Lastverteilung je nach Anfragemuster
- Optimierung der Ressourcenausnutzung z.B. in *data gathering* Anwendungen

Einführung

Routing

Problemstellung

- schnelle Bestimmung (fast) optimaler kürzester Wege in einem Verbindungsnetz
- statisches Netz, zentrale Berechnung



Motivation

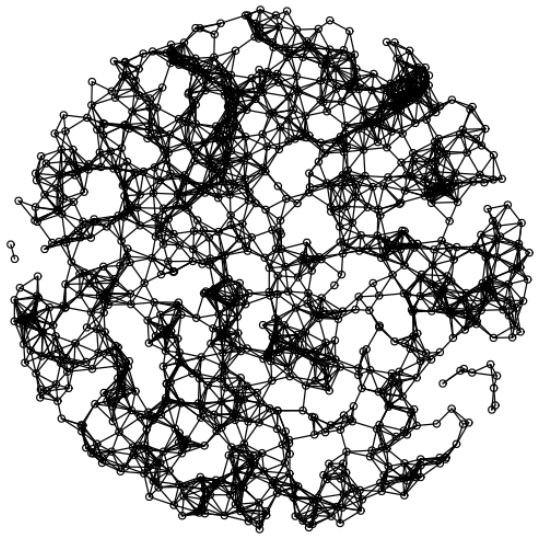
- theoretische Untersuchungen
  - maximale Kapazität des Netzes
  - Erkennung von ”hot spots“
  - Lastverteilung je nach Anfragemuster
- Optimierung der Ressourcenausnutzung z.B. in *data gathering* Anwendungen

Einführung

Routing

Problemstellung

- schnelle Bestimmung (fast) optimaler kürzester Wege in einem Verbindungsnetz
- statisches Netz, zentrale Berechnung



Motivation

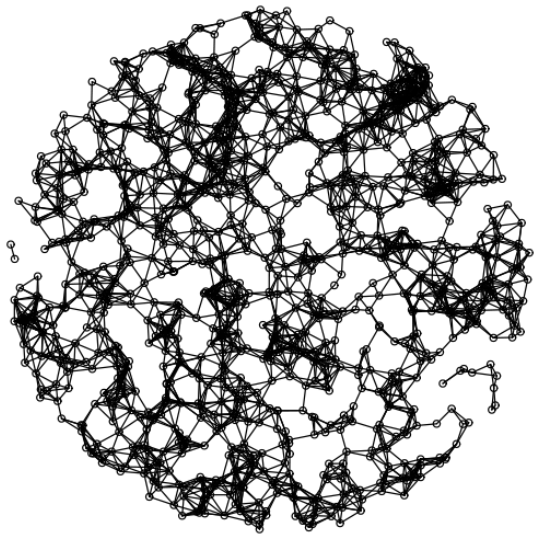
- theoretische Untersuchungen
  - maximale Kapazität des Netzes
  - Erkennung von ”hot spots“
  - Lastverteilung je nach Anfragemuster
- Optimierung der Ressourcenausnutzung z.B. in *data gathering* Anwendungen

Einführung

Routing

Problemstellung

- schnelle Bestimmung (fast) optimaler kürzester Wege in einem Verbindungsnetz
- statisches Netz, zentrale Berechnung



Motivation

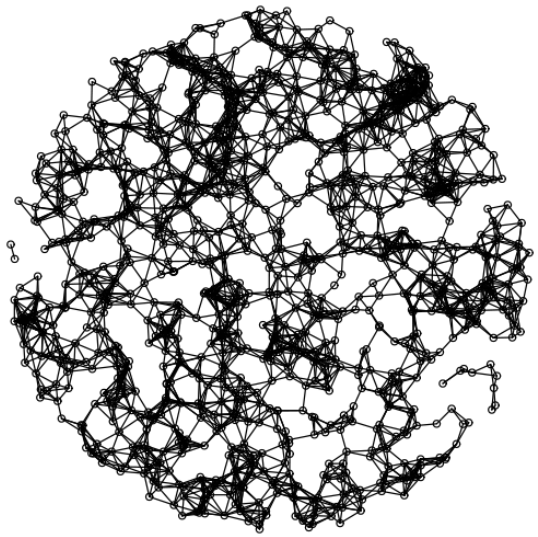
- theoretische Untersuchungen
  - maximale Kapazität des Netzes
  - Erkennung von ”hot spots“
  - Lastverteilung je nach Anfragemuster
- Optimierung der Ressourcenausnutzung z.B. in *data gathering* Anwendungen

Einführung

Routing

Problemstellung

- schnelle Bestimmung (fast) optimaler kürzester Wege in einem Verbindungsnetz
- statisches Netz, zentrale Berechnung



Motivation

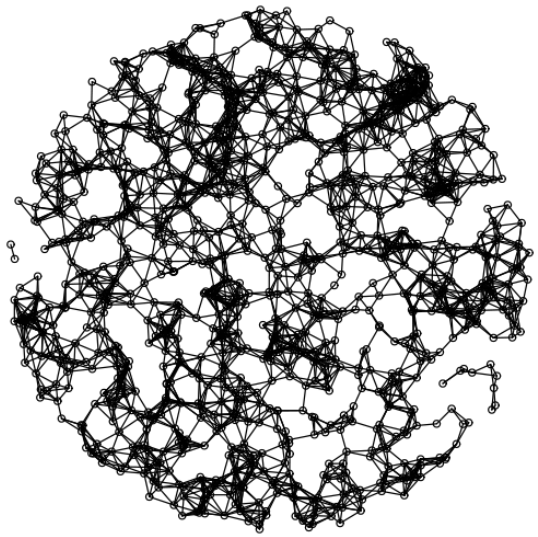
- theoretische Untersuchungen
  - maximale Kapazität des Netzes
  - Erkennung von ”hot spots“
  - Lastverteilung je nach Anfragemuster
- Optimierung der Ressourcenausnutzung z.B. in *data gathering* Anwendungen

Einführung

Routing

Problemstellung

- schnelle Bestimmung (fast) optimaler kürzester Wege in einem Verbindungsnetz
- statisches Netz, zentrale Berechnung



Motivation

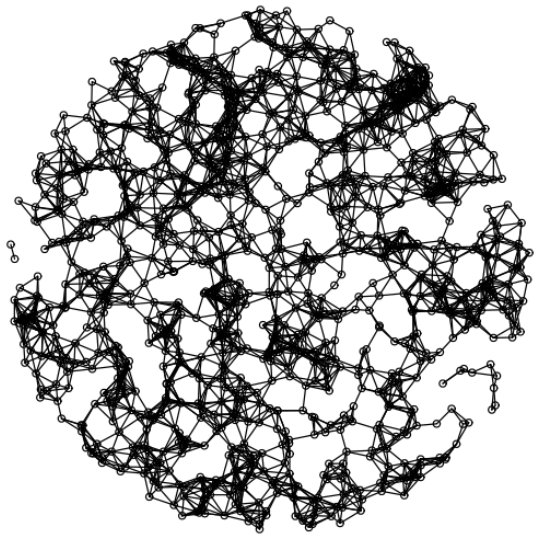
- theoretische Untersuchungen
  - maximale Kapazität des Netzes
  - Erkennung von ”hot spots“
  - Lastverteilung je nach Anfragemuster
- Optimierung der Ressourcenausnutzung z.B. in *data gathering* Anwendungen

Einführung

Routing

Problemstellung

- schnelle Bestimmung (fast) optimaler kürzester Wege in einem Verbindungsnetz
- statisches Netz, zentrale Berechnung



Motivation

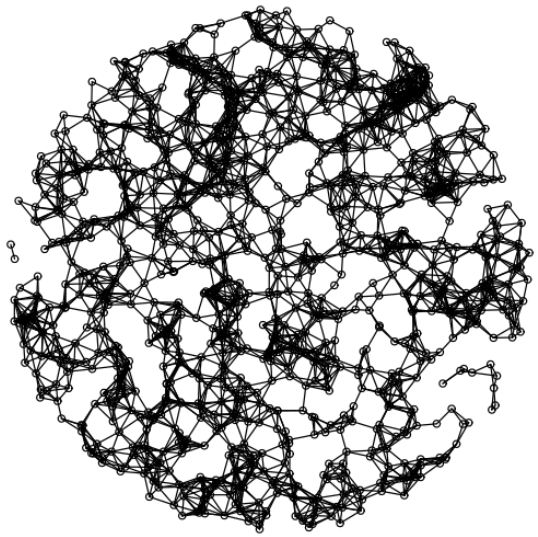
- theoretische Untersuchungen
  - maximale Kapazität des Netzes
  - Erkennung von ”hot spots“
  - Lastverteilung je nach Anfragemuster
- Optimierung der Ressourcenausnutzung z.B. in *data gathering* Anwendungen

Einführung

Routing

Problemstellung

- schnelle Bestimmung (fast) optimaler kürzester Wege in einem Verbindungsnetz
- statisches Netz, zentrale Berechnung



Motivation

- theoretische Untersuchungen
  - maximale Kapazität des Netzes
  - Erkennung von ”hot spots“
  - Lastverteilung je nach Anfragemuster
- Optimierung der Ressourcenausnutzung z.B. in *data gathering* Anwendungen

Einführung
Routing

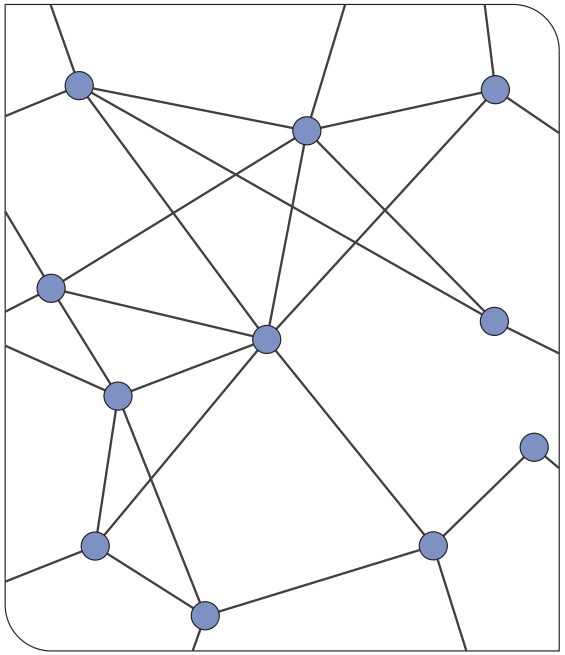
Netzwerkmodell

- Graph $G = (V, E)$ mit Knoten V, Kanten E
- Kostenfunktion $\mathrm{c} : E \to \mathbb{R}_+$

gesucht

- Pfad $\langle s, \ldots, t \rangle$ mit minimaler Distanz $\mathrm{d}(s, t) = \mathrm{d}_{opt}(s, t)$

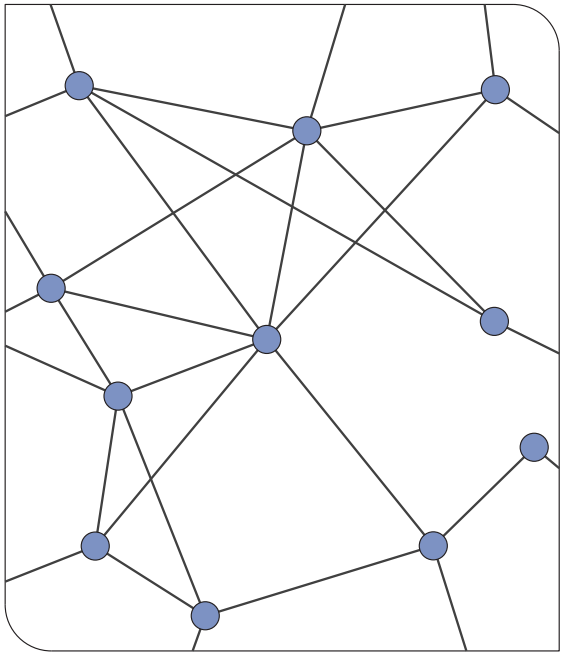
Lösung: Algorithmus von Dijkstra



Einführung
Routing

Netzwerkmodell

- Graph $G = (V, E)$ mit Knoten V, Kanten E
- Kostenfunktion $c : E \to \mathbb{R}_+$



gesucht

- Pfad $\langle s, \ldots, t \rangle$ mit minimaler Distanz $d(s, t) = d_{opt}(s, t)$

Lösung: Algorithmus von Dijkstra

Einführung
Routing

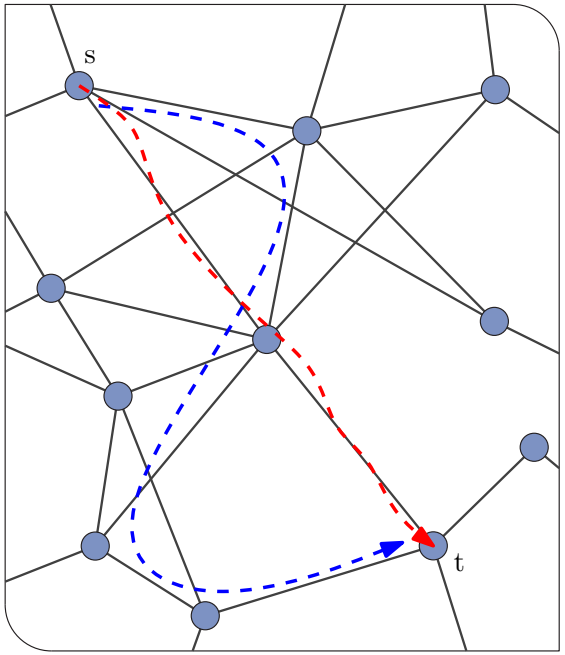
Netzwerkmodell

- Graph $G = (V, E)$ mit Knoten V, Kanten E
- Kostenfunktion $\mathrm{c} : E \to \mathbb{R}_+$

gesucht

- Pfad $\langle s, \ldots, t \rangle$ mit minimaler Distanz $\mathrm{d}(s, t) = \mathrm{d}_{opt}(s, t)$

Lösung: Algorithmus von Dijkstra



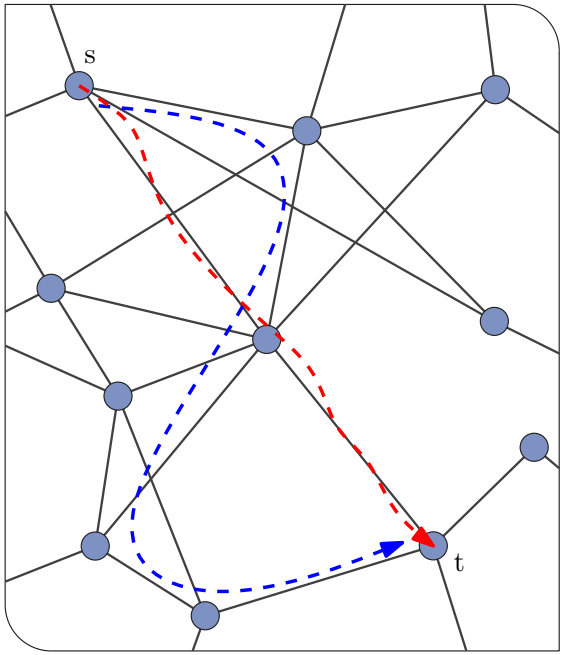
Einführung
Routing

Netzwerkmodell

- Graph $G = (V, E)$ mit Knoten V, Kanten E
- Kostenfunktion $\mathrm{c} : E \to \mathbb{R}_+$

gesucht

- Pfad $\langle s, \ldots, t \rangle$ mit minimaler Distanz $\mathrm{d}(s, t) = \mathrm{d}_{opt}(s, t)$



Lösung: Algorithmus von Dijkstra

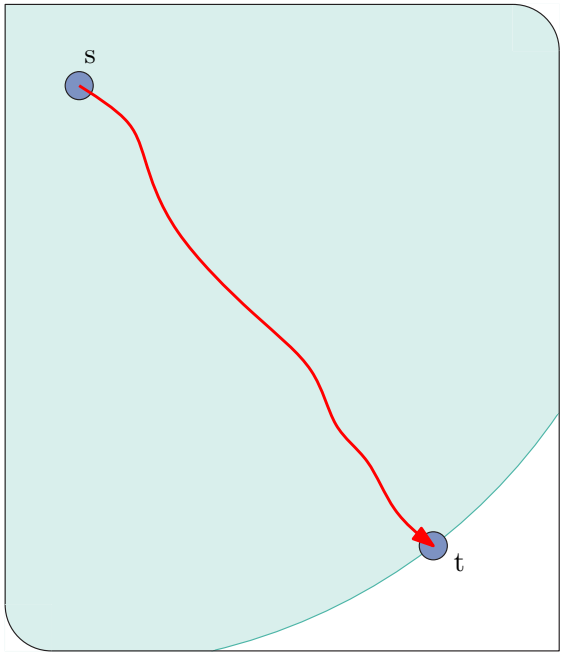
Einführung
Routing

Netzwerkmodell

- Graph $G = (V, E)$ mit Knoten V, Kanten E
- Kostenfunktion $\mathrm{c} : E \to \mathbb{R}_+$

gesucht

- Pfad $\langle s, \ldots, t \rangle$ mit minimaler Distanz $\mathrm{d}(s, t) = \mathrm{d}_{opt}(s, t)$



Lösung: Algorithmus von Dijkstra — langsam!

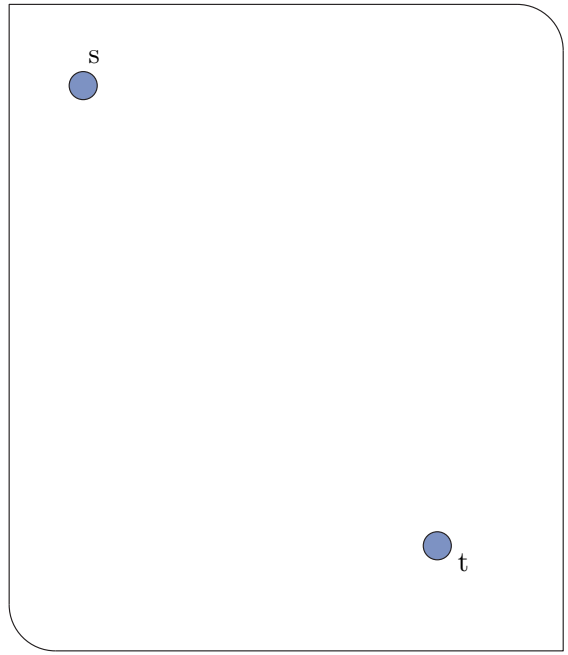
Einführung
Vorherige Arbeiten

Beschleunigungsverfahren

- bidirektional (bidr. Dijkstra)
- zielgerichtet (A*, ALT, ArcFlags)
- hierarchisch (CH, Reach)
- Kombinationen (CALT, CHASE, Real)

Vorverarbeitungsschritt

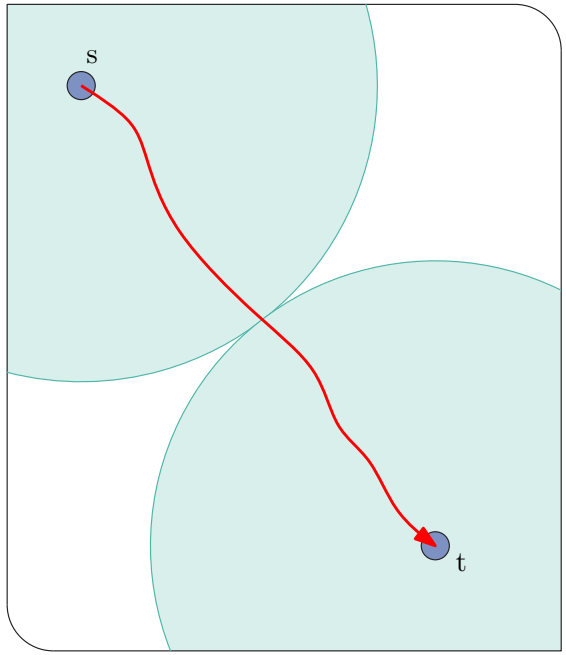
- Aggregation von Hilfsinformationen
- Verlagerung von Zeit- in Speicheraufwand
- gut bei vielen Anfragen auf gleichem Graph



Einführung
Vorherige Arbeiten

Beschleunigungsverfahren

- bidirektional (bidr. Dijkstra)
- zielgerichtet (A*, ALT, ArcFlags)
- hierarchisch (CH, Reach)
- Kombinationen (CALT, CHASE, Real)



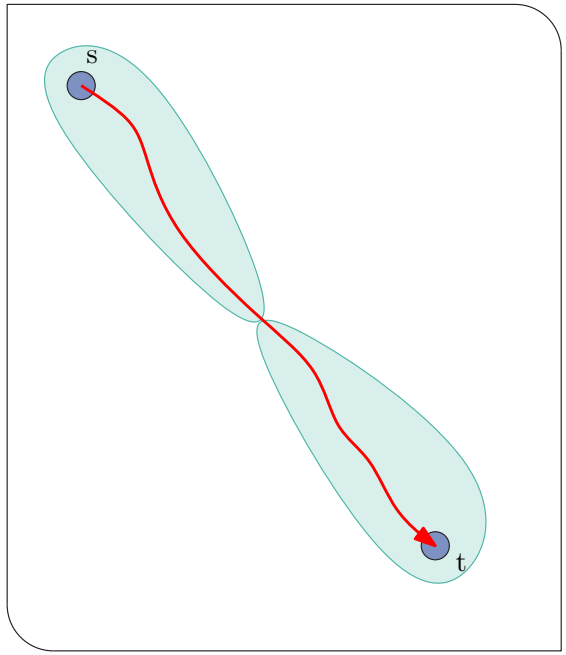
Vorverarbeitungsschritt

- Aggregation von Hilfsinformationen
- Verlagerung von Zeit- in Speicheraufwand
- gut bei vielen Anfragen auf gleichem Graph

Einführung
Vorherige Arbeiten

Beschleunigungsverfahren

- bidirektional (bidr. Dijkstra)
- zielgerichtet (A*, ALT, ArcFlags)
- hierarchisch (CH, Reach)
- Kombinationen (CALT, CHASE, Real)



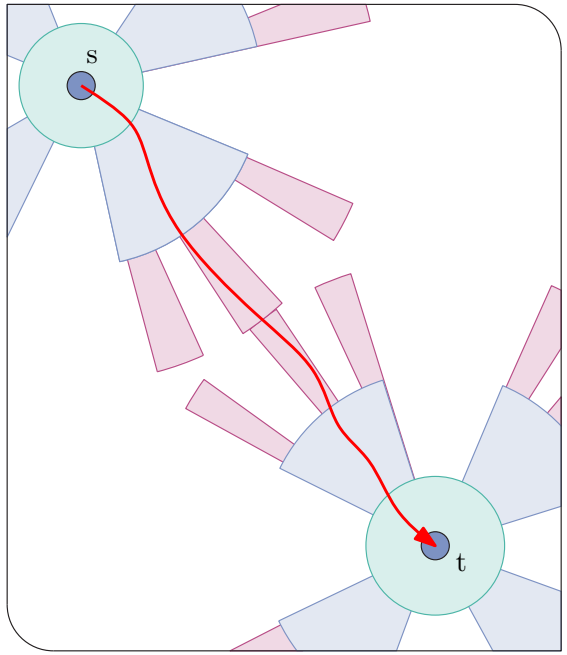
Vorverarbeitungsschritt

- Aggregation von Hilfsinformationen
- Verlagerung von Zeit- in Speicheraufwand
- gut bei vielen Anfragen auf gleichem Graph

Einführung
Vorherige Arbeiten

Beschleunigungsverfahren

- bidirektional (bidr. Dijkstra)
- zielgerichtet (A*, ALT, ArcFlags)
- hierarchisch (CH, Reach)
- Kombinationen (CALT, CHASE, Real)



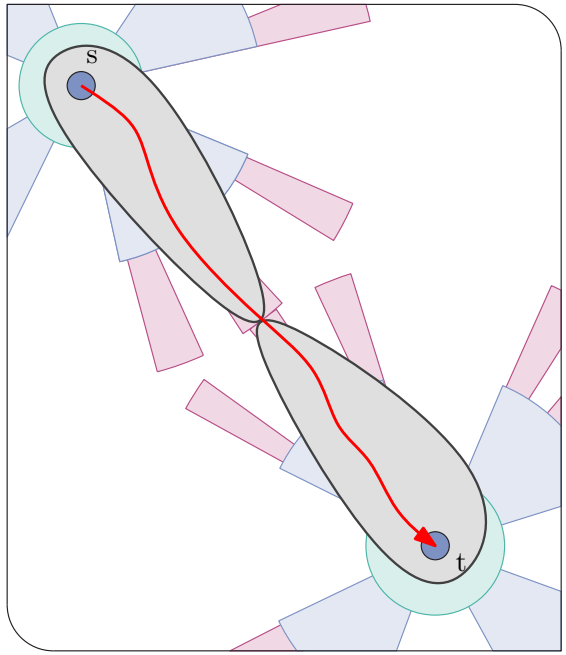
Vorverarbeitungsschritt

- Aggregation von Hilfsinformationen
- Verlagerung von Zeit- in Speicheraufwand
- gut bei vielen Anfragen auf gleichem Graph

Einführung
Vorherige Arbeiten

Beschleunigungsverfahren
- bidirektional (bidr. Dijkstra)
- zielgerichtet (A*, ALT, ArcFlags)
- hierarchisch (CH, Reach)
- Kombinationen (CALT, CHASE, Real)

Vorverarbeitungsschritt
- Aggregation von Hilfsinformationen
- Verlagerung von Zeit- in Speicheraufwand
- gut bei vielen Anfragen auf gleichem Graph



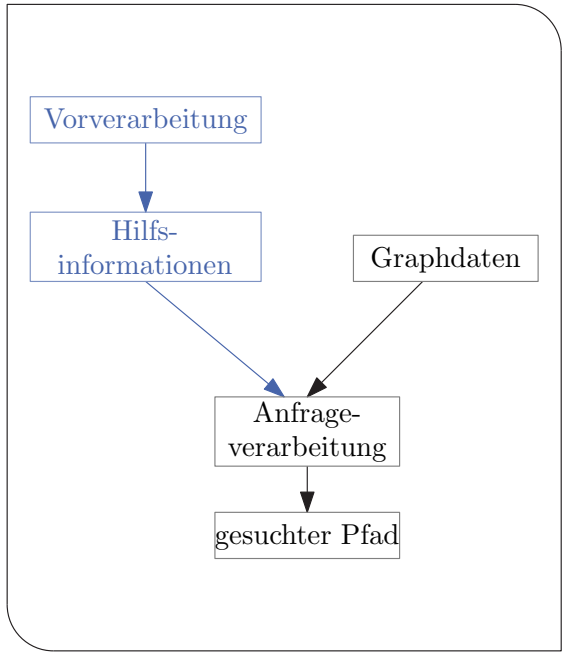
Einführung
Vorherige Arbeiten

Beschleunigungsverfahren

- bidirektional (bidr. Dijkstra)
- zielgerichtet (A*, ALT, ArcFlags)
- hierarchisch (CH, Reach)
- Kombinationen (CALT, CHASE, Real)

Vorverarbeitungsschritt

- Aggregation von Hilfsinformationen
- Verlagerung von Zeit- in Speicheraufwand
- gut bei vielen Anfragen auf gleichem Graph



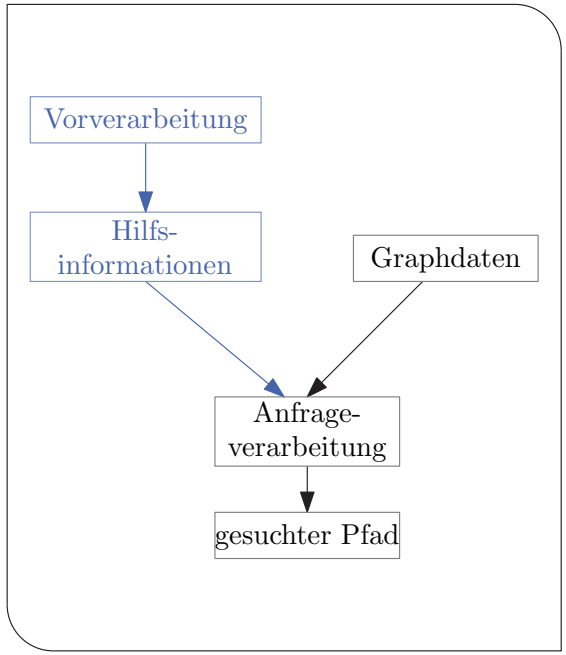
Einführung
Vorherige Arbeiten

Beschleunigungsverfahren

- bidirektional (bidr. Dijkstra)
- zielgerichtet (A*, ALT, ArcFlags)
- hierarchisch (CH, Reach)
- Kombinationen (CALT, CHASE, Real)

Vorverarbeitungsschritt

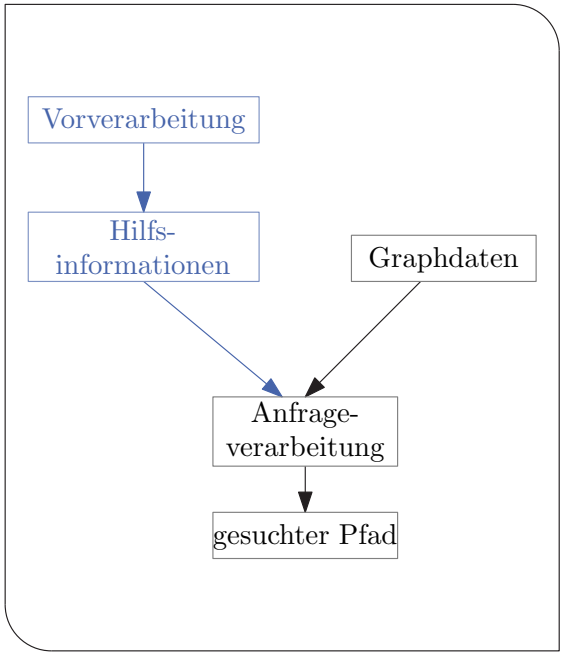
- Aggregation von Hilfsinformationen
- Verlagerung von Zeit- in Speicheraufwand
- gut bei vielen Anfragen auf gleichem Graph



Einführung
Vorherige Arbeiten

Beschleunigungsverfahren

- bidirektional (bidr. Dijkstra)
- zielgerichtet (A*, ALT, ArcFlags)
- hierarchisch (CH, Reach)
- Kombinationen (CALT, CHASE, Real)



Vorverarbeitungsschritt

- Aggregation von Hilfsinformationen
- Verlagerung von Zeit- in Speicheraufwand
- gut bei vielen Anfragen auf gleichem Graph

unterschiedliche Kompromisse zwischen Vorverarbeitungs-
dauer, Speicheraufwand und Anfragedauer

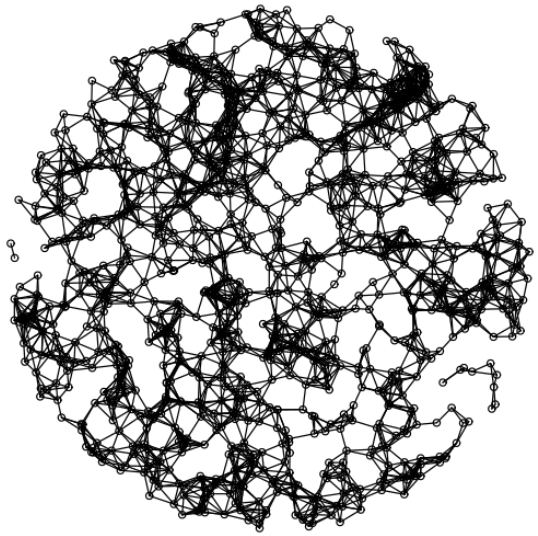
Routingalgorithmus

approximierte Contraction Hierarchies mit ALT (apxCHALT)

Karlsruher Institut für Technologie

Grundlage:

- Contraction Hierarchies (CH)
- Probleme:
  - ungeeignet für dichtere Graphen
  - lange Vorberechnung, hoher Platzbedarf, langsame Anfragen



Verbesserungen durch

- inexakte Pfade mit Approximationsgarantie
  ($\mathrm{d}(s,t) \leq (1+\varepsilon) \cdot \mathrm{d}_{opt}(s,t)$)
- Kombination mit (weighted) ALT

$\Rightarrow$ bisher bestes Verfahren für Sensornetze

Contraction Hierarchies

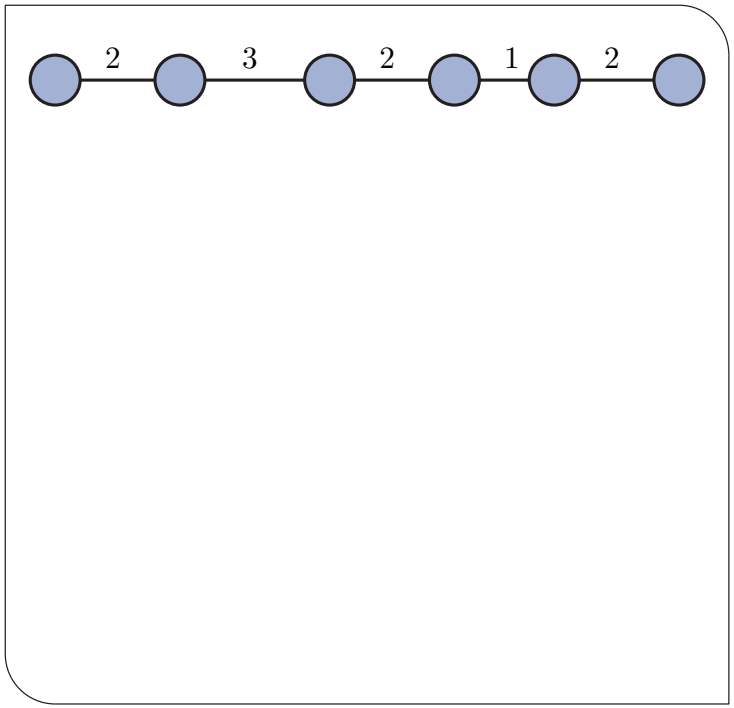
Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*

Suche

- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

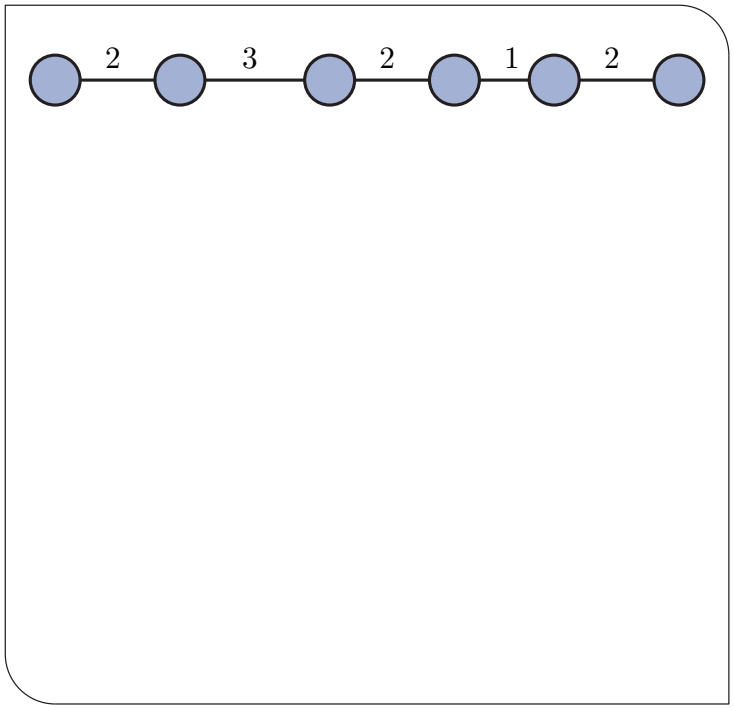


Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

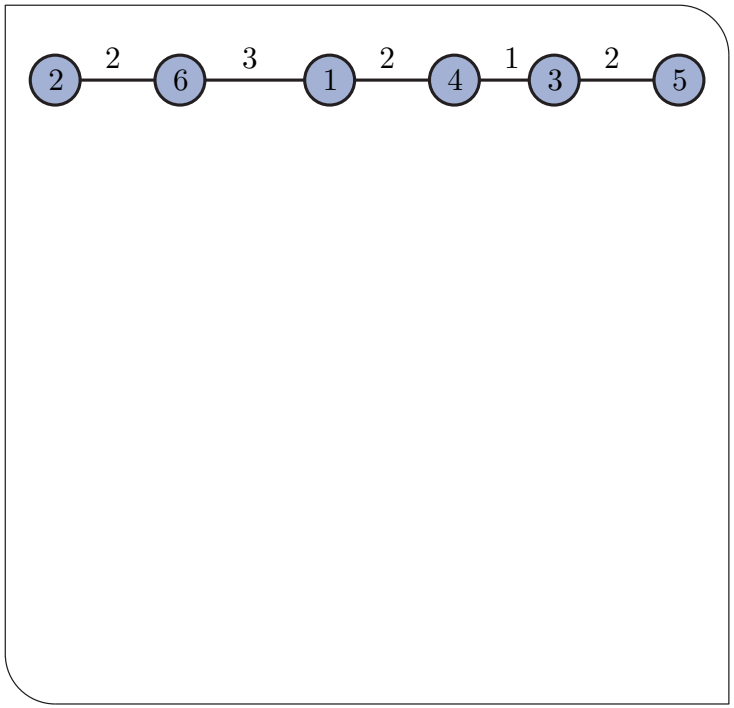
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

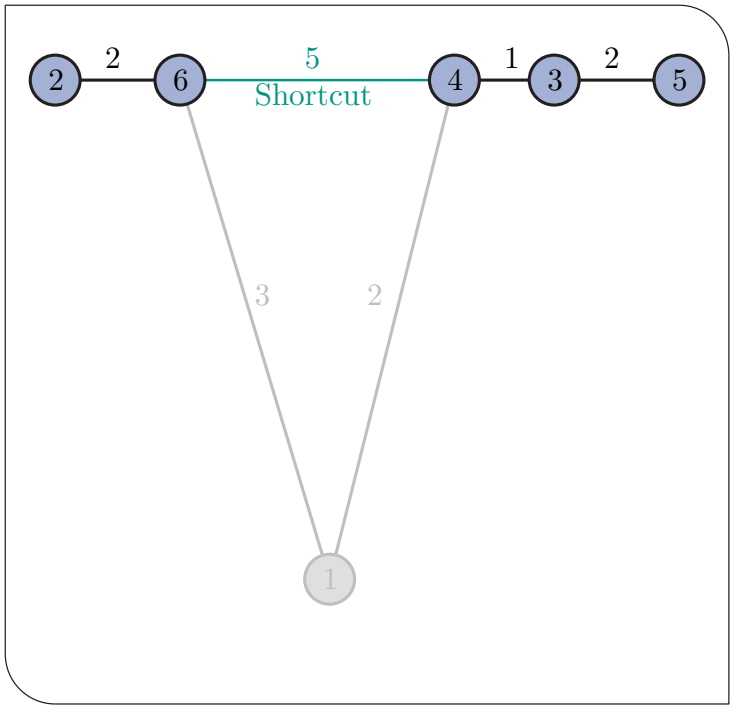
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

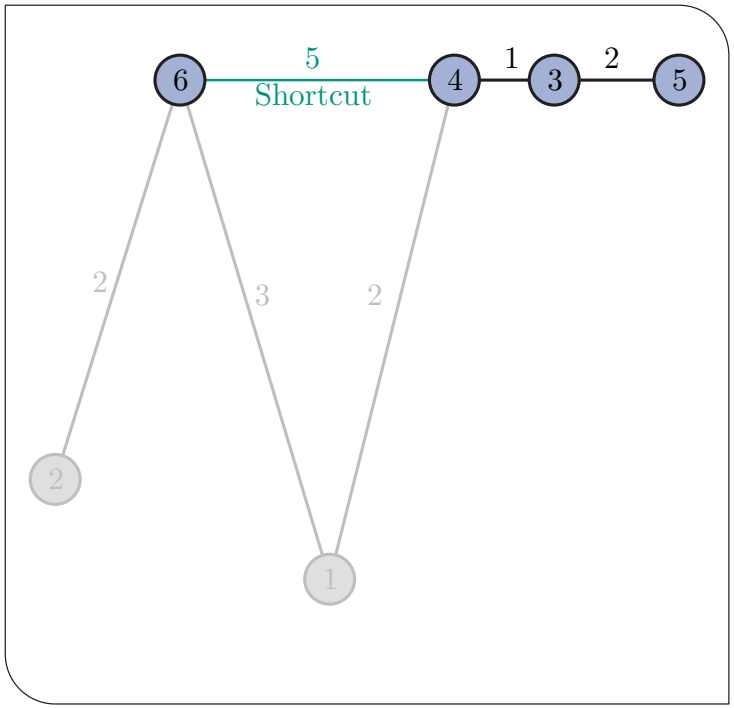
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

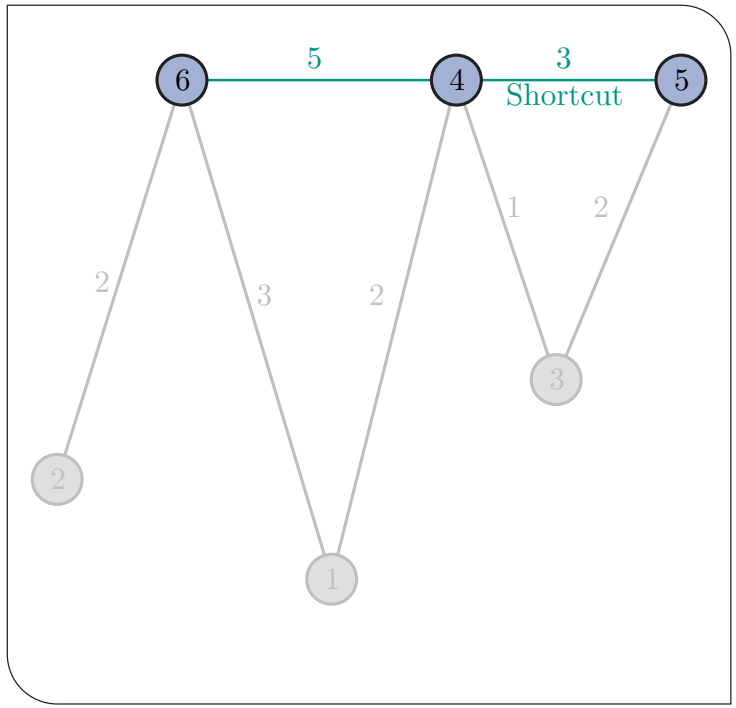
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

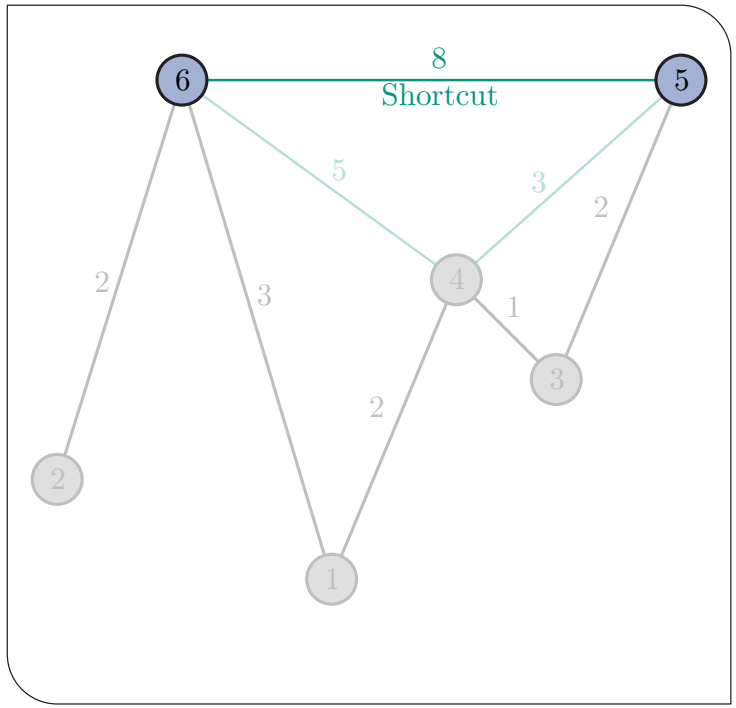
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

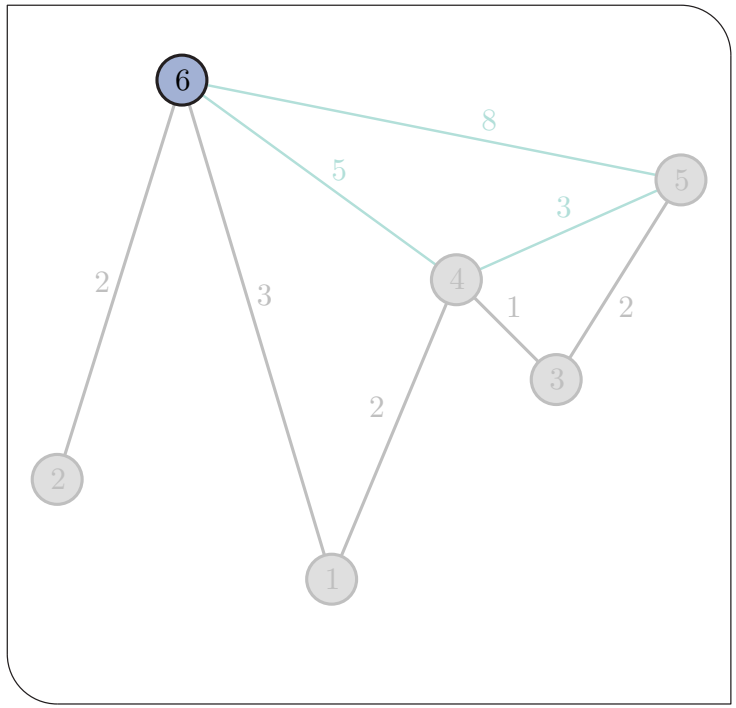
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

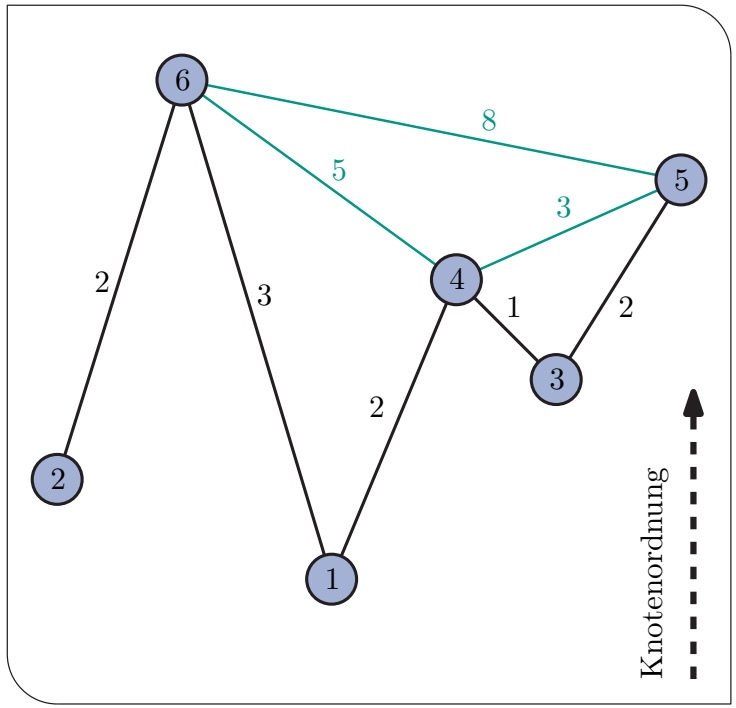
Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*

Suche

- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

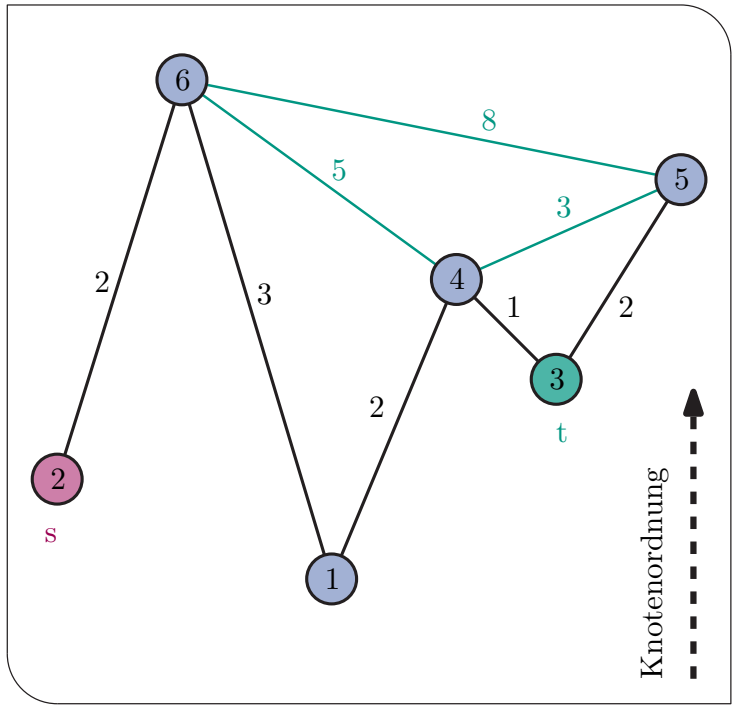


Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

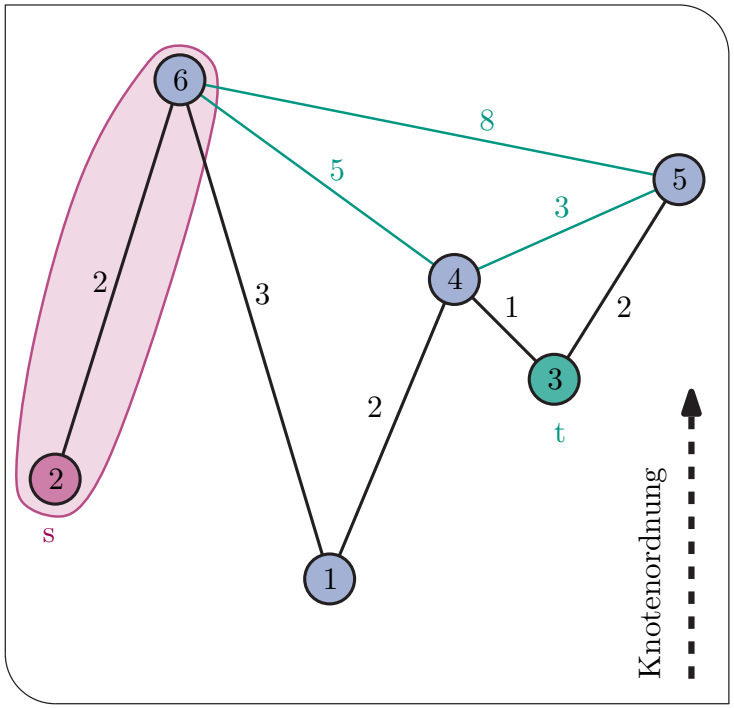
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

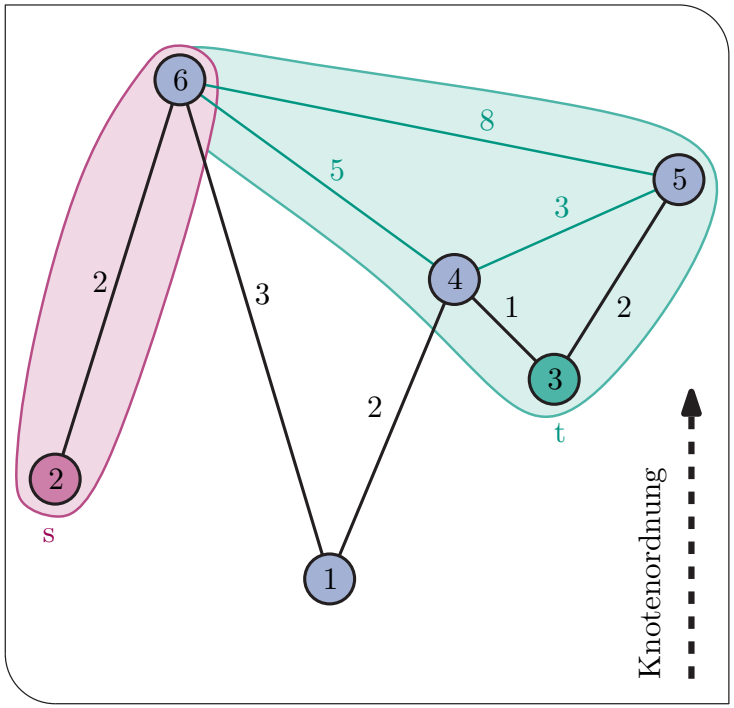
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



Suche

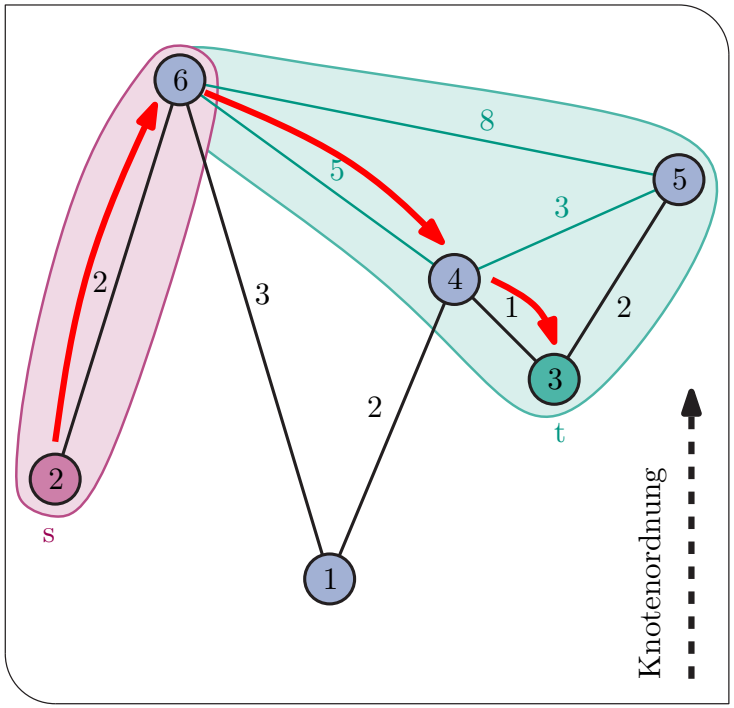
- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

Contraction Hierarchies

Überblick

Vorverarbeitung

- ordne Knoten nach ”Wichtigkeit“
- erzeuge $|V|$-Level Hierarchie durch iteratives Kontrahieren der Knoten
- erhalte kürzeste Wege in jedem Level durch Hinzufügen von *Shortcuts*



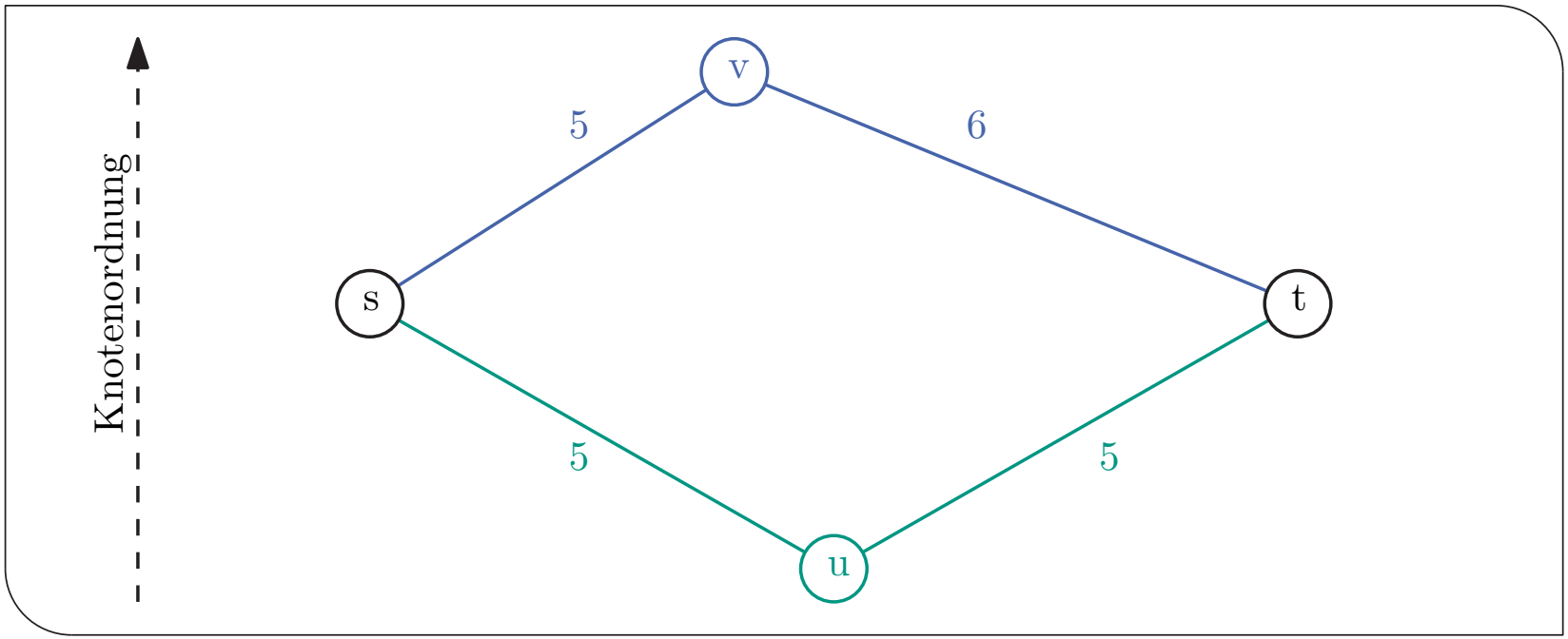
Suche

- suche bidirektional von Start s und Ziel t aufwärts in der Hierarchie (modifizierter Dijkstras Algorithmus)

approximierte Contraction Hierarchies

Grundidee

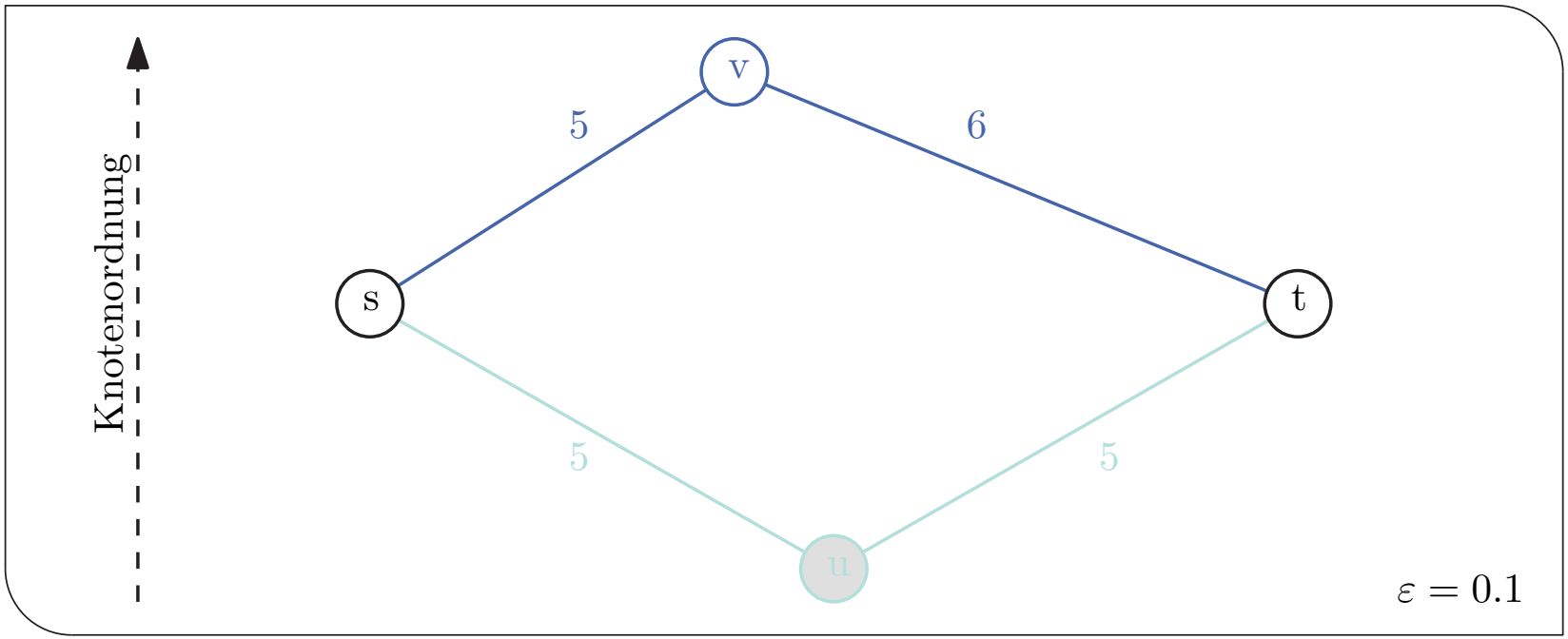
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1 + \varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

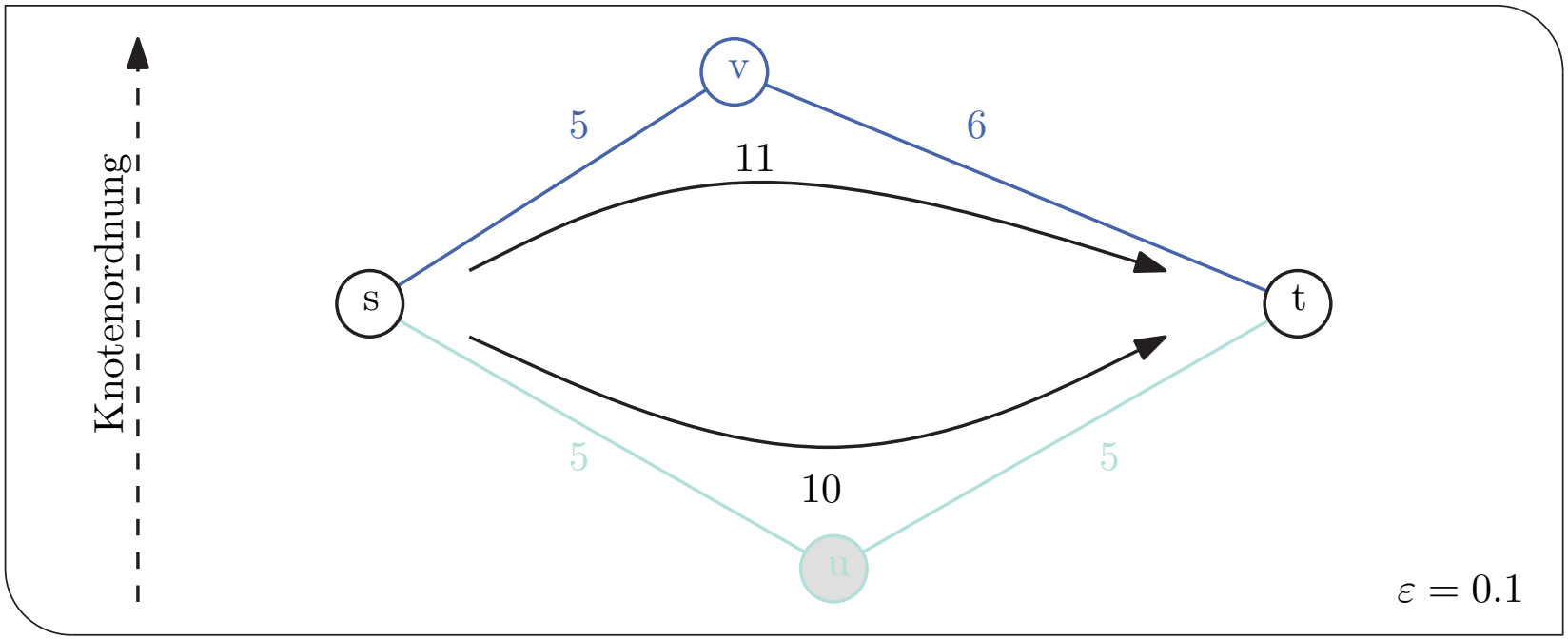
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1 + \varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

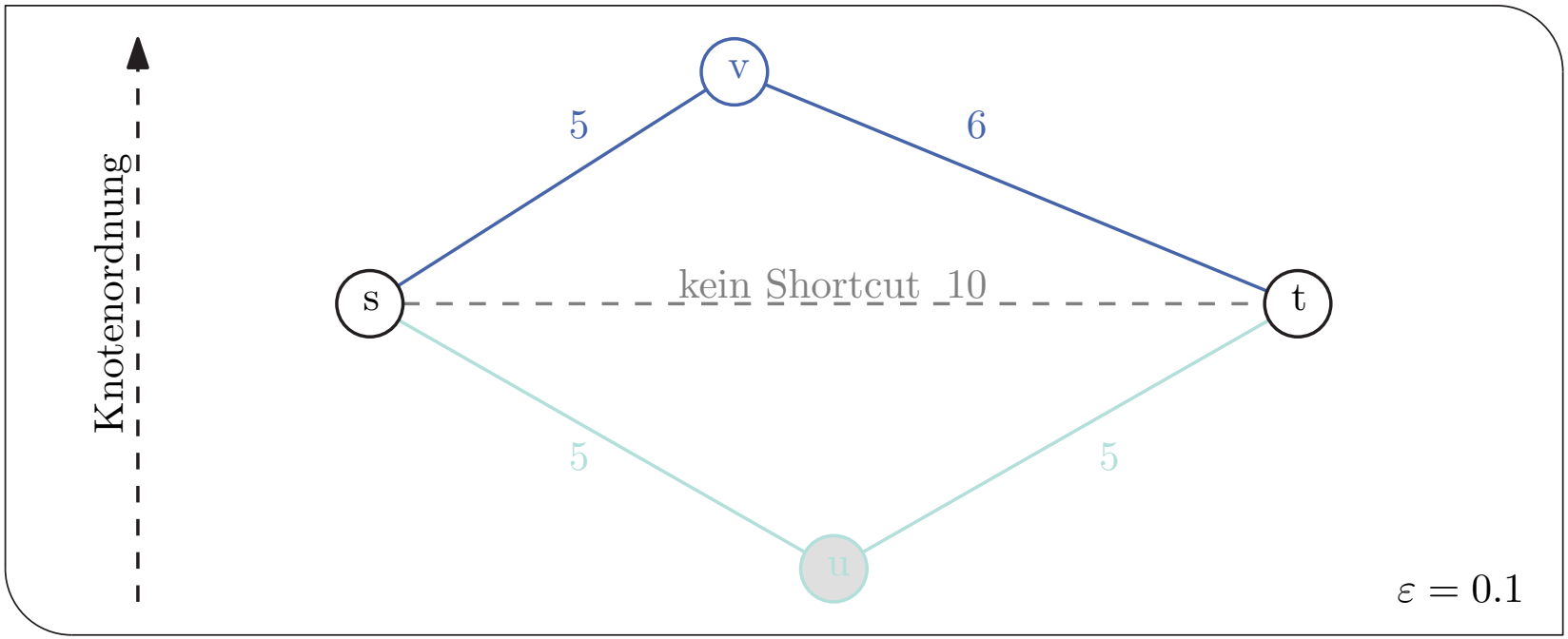
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1 + \varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

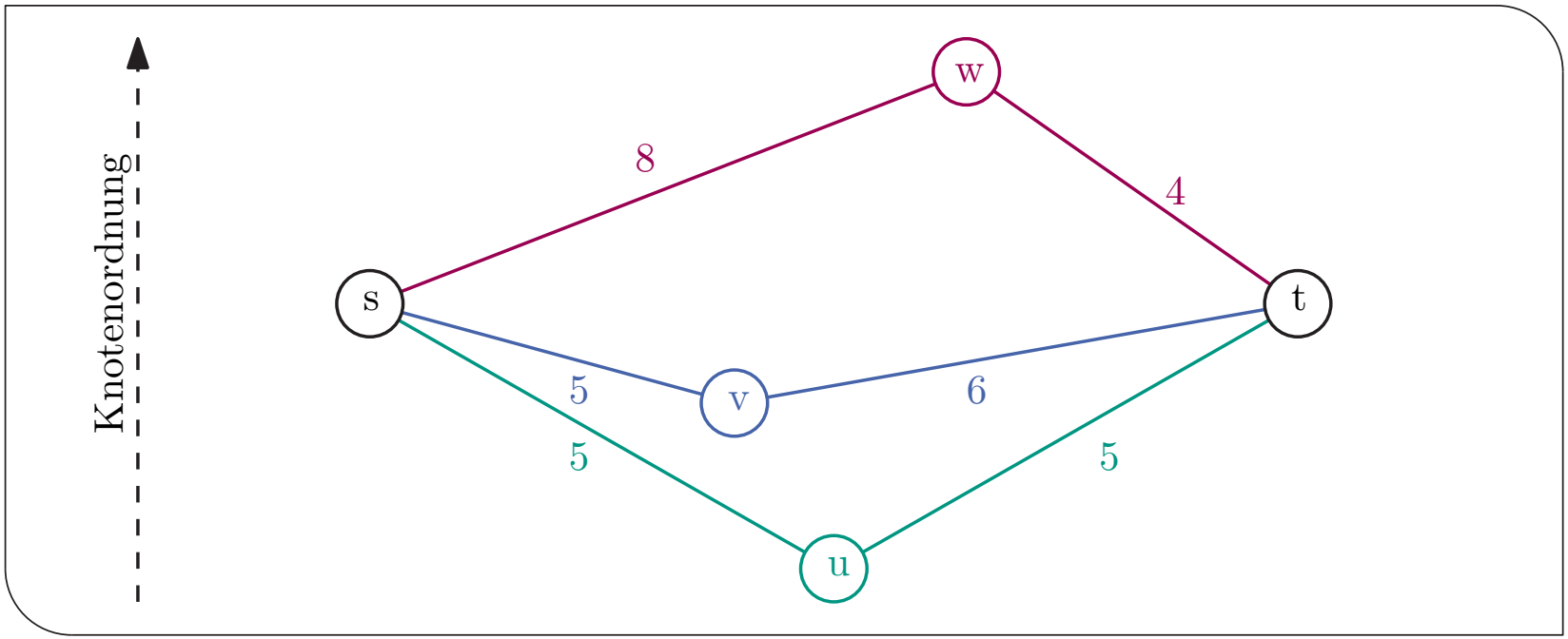
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1 + \varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

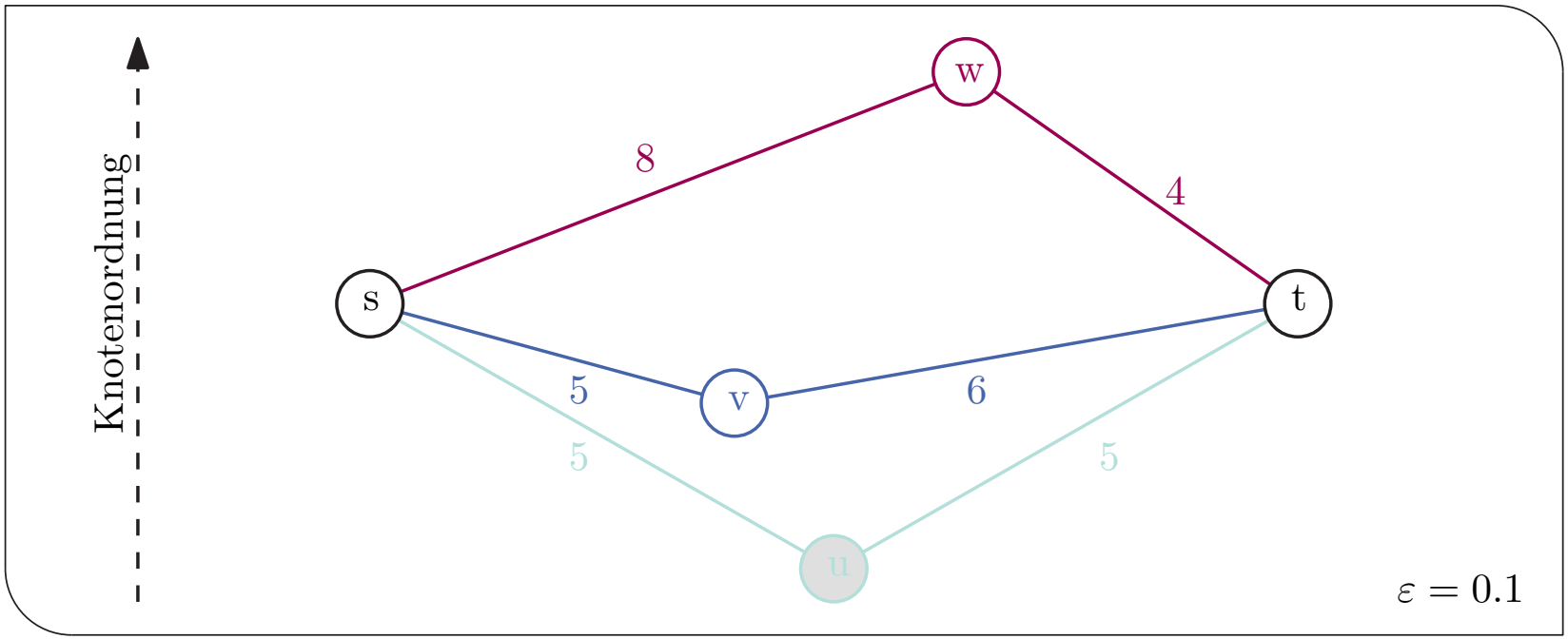
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1+\varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

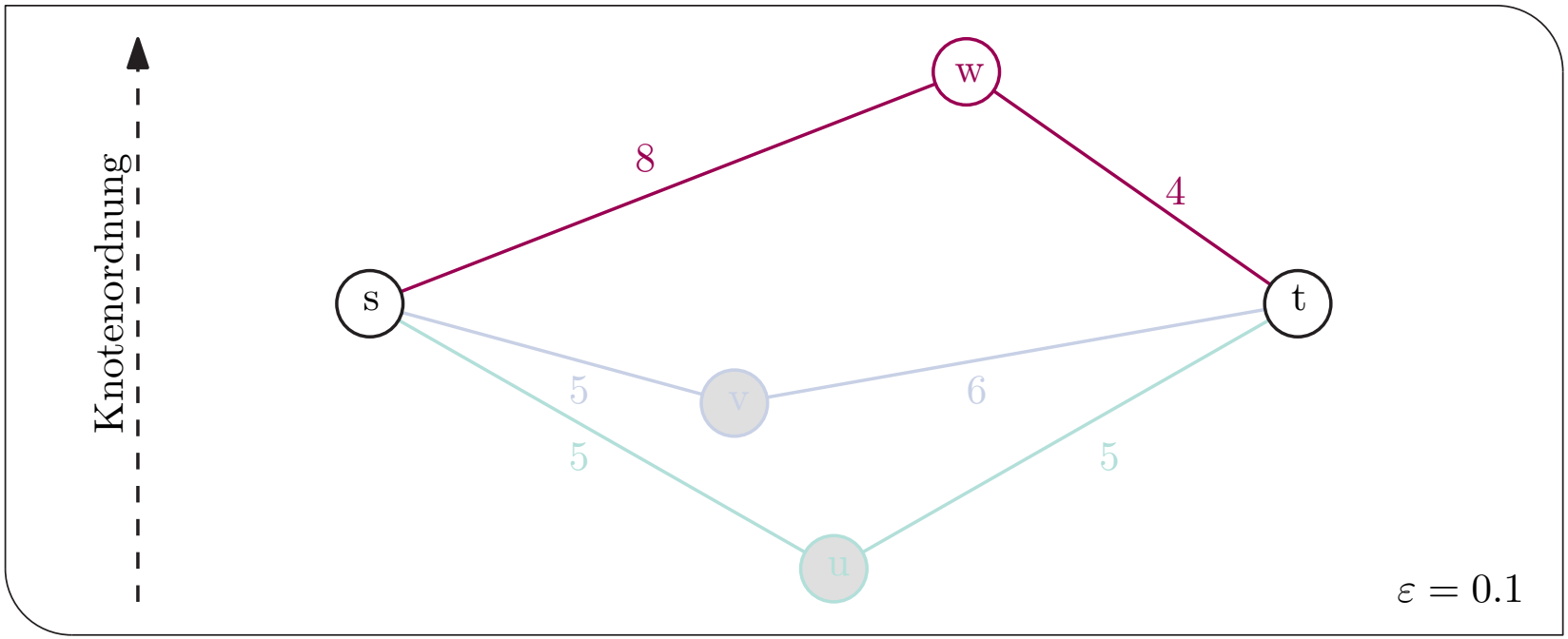
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1 + \varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

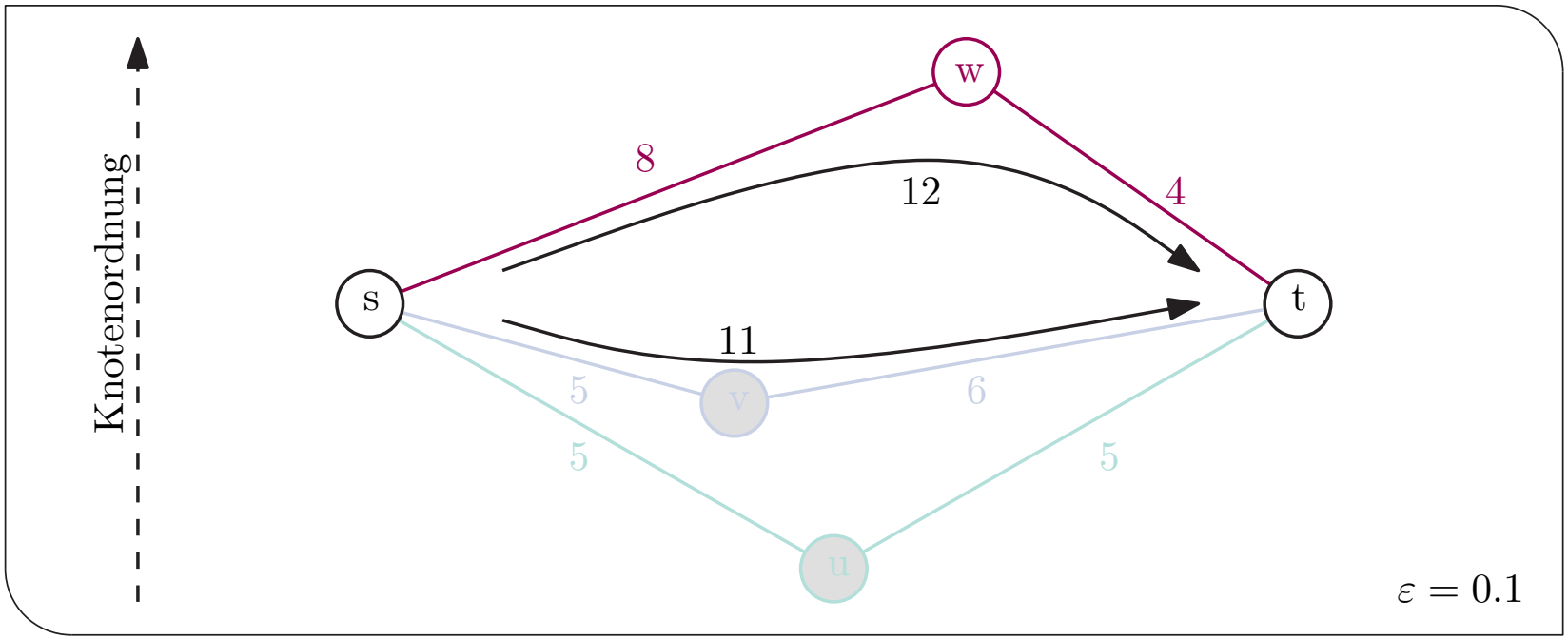
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1 + \varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

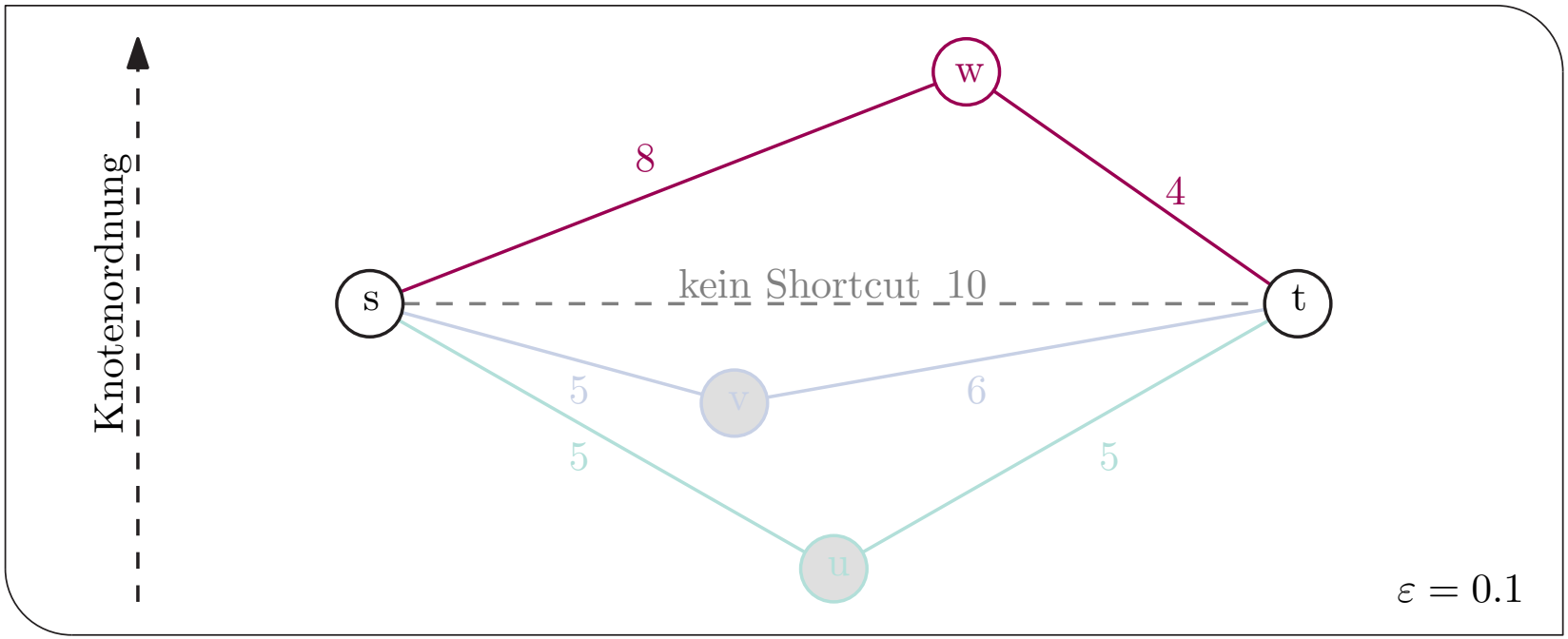
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1+\varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

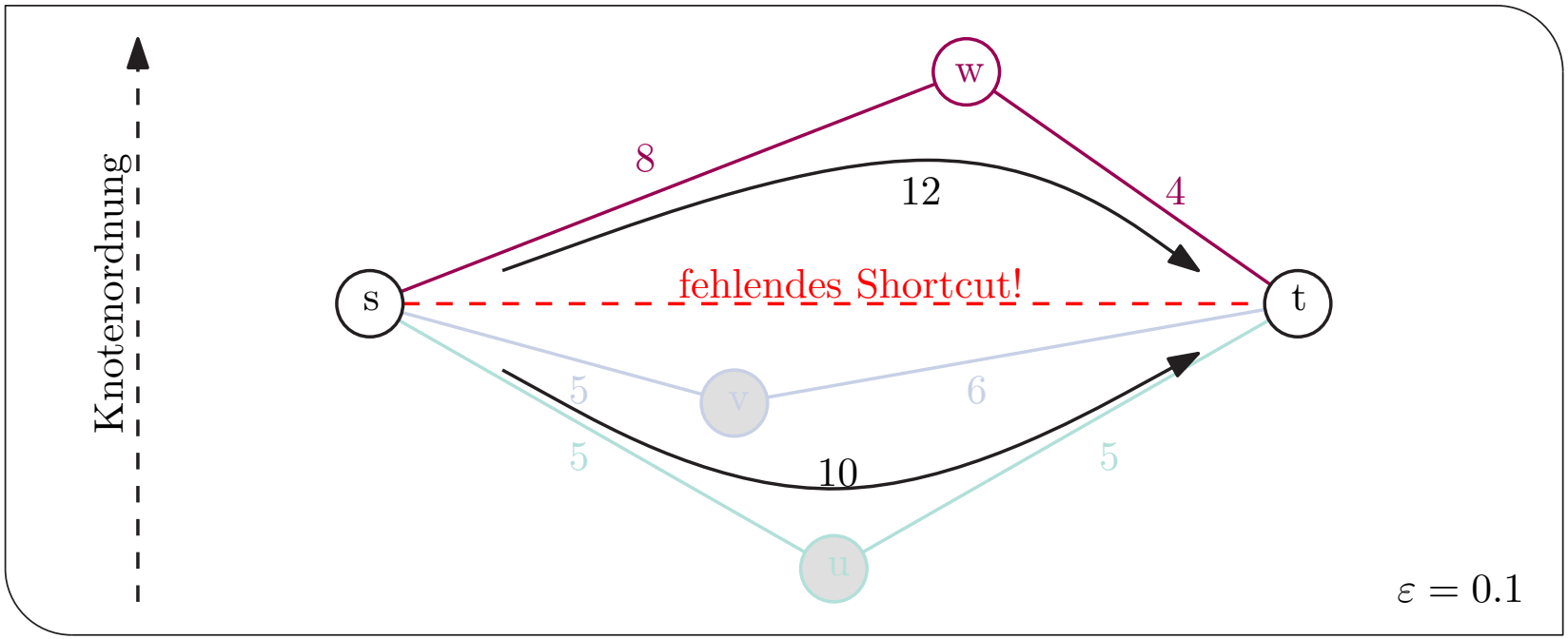
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1 + \varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Grundidee

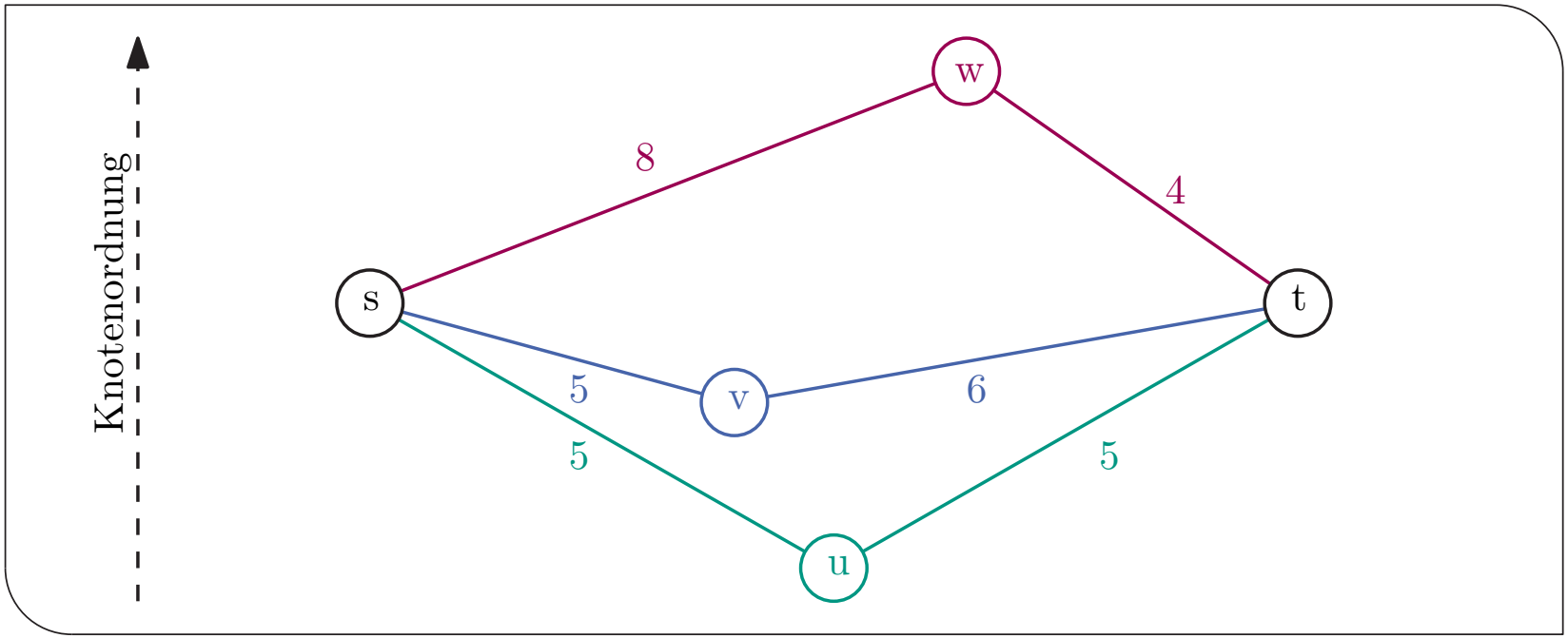
- erzeuge kein *Shortcut*, falls ein Pfad (*Umweg*) exisitert, der maximal Faktor $(1 + \varepsilon)$ länger ist, als der entfernte
- Problem: naive Implementierung führt zu wachsendem Fehler



approximierte Contraction Hierarchies

Details

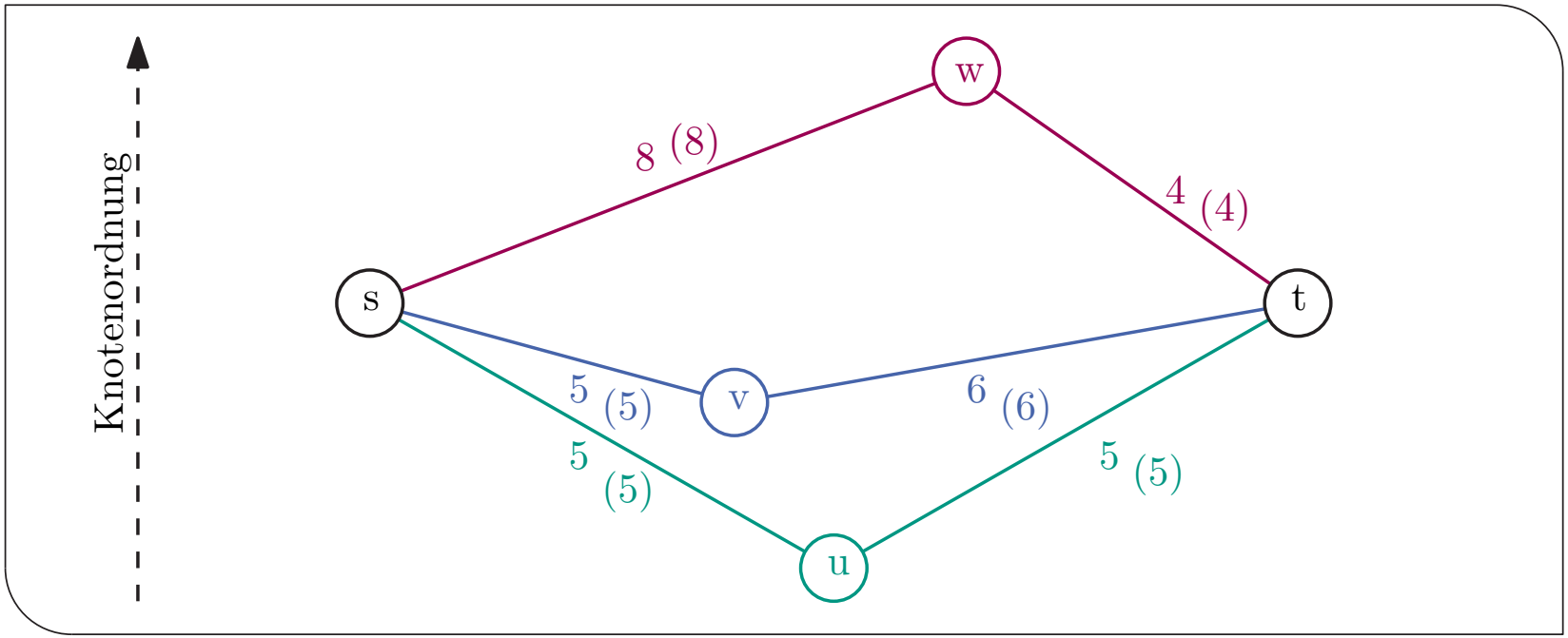
- speichere zusätzliches Gewicht $\tilde{c}$ (zu Beginn: $\tilde{c} = c$)
- erzeuge *Shortcut* falls $c(\text{Umweg}) > (1+\varepsilon)\tilde{c}(\text{Shortcut})$
- sonst, ändere $\tilde{c}(x,y) = \min\left\{\tilde{c}(x,y), \tilde{c}(\text{Shortcut})\frac{c(x,y)}{c(\text{Umweg})}\right\}$ auf allen Kanten (x,y) des *Umwegs*, der den *Shortcut* verhinderte.



approximierte Contraction Hierarchies

Details

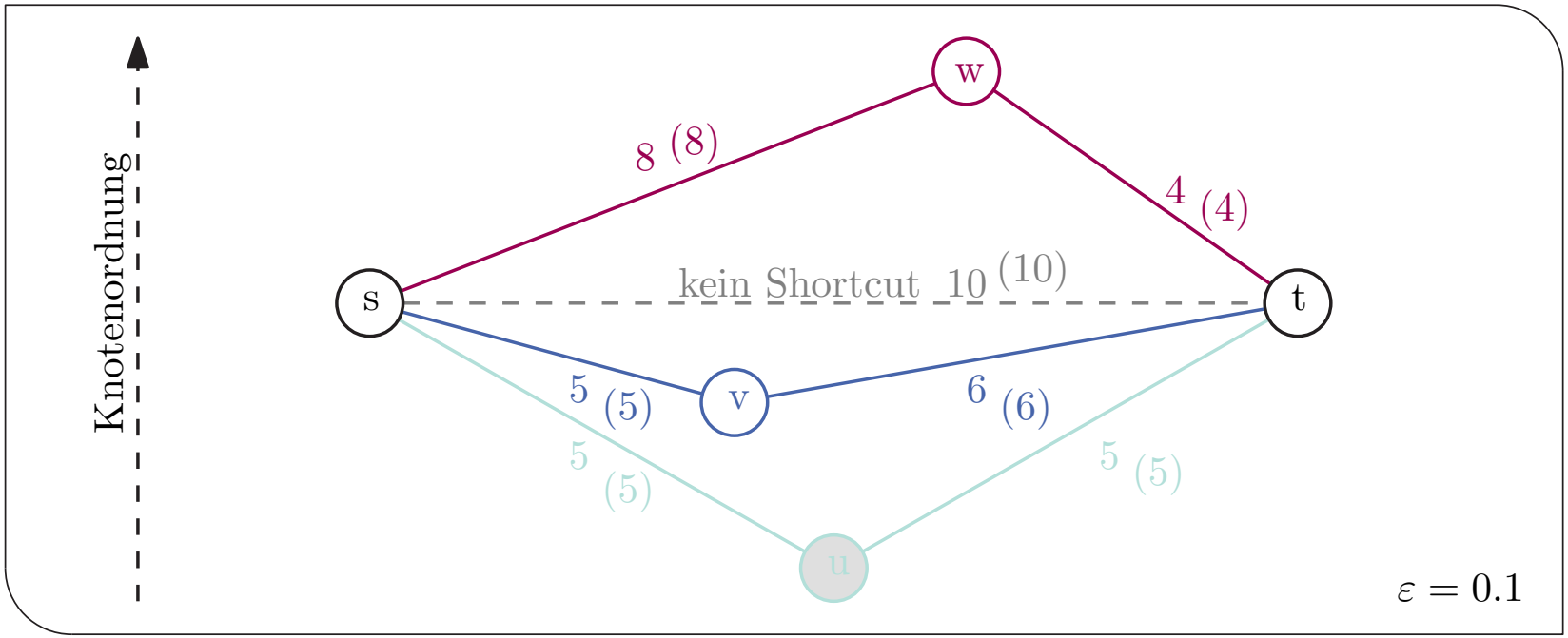
- speichere zusätzliches Gewicht $\tilde{c}$ (zu Beginn: $\tilde{c} = c$)
- erzeuge *Shortcut* falls $c(\text{Umweg}) > (1+\varepsilon)\tilde{c}(\text{Shortcut})$
- sonst, ändere $\tilde{c}(x,y) = \min\left\{\tilde{c}(x,y), \tilde{c}(\text{Shortcut})\frac{c(x,y)}{c(\text{Umweg})}\right\}$ auf allen Kanten (x,y) des *Umwegs*, der den *Shortcut* verhinderte.



approximierte Contraction Hierarchies

Details

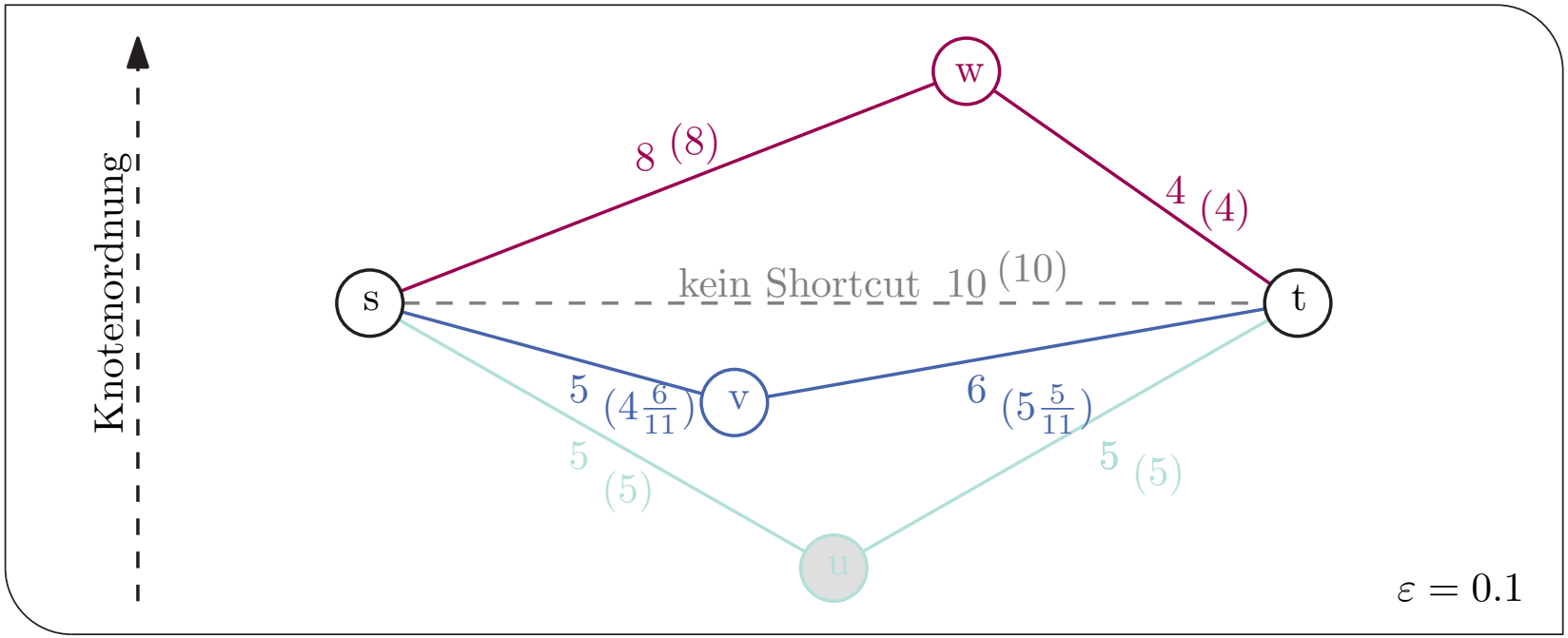
- speichere zusätzliches Gewicht $\tilde{c}$ (zu Beginn: $\tilde{c} = c$)
- erzeuge *Shortcut* falls $c(\text{Umweg}) > (1+\varepsilon)\tilde{c}(\text{Shortcut})$
- sonst, ändere $\tilde{c}(x,y) = \min\left\{\tilde{c}(x,y), \tilde{c}(\text{Shortcut})\frac{c(x,y)}{c(\text{Umweg})}\right\}$ auf allen Kanten (x,y) des *Umwegs*, der den *Shortcut* verhinderte.



approximierte Contraction Hierarchies

Details

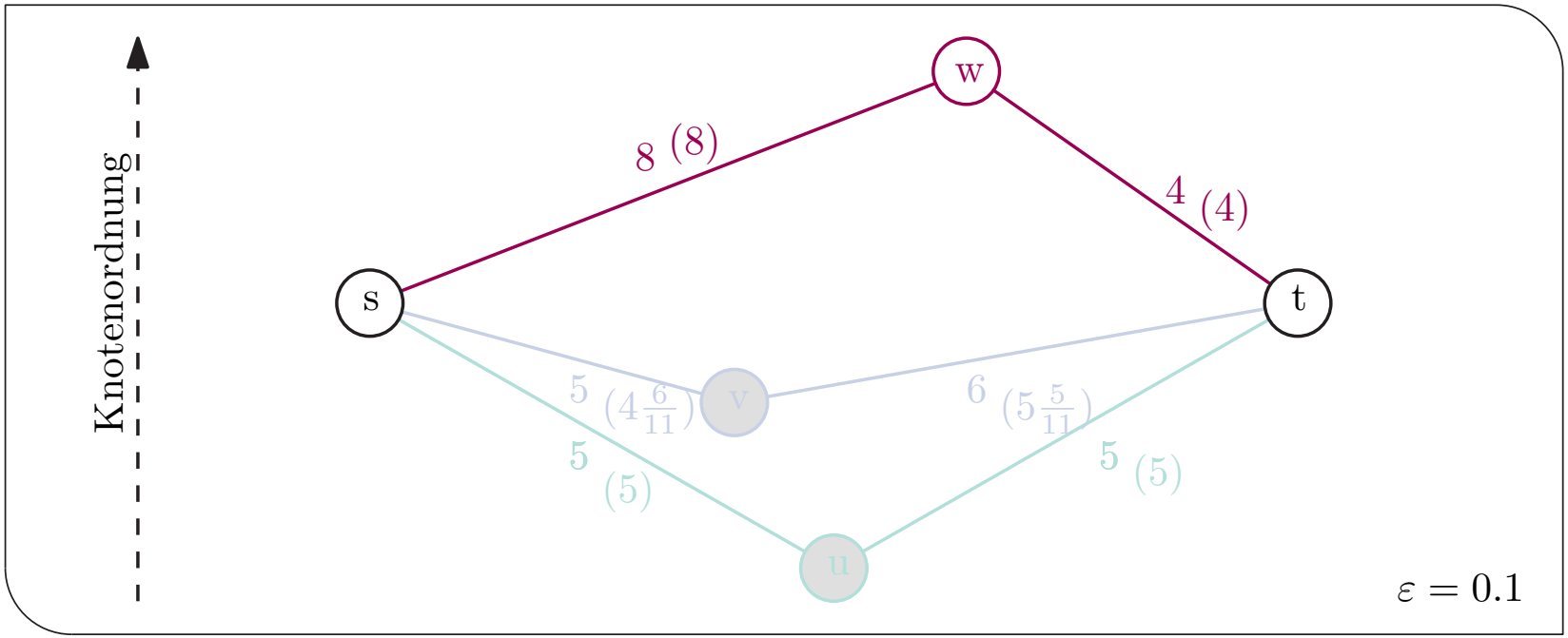
- speichere zusätzliches Gewicht $\tilde{c}$ (zu Beginn: $\tilde{c} = c$)
- erzeuge *Shortcut* falls $c(\text{Umweg}) > (1+\varepsilon)\tilde{c}(\text{Shortcut})$
- sonst, ändere $\tilde{c}(x,y) = \min\left\{\tilde{c}(x,y), \tilde{c}(\text{Shortcut})\frac{c(x,y)}{c(\text{Umweg})}\right\}$ auf allen Kanten (x,y) des *Umwegs*, der den *Shortcut* verhinderte.



approximierte Contraction Hierarchies

Details

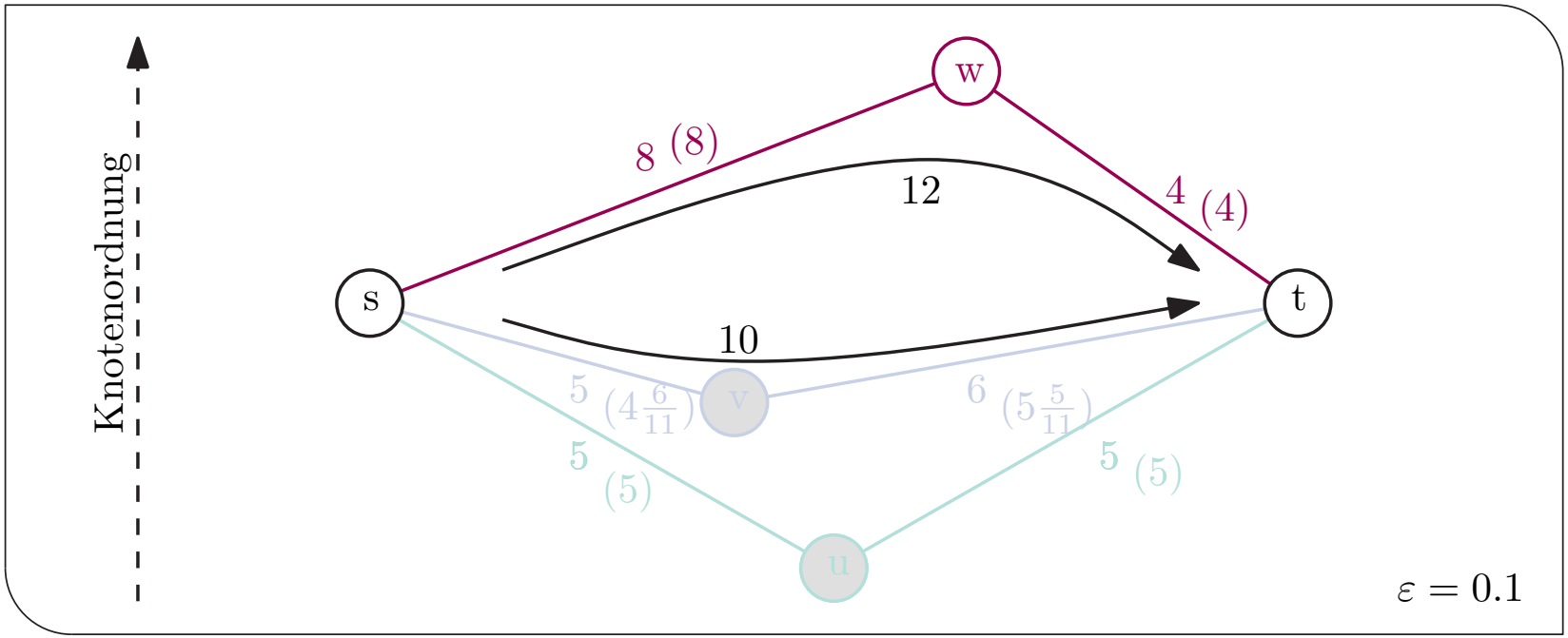
- speichere zusätzliches Gewicht $\tilde{c}$ (zu Beginn: $\tilde{c} = c$)
- erzeuge *Shortcut* falls $c(\text{Umweg}) > (1+\varepsilon)\tilde{c}(\text{Shortcut})$
- sonst, ändere $\tilde{c}(x,y) = \min\left\{\tilde{c}(x,y), \tilde{c}(\text{Shortcut})\frac{c(x,y)}{c(\text{Umweg})}\right\}$ auf allen Kanten (x,y) des *Umwegs*, der den *Shortcut* verhinderte.



approximierte Contraction Hierarchies

Details

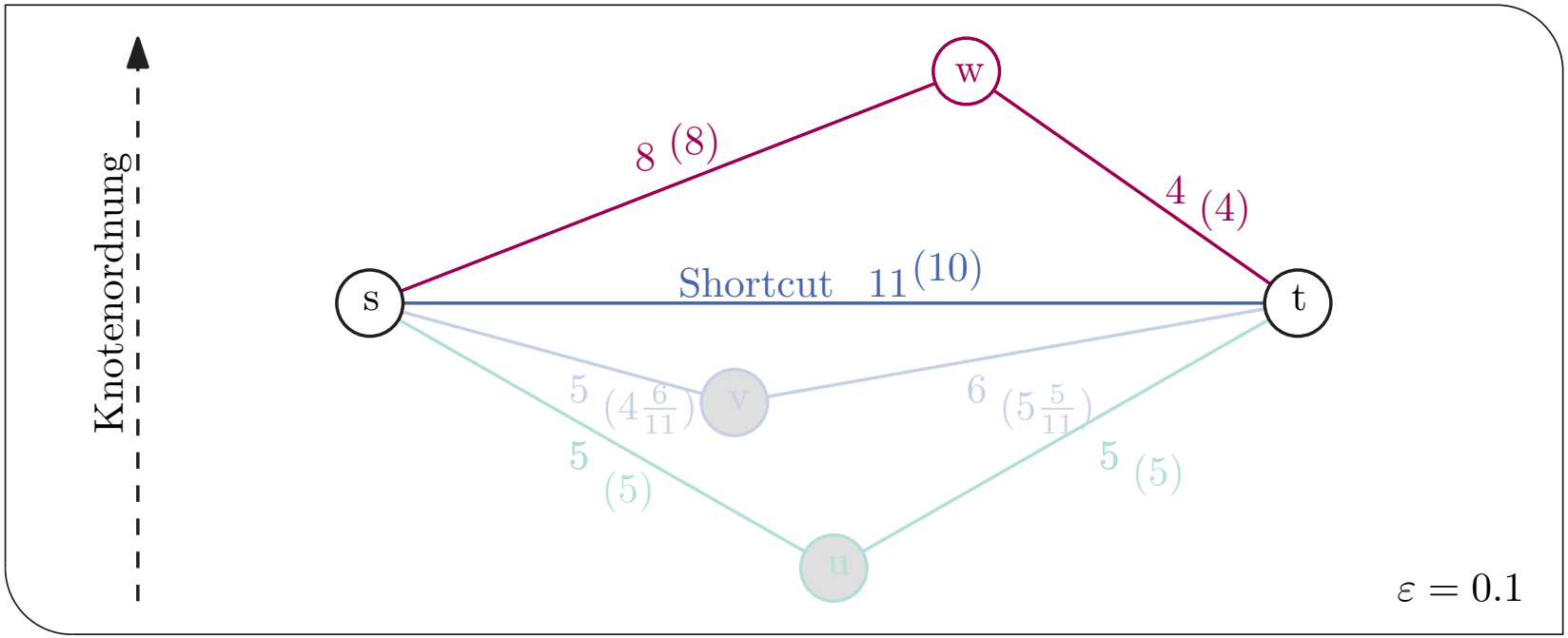
- speichere zusätzliches Gewicht $\tilde{c}$ (zu Beginn: $\tilde{c} = c$)
- erzeuge *Shortcut* falls $c(\text{Umweg}) > (1+\varepsilon)\tilde{c}(\text{Shortcut})$
- sonst, ändere $\tilde{c}(x,y) = \min\left\{\tilde{c}(x,y), \tilde{c}(\text{Shortcut})\frac{c(x,y)}{c(\text{Umweg})}\right\}$ auf allen Kanten (x,y) des *Umwegs*, der den *Shortcut* verhinderte.



approximierte Contraction Hierarchies

Details

- speichere zusätzliches Gewicht $\tilde{c}$ (zu Beginn: $\tilde{c} = c$)
- erzeuge *Shortcut* falls $c(\text{Umweg}) > (1+\varepsilon)\tilde{c}(\text{Shortcut})$
- sonst, ändere $\tilde{c}(x,y) = \min\left\{\tilde{c}(x,y), \tilde{c}(\text{Shortcut})\frac{c(x,y)}{c(\text{Umweg})}\right\}$ auf allen Kanten (x,y) des *Umwegs*, der den *Shortcut* verhinderte.

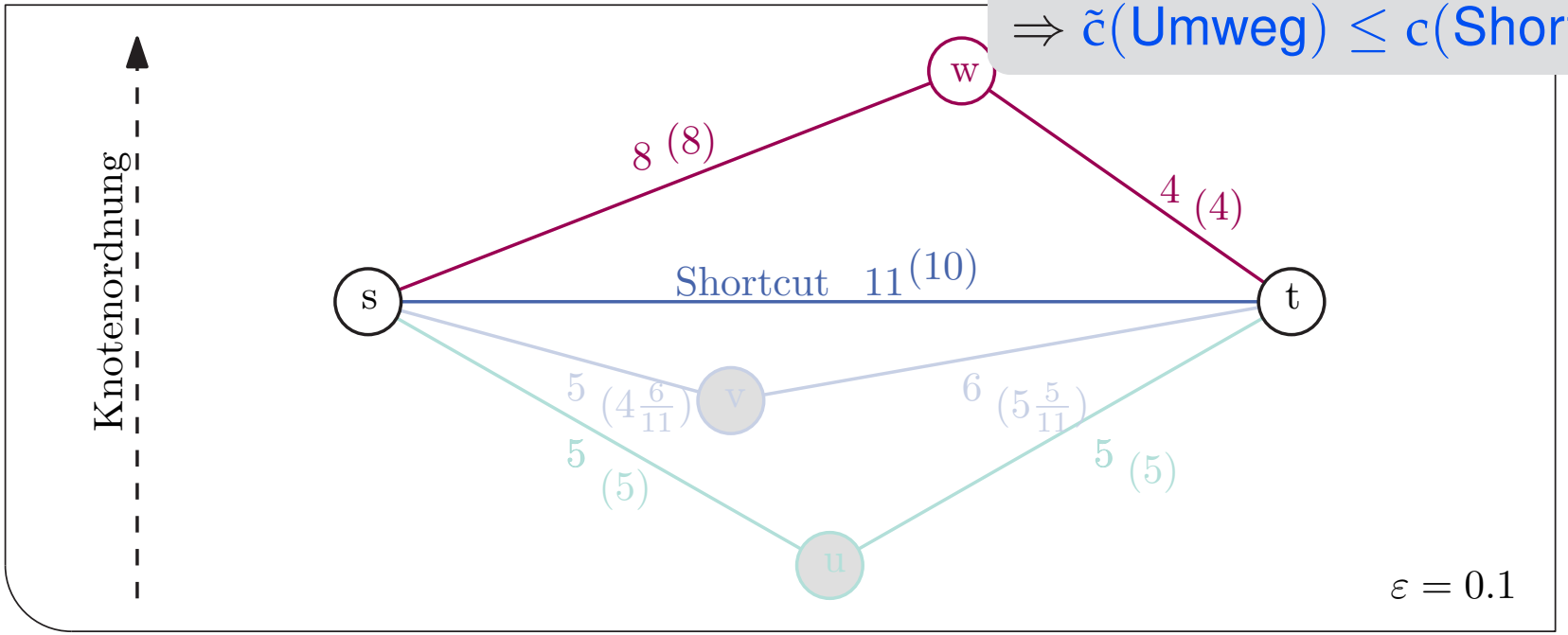


approximierte Contraction Hierarchies

Details

- speichere zusätzliches Gewicht $\tilde{c}$ (zu Beginn: $\tilde{c} = c$)
- erzeuge *Shortcut* falls $c(\text{Umweg}) > (1+\varepsilon)\tilde{c}(\text{Shortcut})$
- sonst, ändere $\tilde{c}(x,y) = \min\left\{\tilde{c}(x,y), \tilde{c}(\text{Shortcut})\frac{c(x,y)}{c(\text{Umweg})}\right\}$ auf allen Kanten (x,y) des *Umwegs*, der den *Shortcut* verhinderte.

$\Rightarrow \tilde{c}(\text{Umweg}) \leq c(\text{Shortcut})$

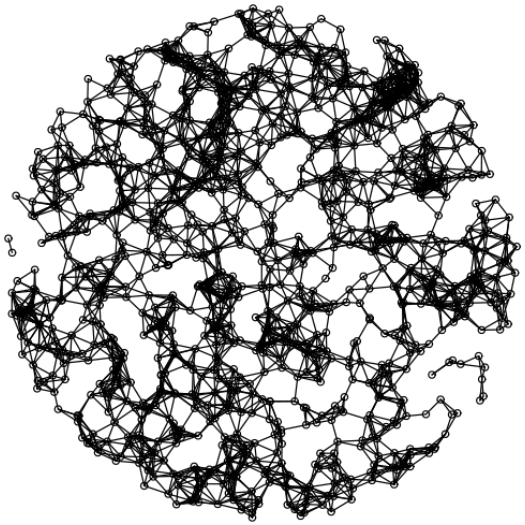


approximierte Contraction Hierarchies

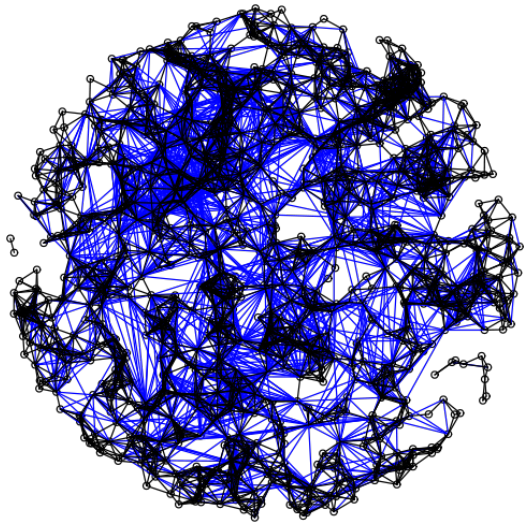
Details

Visualisierung

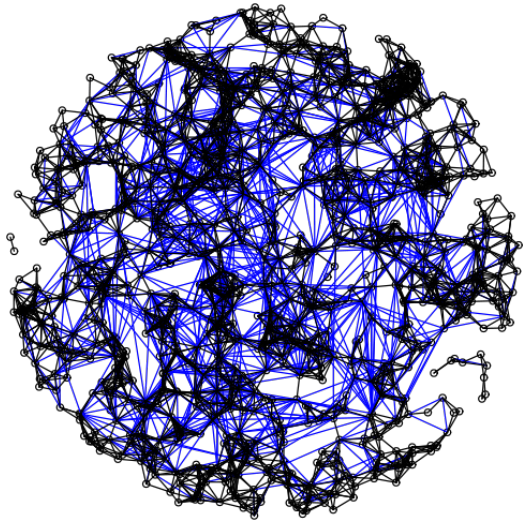
- approximierte CH brauchen weniger Shortcuts



Sensornetz



CH



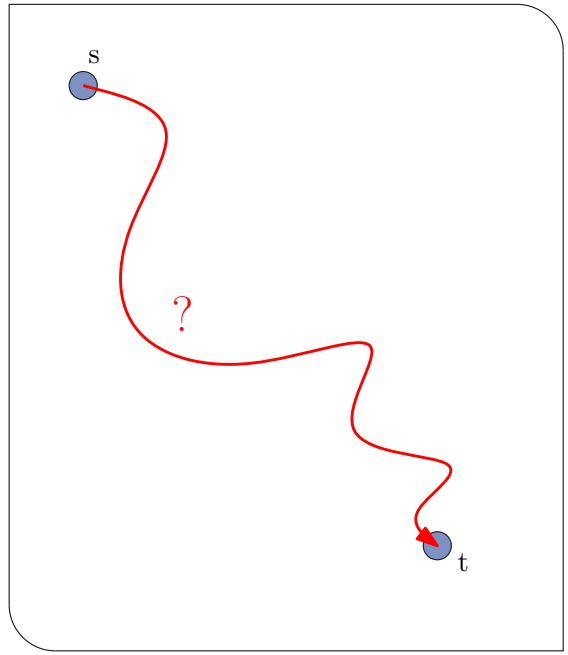
apxCH-10%

Kombination mit ALT

Überblick

Warum ALT?

- bester zielgerichteter Ansatz für diese Art von Graphen (Sensornetze)
- trivial erweiterbar zu approx. Verfahren (*weighted ALT*)



Ablauf

- zweistufiges Suchverfahren
- Zielrichtung auf obersten 5% der Hierarchie

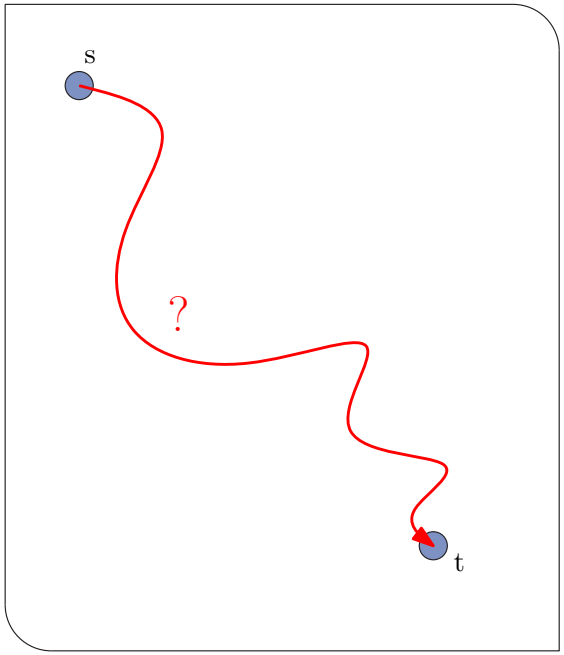
$\Rightarrow$ wenig zusätzlicher Speicherbedarf, dennoch große Beschleunigung

Kombination mit ALT

Überblick

Warum ALT?

- bester zielgerichteter Ansatz für diese Art von Graphen (Sensornetze)
- trivial erweiterbar zu approx. Verfahren (*weighted ALT*)



Ablauf

- zweistufiges Suchverfahren
- Zielrichtung auf obersten 5% der Hierarchie

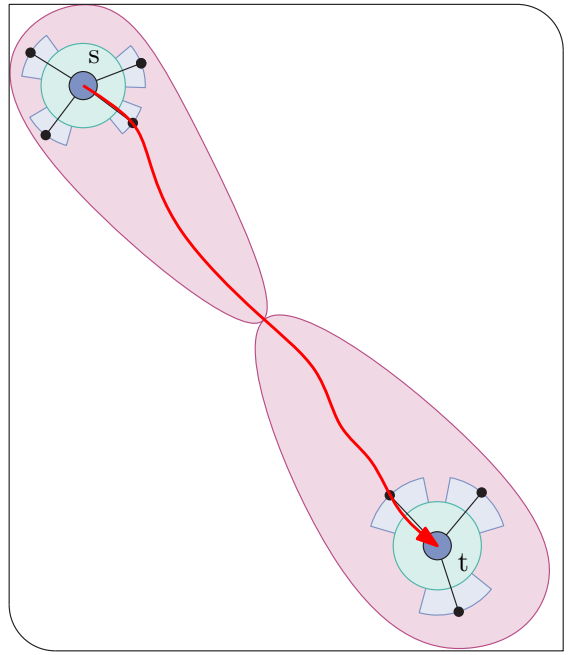
$\Rightarrow$ wenig zusätzlicher Speicherbedarf, dennoch große Beschleunigung

Kombination mit ALT

Überblick

Warum ALT?

- bester zielgerichteter Ansatz für diese Art von Graphen (Sensornetze)
- trivial erweiterbar zu approx. Verfahren (*weighted ALT*)



Ablauf

- zweistufiges Suchverfahren
- Zielrichtung auf obersten 5% der Hierarchie

$\Rightarrow$ wenig zusätzlicher Speicherbedarf, dennoch große Beschleunigung

Simulation

Ergebnisse

Sensornetzwerk, 1 000 000 Knoten, durchschnittlicher Knotengrad 10

	Vorbereitung		Suchanfrage		
	[s]	[B/n]	#besucht	[ms]	Fehler
bidir. Dijkstra	0	0	326 597	127.1	-
bidir. ALT-a64	194	512	3 173	2.8	-
unidir. A*	0	16	57 385	36.6	-
unidir. WA*-10%	0	16	1 234	1.0	1.25%
unidir. WA*-21%	0	16	724	0.7	2.87%
CH	1 887	0	2 969	4.0	-
apxCH-1%	993	-4	2 742	2.7	0.16%
apxCH-10%	474	-18	2 584	1.9	2.17%
apxCHALT-10%	489	7	215	0.3	2.16%
apxCHALT-10% W-10%	489	7	102	0.2	3.56%

Simulation

Ergebnisse

Sensornetzwerk, 1 000 000 Knoten, durchschnittlicher Knotengrad 10

	Vorbereitung		Suchanfrage		
	[s]	[B/n]	#besucht	[ms]	Fehler
bidir. Dijkstra	0	0	326 597	127.1	-
bidir. ALT-a64	194	512	3 173	2.8	-
unidir. A*	0	16	57 385	36.6	-
unidir. WA*-10%	0	16	1 234	1.0	1.25%
unidir. WA*-21%	0	16	724	0.7	2.87%
CH	1 887	0	2 969	4.0	-
apxCH-1%	993	-4	2 742	2.7	0.16%
apxCH-10%	474	-18	2 584	1.9	2.17%
apxCHALT-10%	489	7	215	0.3	2.16%
apxCHALT-10% W-10%	489	7	102	0.2	3.56%

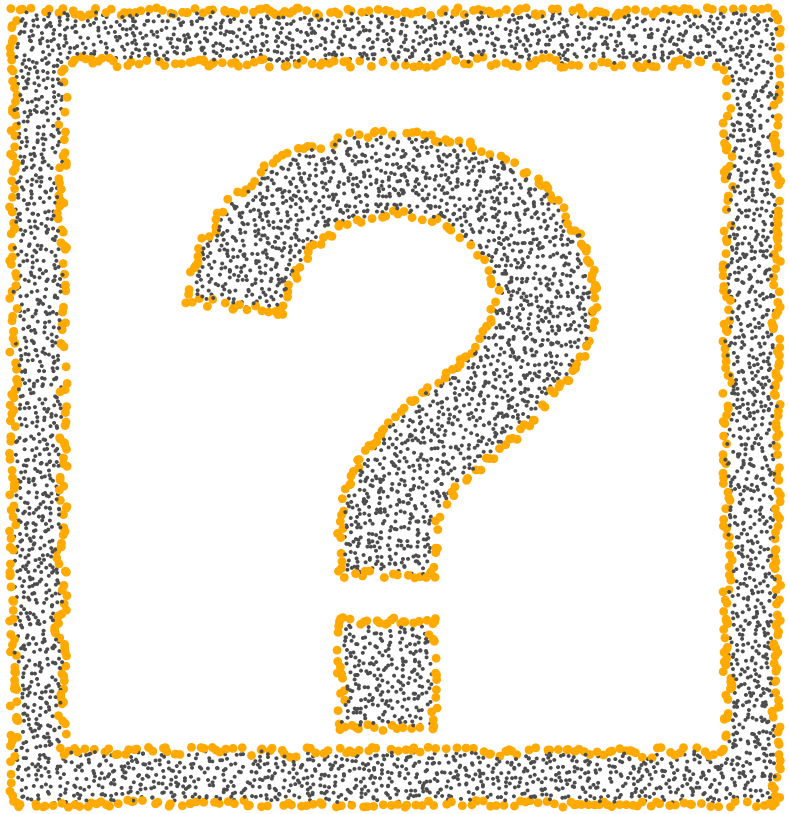
Simulation

Ergebnisse

Sensornetzwerk, 1 000 000 Knoten, durchschnittlicher Knotengrad 10

	Vorbereitung		Suchanfrage		
	[s]	[B/n]	#besucht	[ms]	Fehler
bidir. Dijkstra	0	0	326 597	127.1	-
bidir. ALT-a64	194	512	3 173	2.8	-
unidir. A*	0	16	57 385	36.6	-
unidir. WA*-10%	0	16	1 234	1.0	1.25%
unidir. WA*-21%	0	16	724	0.7	2.87%
CH	1 887	0	2 969	4.0	-
apxCH-1%	993	-4	2 742	2.7	0.16%
apxCH-10%	474	-18	2 584	1.9	2.17%
apxCHALT-10%	489	7	215	0.3	2.16%
apxCHALT-10% W-10%	489	7	102	0.2	3.56%

Danke für Ihre Aufmerksamkeit!



Zeit für Fragen

- III -

Backupfolien

Ergebnisse, Bilder

Ergebnisse

Simulationsaufbau

Verwendete Netzwerkstrukuren

- 5 Lochmuster, je 100 Knotenverteilungen generiert
- *Knotenverteilung*: auf pertubiertem Gitter / zufällig
- *Verbindungsmodell*: Unit Disk Graph / Quasi Unit Disk Graph

Verwendete Lochmuster

Ergebnisse

Simulationsaufbau

Verwendete Netzwerkstrukuren

- 5 Lochmuster, je 100 Knotenverteilungen generiert
- *Knotenverteilung*: auf pertubiertem Gitter / zufällig
- *Verbindungsmodell*: Unit Disk Graph / Quasi Unit Disk Graph

Bemerkungen

- eine der ersten quantitativen Analysen
- verwendet einfache Variante von MDS-BR mit $\alpha_{min} = 0.5\pi$, $r_{min} = 5$

Ergebnisse

Simulationsaufbau

Verwendete Netzwerkstrukuren

- 5 Lochmuster, je 100 Knotenverteilungen generiert
- *Knotenverteilung*: auf pertubiertem Gitter / zufällig
- *Verbindungsmodell*: Unit Disk Graph / Quasi Unit Disk Graph

Bemerkungen

- eine der ersten quantitativen Analysen
- verwendet einfache Variante von MDS-BR mit $\alpha_{min} = 0.5\pi$, $r_{min} = 5$

Ergebnisse

Simulationsaufbau

Verwendete Netzwerkstrukuren

- 5 Lochmuster, je 100 Knotenverteilungen generiert
- *Knotenverteilung*: auf pertubiertem Gitter / zufällig
- *Verbindungsmodell*: Unit Disk Graph / Quasi Unit Disk Graph

Bemerkungen

- eine der ersten quantitativen Analysen
- verwendet einfache Variante von MDS-BR mit $\alpha_{min} = 0.5\pi$, $r_{min} = 5$

Ergebnisse

Vergleiche

Vergleich verschiedener Verfahren

- für alle Grapharten gute Ergebnisse ohne spezielle Optimierungen
- besser als bestehende Verfahren

Fehlklassifikation von Randknoten und inneren Knoten in %:

	12	15-rnd	12-qudg
MDS-BR	3.0 / 0.3	5.2 / 12.2	5.6 / 4.6
Funke05	6.3 / 3.5	15.5 / 16.1	7.9 / 11.8
Funke06	13.8 / 1.4	45.8 / 7.9	15.8 / 12.2
Kroeller04	14.2 / 3.5	26.2 / 13.0	17.2 / 8.0

Ergebnisse

Vergleiche

Vergleich verschiedener Verfahren

- für alle Grapharten gute Ergebnisse ohne spezielle Optimierungen
- besser als bestehende Verfahren

Fehlklassifikation von Randknoten und inneren Knoten in %:

	12	15-rnd	12-qudg
MDS-BR	3.0 / 0.3	5.2 / 12.2	5.6 / 4.6
Funke05	6.3 / 3.5	15.5 / 16.1	7.9 / 11.8
Funke06	13.8 / 1.4	45.8 / 7.9	15.8 / 12.2
Kroeller04	14.2 / 3.5	26.2 / 13.0	17.2 / 8.0

Ergebnisse

Vergleiche

Vergleich verschiedener MDS Varianten

- *MDS3*: verwendet 3-hop Nachbarschaft
- *SSMDS*: verwendet Signalstärke
- *OPT*: verwendet wahre Winkel

Fehlklassifikation von Randknoten und inneren Knoten in %:

	15	15-rnd	15-qudg
MDS	3.5 / 0.1	5.1 / 12.4	6.8 / 0.6
MDS3	1.8 / 0.2	3.7 / 12.1	3.5 / 0.6
SSMDS	0.6 / 0.2	1.9 / 14.9	1.3 / 1.0
OPT	0.4 / 0.1	6.5 / 10.5	0.8 / 0.4

Ergebnisse

Vergleiche

Vergleich verschiedener MDS Varianten

- *MDS3*: verwendet 3-hop Nachbarschaft
- *SSMDS*: verwendet Signalstärke
- *OPT*: verwendet wahre Winkel

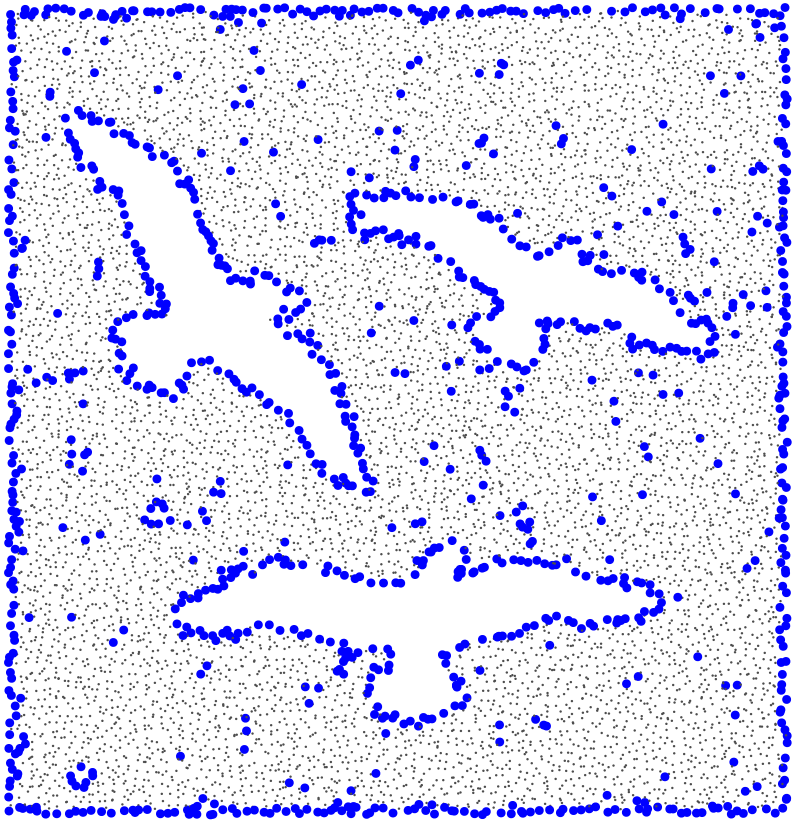
Fehlklassifikation von Randknoten und inneren Knoten in %:

	15	15-rnd	15-qudg
MDS	3.5 / 0.1	5.1 / 12.4	6.8 / 0.6
MDS3	1.8 / 0.2	3.7 / 12.1	3.5 / 0.6
SSMDS	0.6 / 0.2	1.9 / 14.9	1.3 / 1.0
OPT	0.4 / 0.1	6.5 / 10.5	0.8 / 0.4

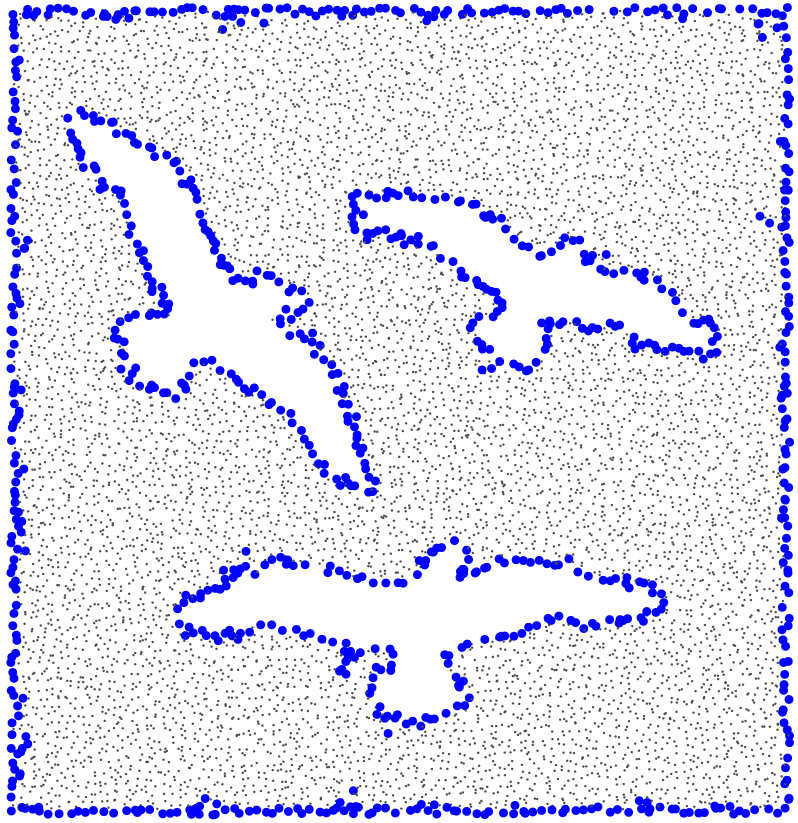
Ergebnisse

MDS-BR

ohne Nachoptimierung

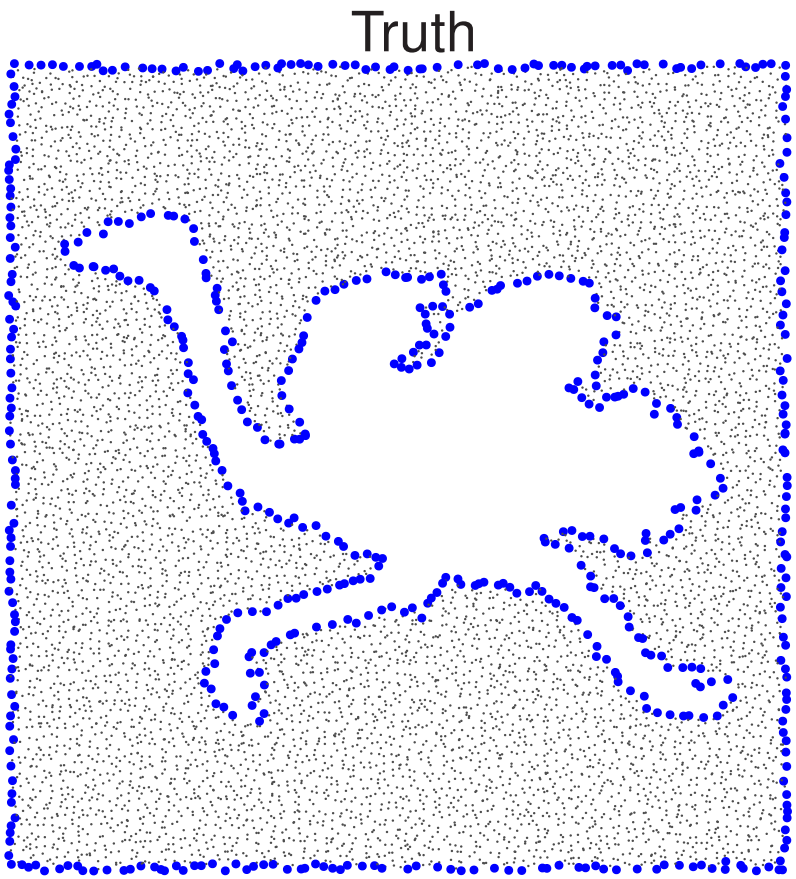


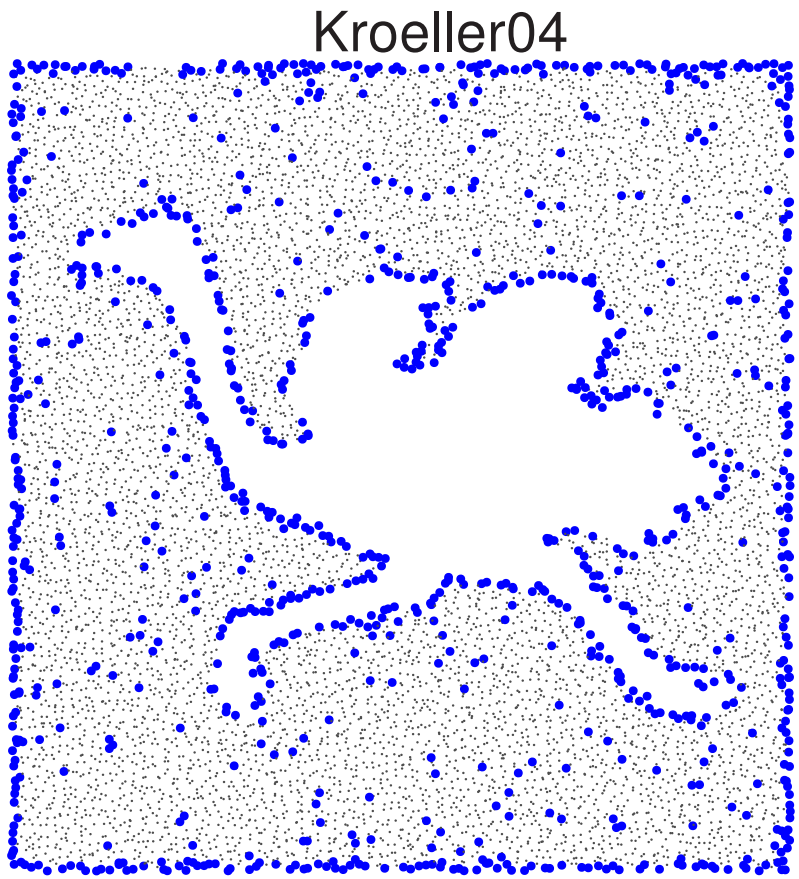
mit Nachoptimierung



Ergebnisse

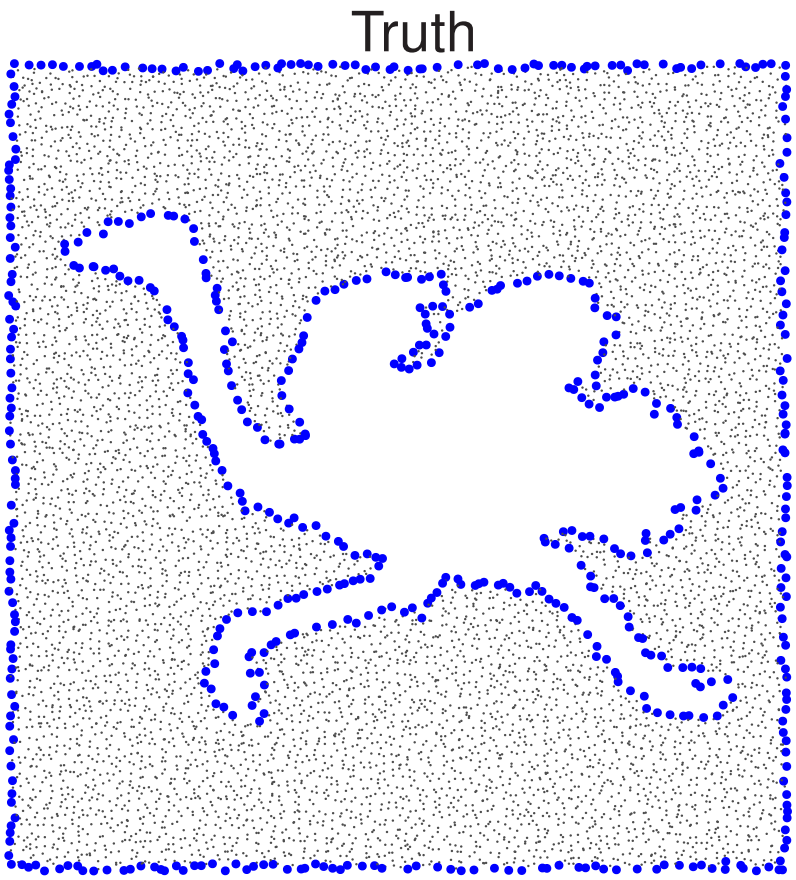
Vergleiche

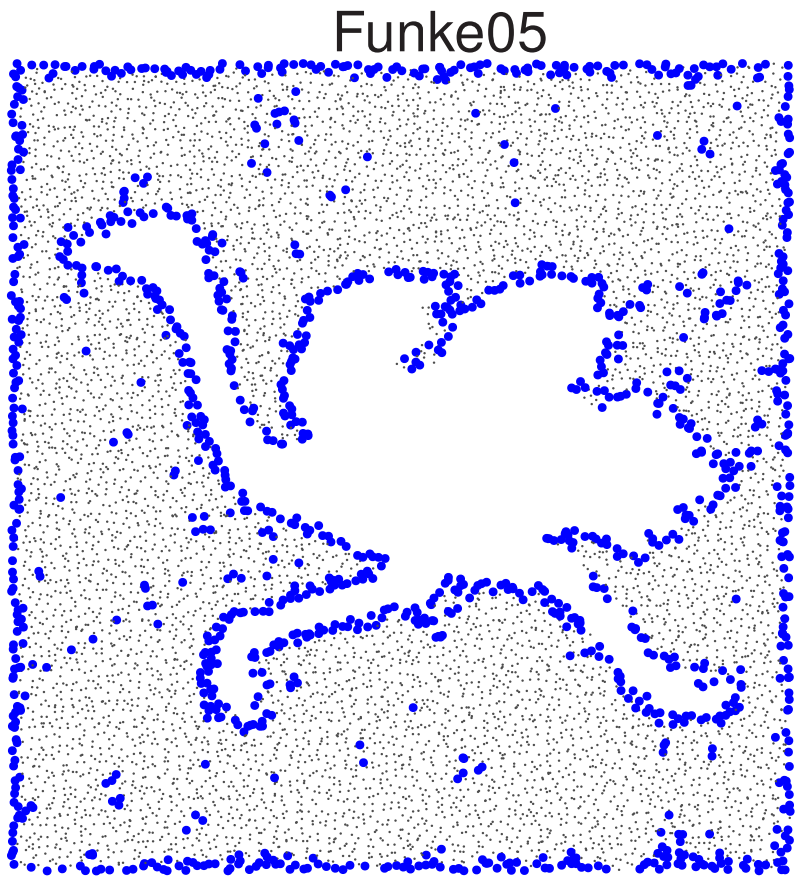




Ergebnisse

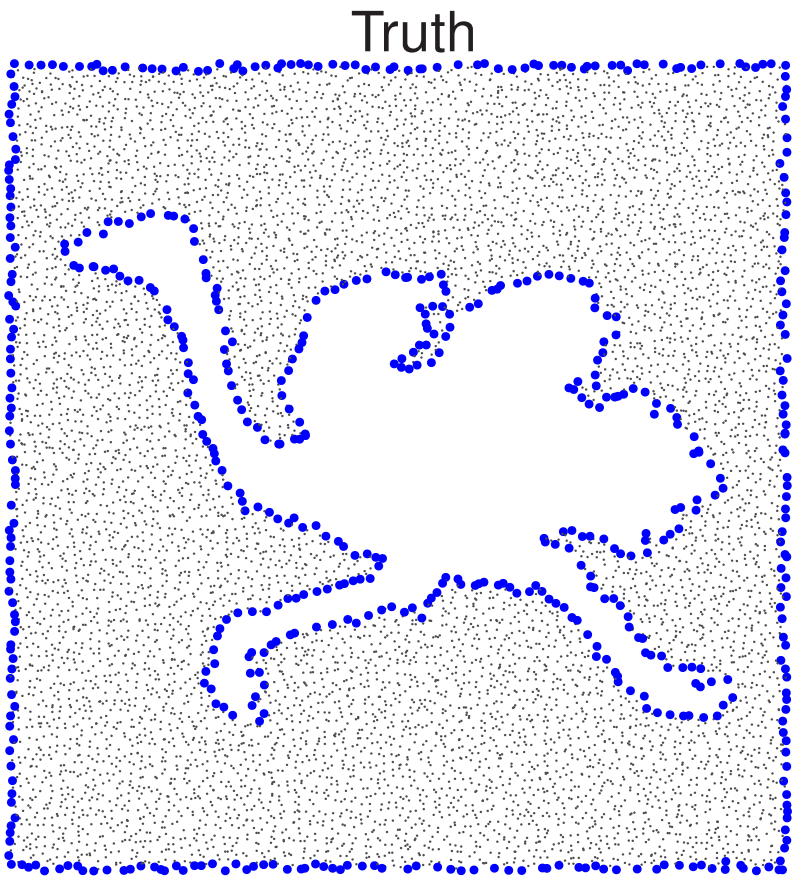
Vergleiche

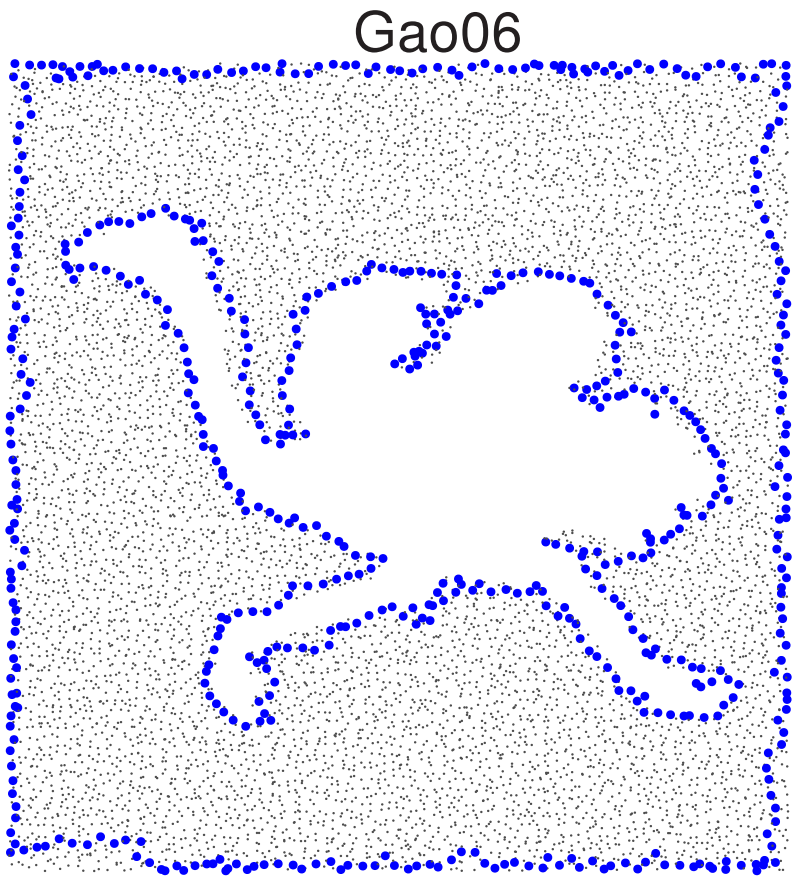




Ergebnisse

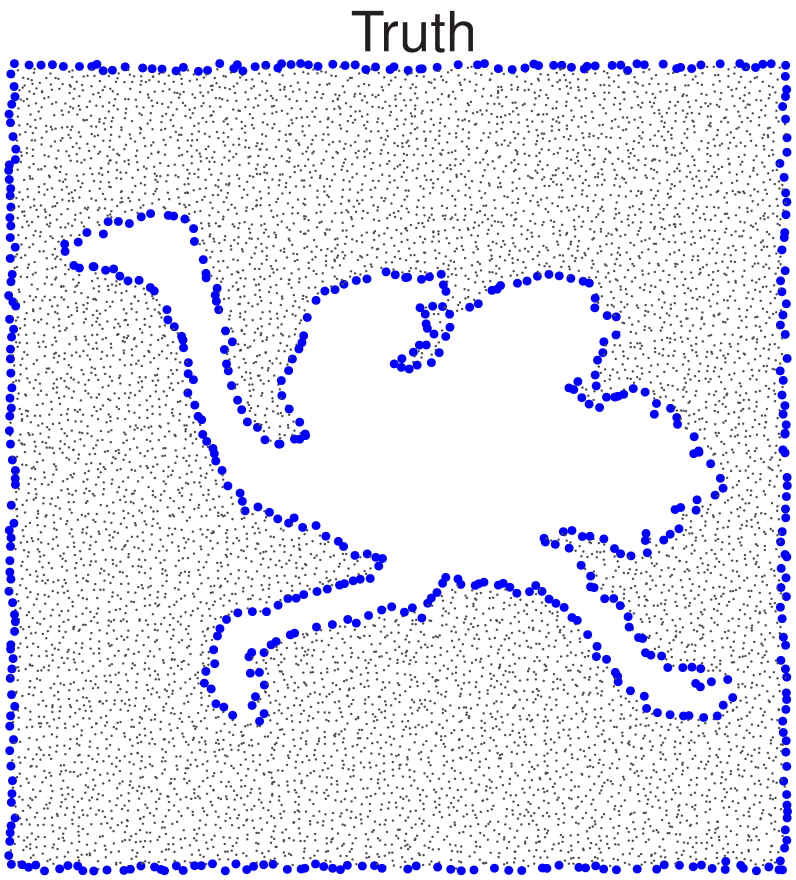
Vergleiche

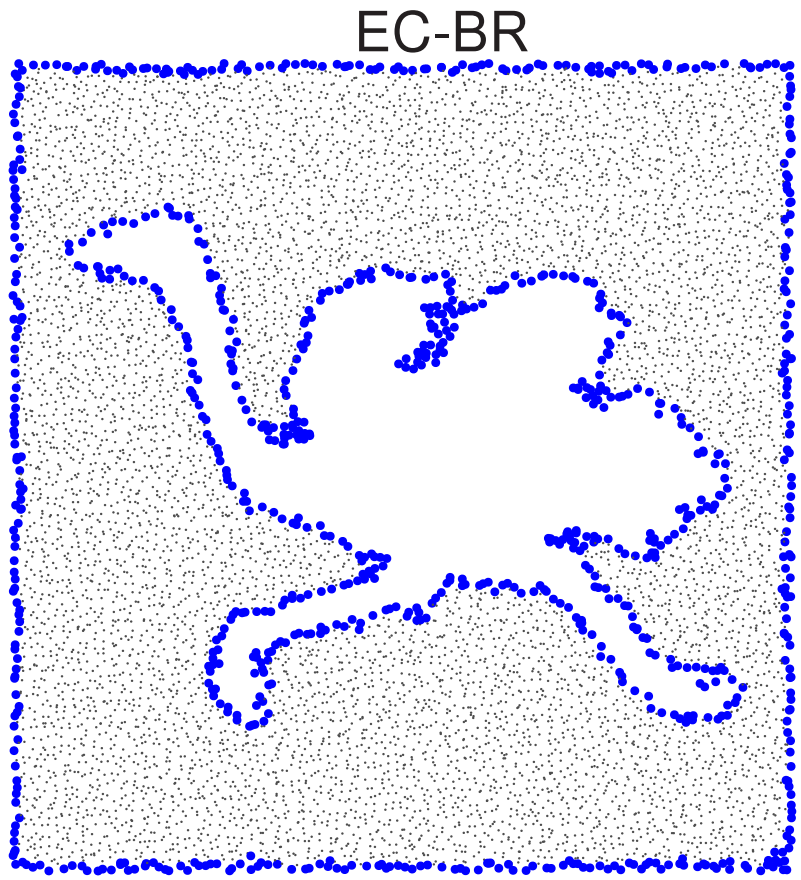




Ergebnisse

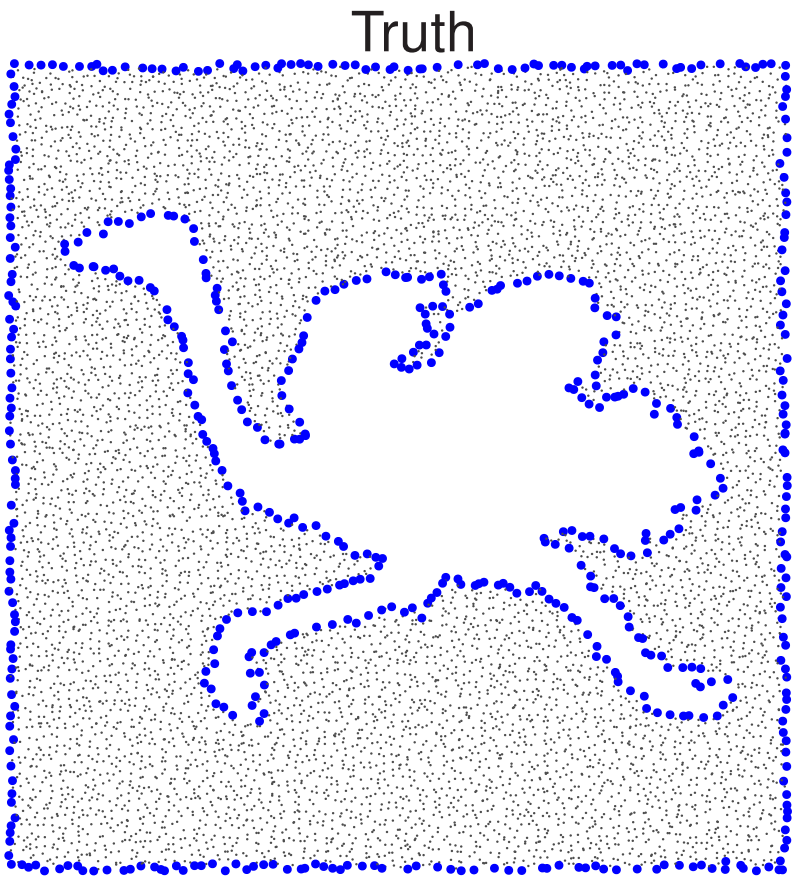
Vergleiche

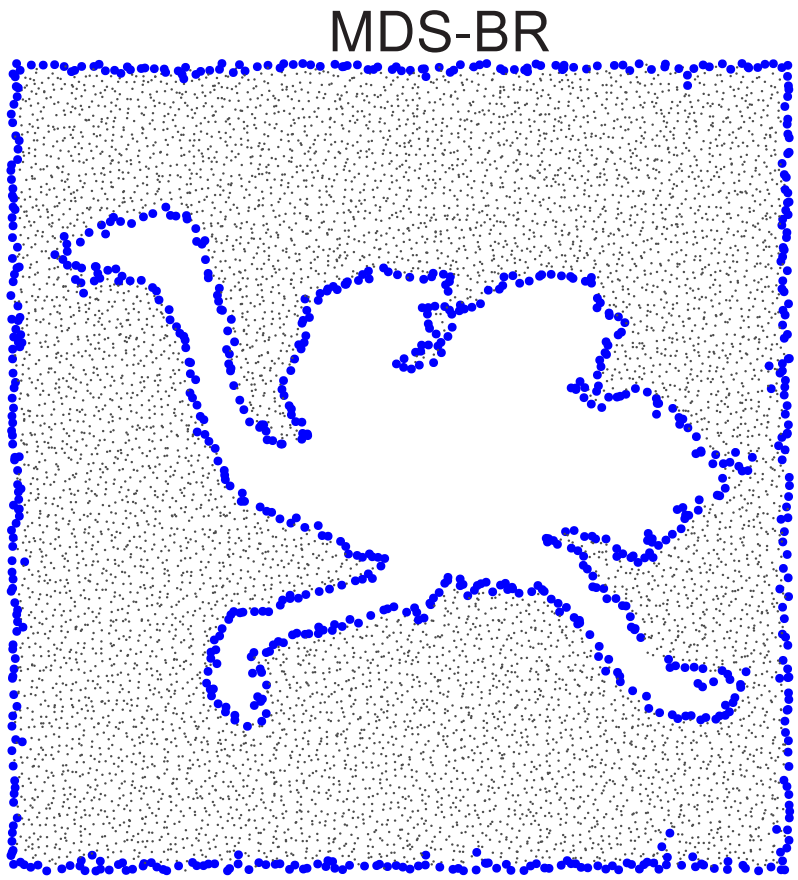




Ergebnisse

Vergleiche

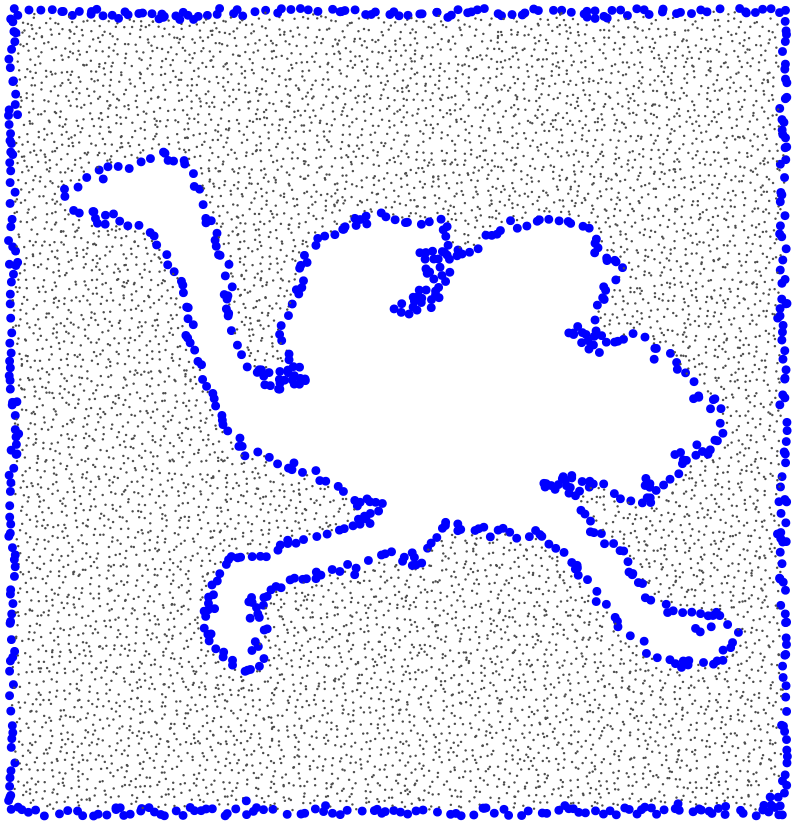




Ergebnisse

Vergleiche

EC-BR



MDS-BR