UNIVERSITÄT PASSAU

Fakultät für Mathematik und Informatik

Kreuzungen in Cluster-Level-Graphen

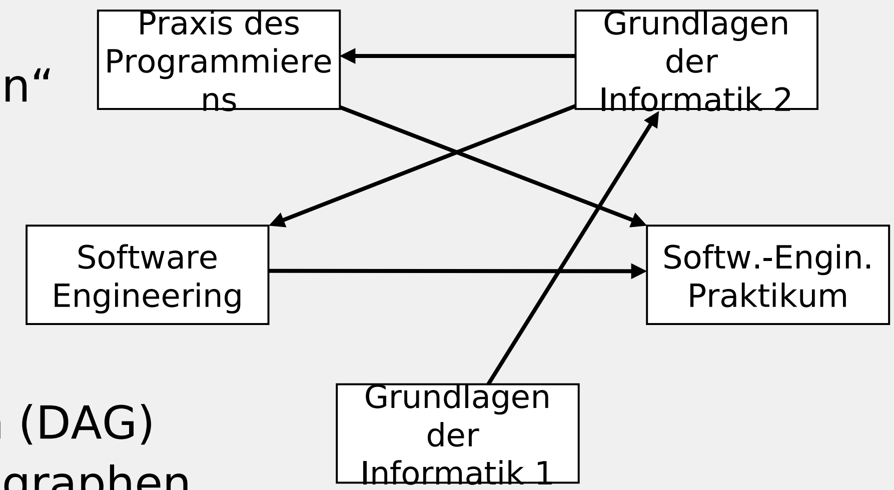
Michael Forster

forster@fmi.uni-passau.de

Lehrstuhl für Theoretische Informatik
Prof. Dr. Franz J. Brandenburg

Beispiel

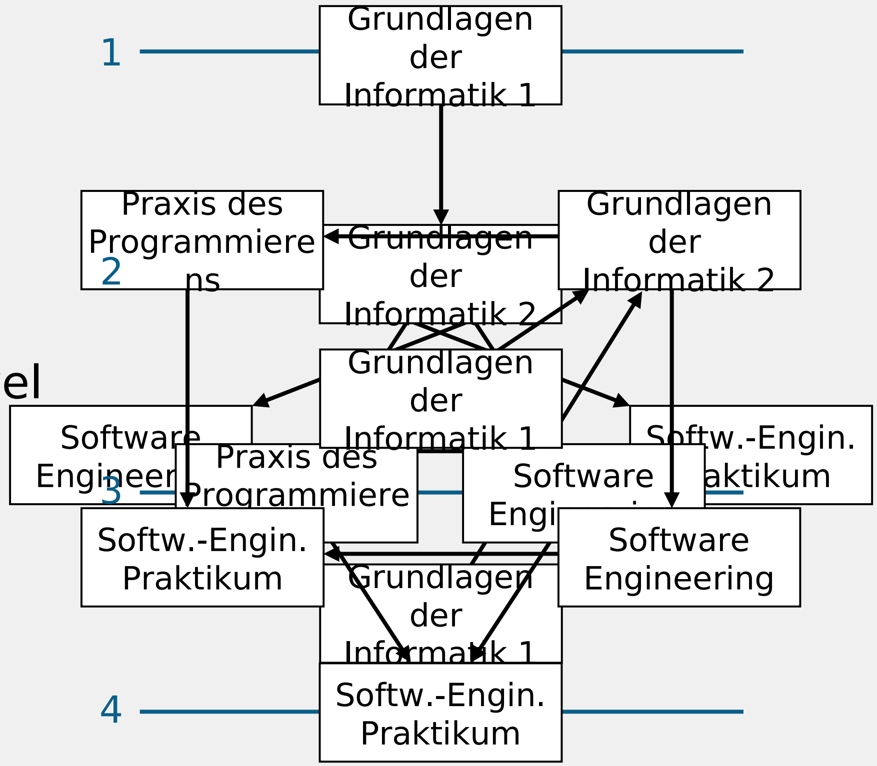
- Vorlesungsgraph
  - Informatik Grundstudium
  - Universität Passau
- Knoten
  - Vorlesungen
- Kanten
  - Inhaltliche Abhängigkeiten
  - „Vorlesung X soll vor Vorlesung Y gehört werden“
- Eigenschaften
  - Gerichtet
  - Azyklisch



→ Gerichteter azyklischer Graph (DAG)
  - Typisch für Abhängigkeitsgraphen
  - Führt beim Zeichnen zu Level-Graphen

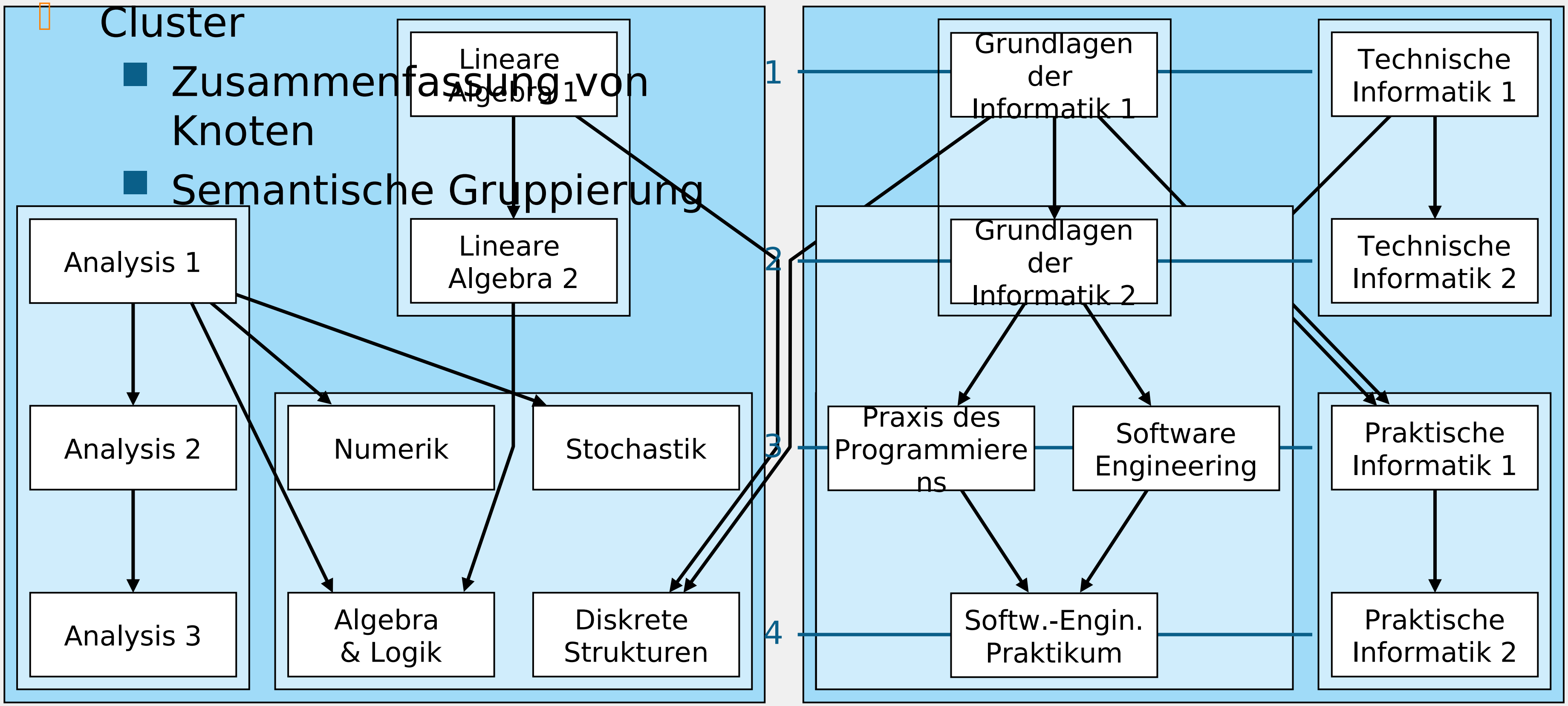
Level-Graphen

- Problemstellung
  - Gegeben: Gerichteter azyklischer Graph (DAG)
  - Gesucht: Übersichtliche Zeichnung
- Hauptziele
  - Wenig Kantenkreuzungen
  - Uniforme Kantenrichtung
- Standardalgorithmus
  - Sugiyama-Algorithmus
  - Methode: Einteilung in Level
- Level-Graph
  - Graph + Level-Enteilung
  - Zwischenergebnis des Sugiyama-Algorithmus
  - Manchmal: Level sind vorgegeben



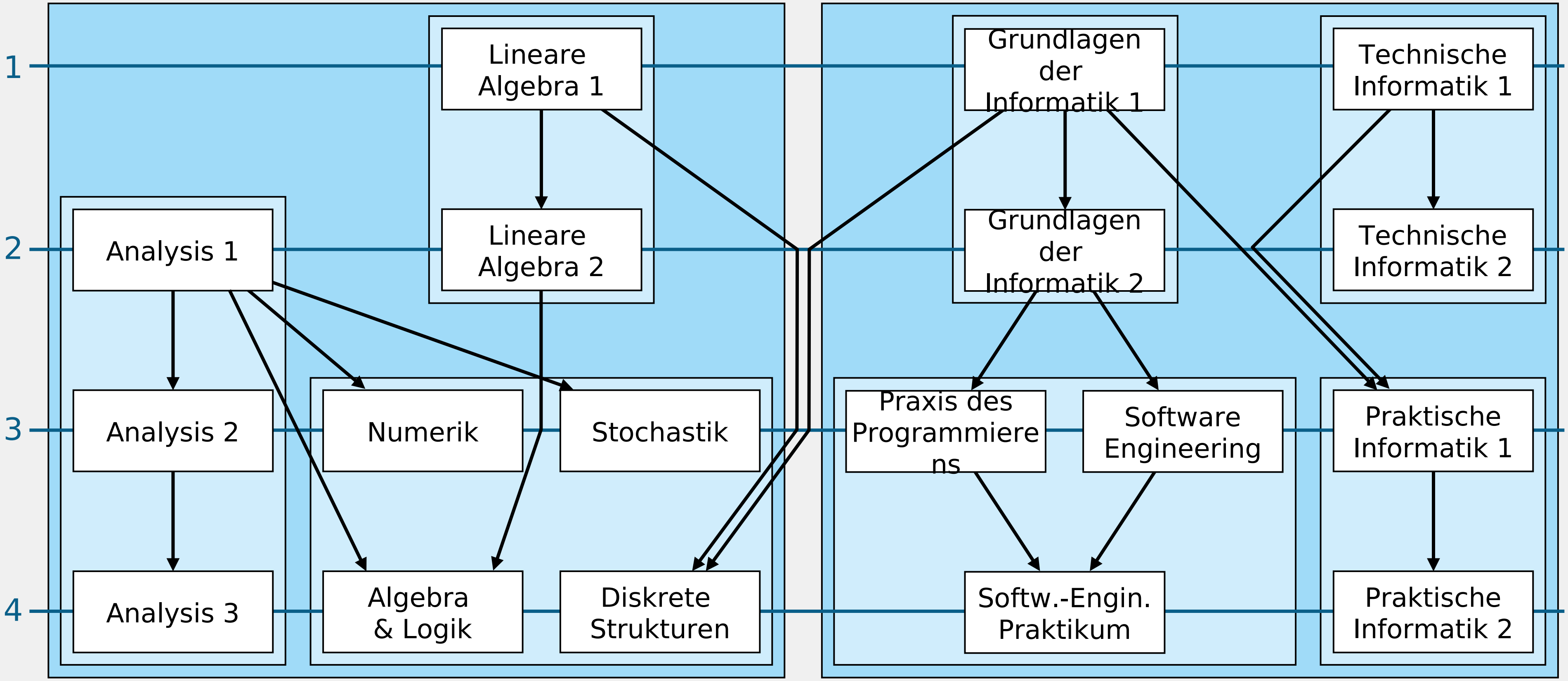
Cluster-Graphen

- Cluster-Graph
  - (Un)gerichteter Graph
  - Menge von Clustern
- Cluster
  - Zusammenfassung von Knoten
  - Semantische Gruppierung
- Eigenschaften
  - Verboten: Überlappung
  - Erlaubt: Schachtelung



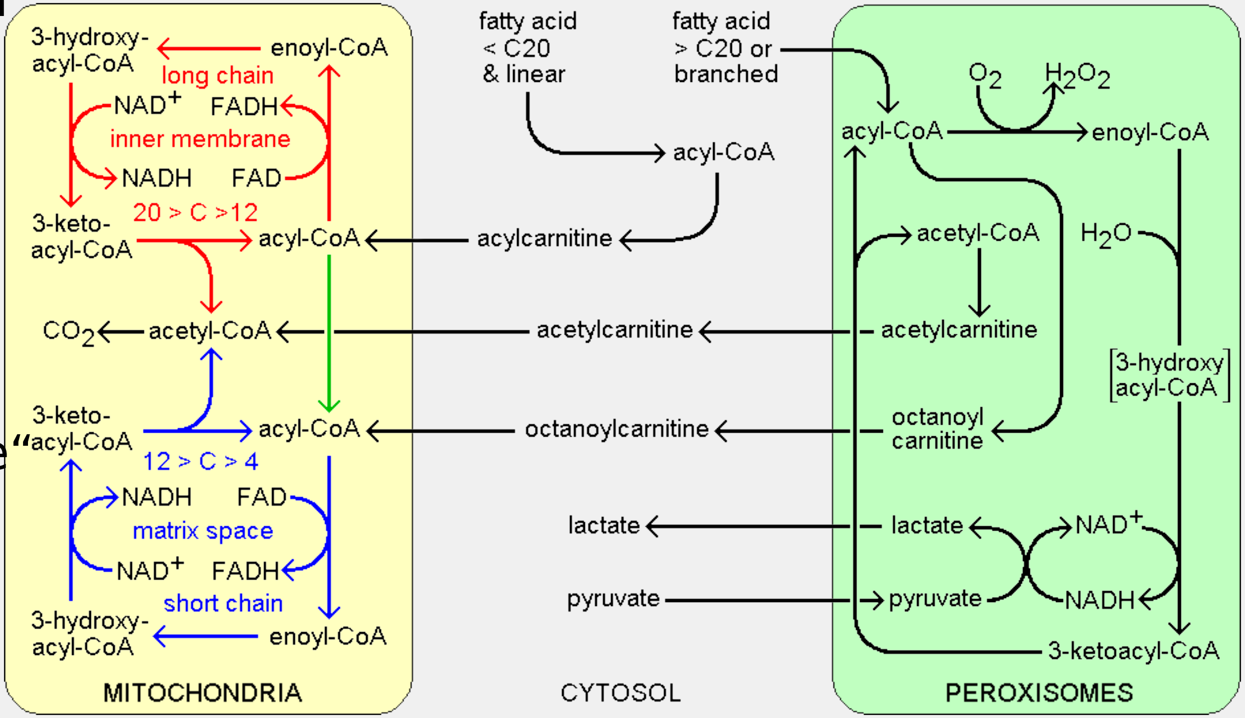
Problemstellung

- Gegeben: Cluster-Level-Graph
  - Graph + Level + Cluster
  - Gerichtet oder ungerichtet
- Ziel
  - Übersichtliche Zeichnung
  - Insbesondere: Wenig Kreuzungen



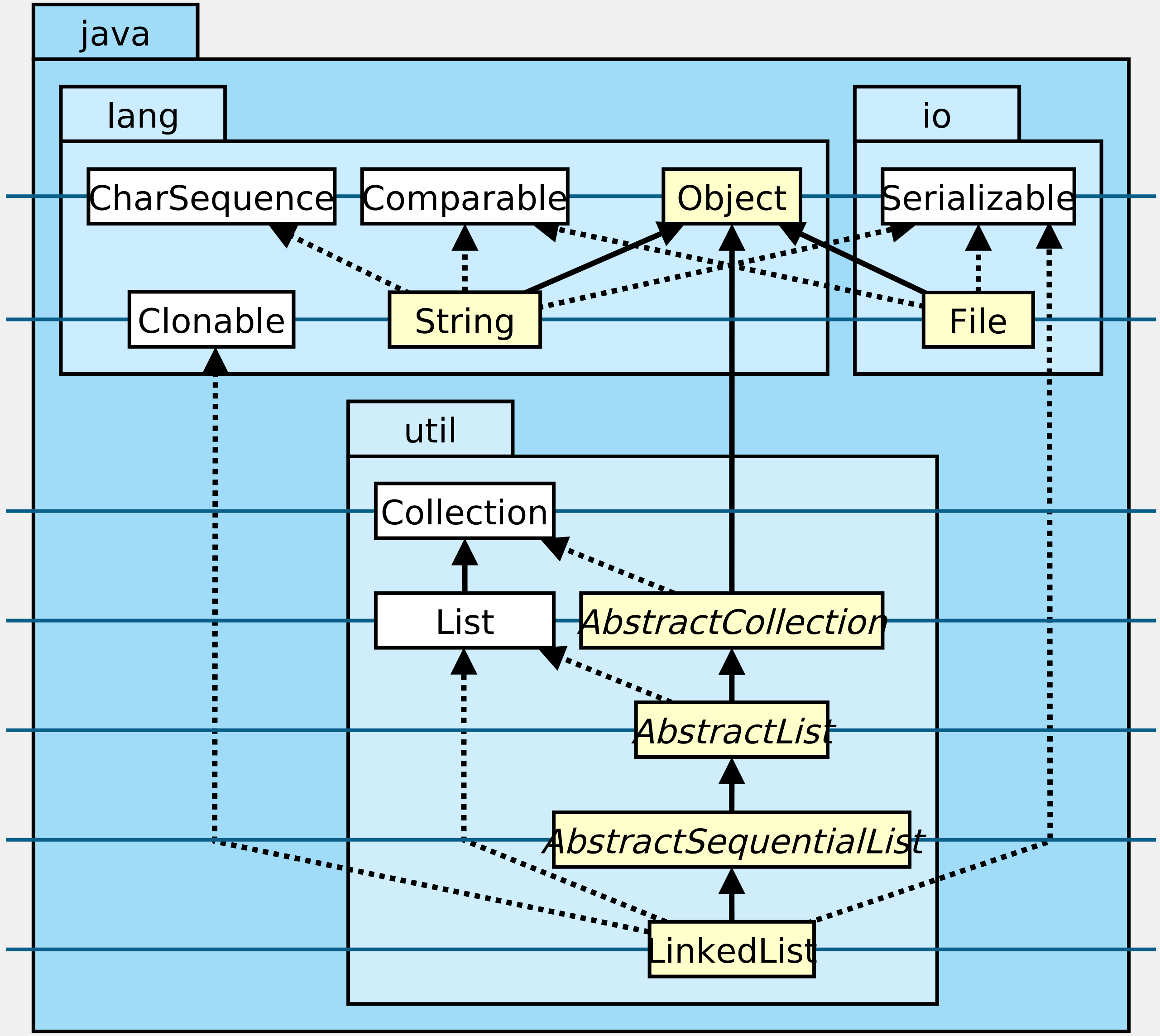
Anwendung: Biochemical Pathways

- Biochemische Reaktionsnetze
- Knoten
  - (Co-)Substanzen
  - Enzyme
- Gerichtete Kanten
  - Reaktionen
  - Regulation
- Cluster
  - „Kompartimente“
  - Abgegrenzte Zellbereiche
  - Ineinander geschachtelt



Anwendung: UML Klassen-Diagramme

- Knoten
  - Klassen
  - Interfaces
- Gerichtete Kanten
  - Vererbung
  - Assoziation
- Cluster
  - Packages

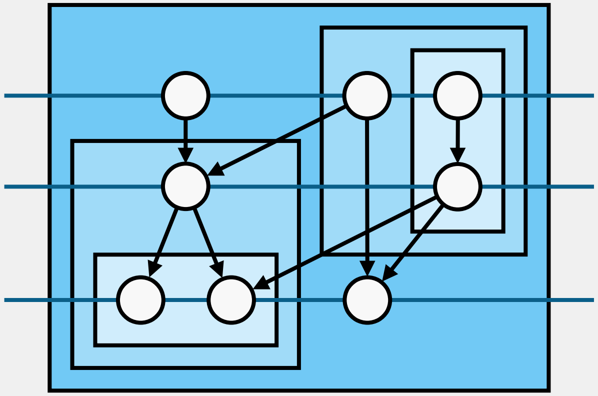


Überblick

- Einleitung
- Grundlagen
  - Notation
  - Charakterisierung von Kreuzungen
- Kreuzungsreduzierung
  - In Cluster-Level-Graphen
  - In Level-Graphen mit Constraints
- Cluster-Level-Planarität
  - Frage: Gibt es eine Zeichnung ohne Kreuzungen?
- Ausblick

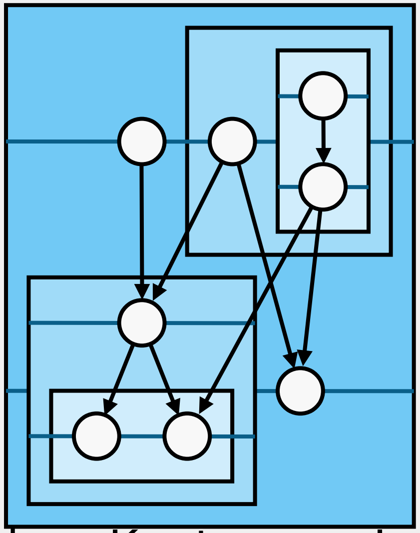
Level: Global oder Lokal?

- Globale Level
  - Eine Menge von Level für alle Knoten
  - Cluster können über mehrere Level gehen



- Kompaktere Zeichnungen
- Algorithmus: Sander (1996)

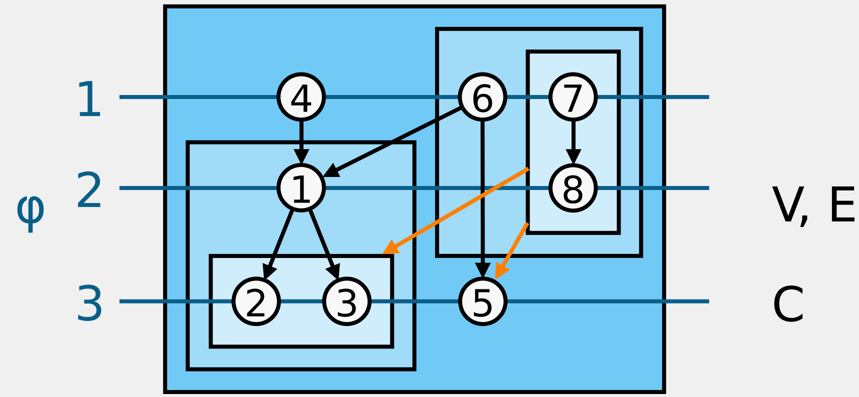
- Lokale Level
  - Jeder Cluster enthält seine eigenen Level
  - Cluster dürfen nicht über mehrere Level gehen



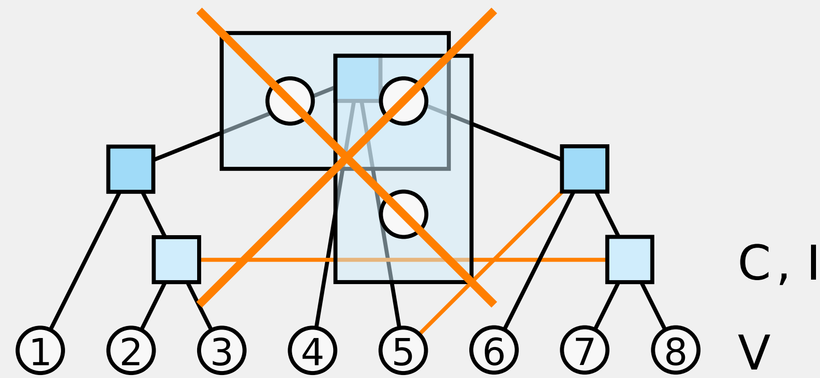
- Weniger Knoten pro Level
- Algorithmus: Sugiyama/Misue (1991)

Cluster-Level-Graphen

- Cluster-Level-Graph: G = (V, E, C, I, φ)
  - Gerichteter Graph
    - Knoten: V
    - Kanten: $E \subseteq V \times V$
  - Cluster-Baum: $T = (V \cup C, I)$
    - Blätter: Knoten V
    - Innere Knoten: Cluster C
    - Kanten: Inklusionsrelation I
  - Level-Einteilung: $\varphi: V \rightarrow \mathbb{N}$
    - Level eines Knoten: $\varphi(v)$
    - Knoten eines Levels: $V_i = \varphi^{-1}(i)$
    - Alle Kanten (u, v) zeigen nach unten: $\varphi(u) < \varphi(v)$

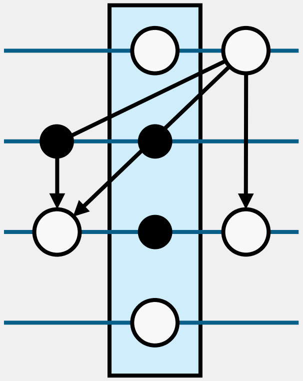


- Eigenschaften
  - Kanten verlaufen nur zwischen Knoten, nicht zwischen Clustern
  - Cluster können sich gegenseitig enthalten aber nicht überlappen



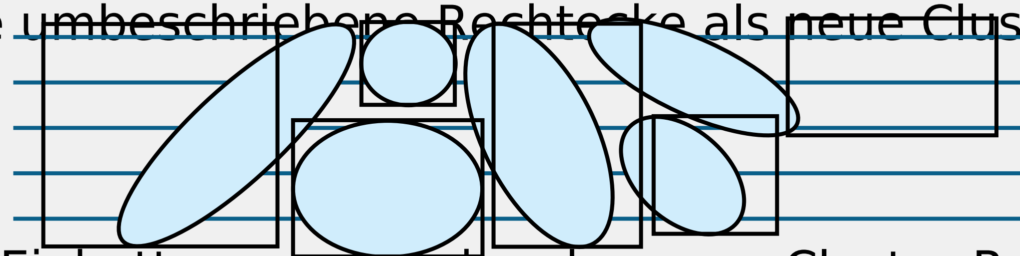
Einbettung

- Gesucht: Einbettung
  - Lineare Ordnung der Knoten innerhalb jedes Levels
- Ziel
  - Wenig Kreuzungen
- Problem
  - Kreuzungen sind nicht direkt aus der Einbettung ablesbar
- Definition: „Proper“
  - Keine langen Kanten
  - Keine leeren Level innerhalb eines Clusters
- Graphen können mit Dummy-Knoten proper gemacht werden
  - Bis zu $O(n^2)$ viele!



Cluster-Regionen

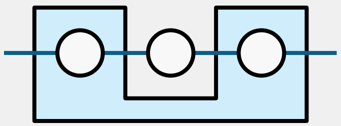
- Weiteres Ziel
  - „Einfache“ Cluster-Regionen
  - Hier: Konvexe Regionen
- Satz
  - Eine Einbettung kann mit konvexen Cluster-Regionen gezeichnet werden
    ⇒ Sie kann sogar mit rechteckigen Cluster-Regionen gezeichnet werden
- Beweisidee
  - Schiebe Cluster unter Beibehaltung der partiellen Ordnung horizontal auseinander ⇒ Einbettung ändert sich nicht
  - Benutze umbeschriebene Rechtecke als neue Cluster-Regionen



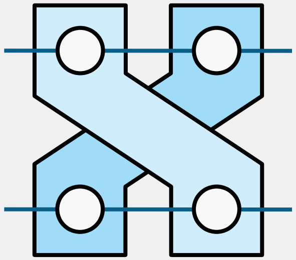
- Frage
  - Welche Einbettungen erlauben konvexe Cluster-Regionen?

Bedingungen für die Einbettung

- Konvexe Cluster-Regionen
  - Cluster/Level-Bedingung

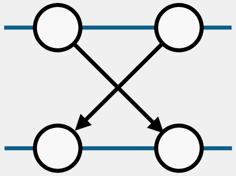


  - Cluster/Cluster-Bedingung

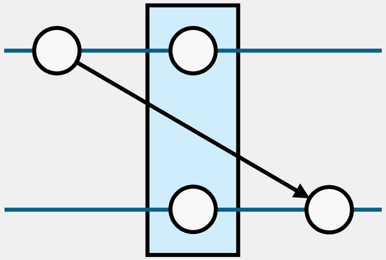


- Verboten
  - Vermeidung immer möglich
  - Folge: Cluster können als Rechtecke gezeichnet werden

- Kreuzungen
  - Kanten/Kanten-Bedingung



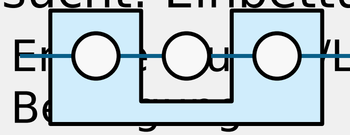
  - Kanten/Cluster-Bedingung

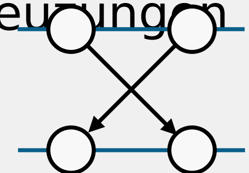


- Erlaubt aber unerwünscht
  - Vermeidung nicht immer möglich
  - Ziel: Minimierung

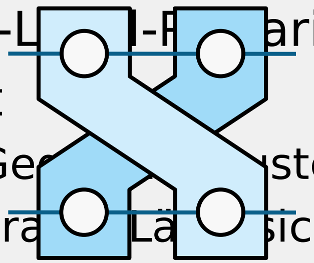
Problemstellungen

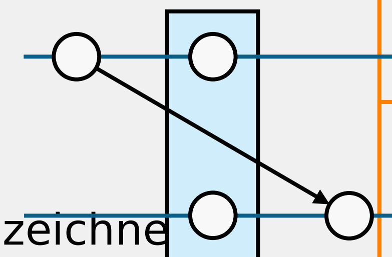
- Cluster-Level-Kreuzungsreduzierung
  - Gegeben: Cluster-Level-Graph G
  - Gesucht: Einbettung von G mit wenig Kreuzungen
    - Erfülle Cluster/Level- und Cluster/Cluster-Bedingungen
    - Minimiere Verletzungen der Kanten/Kanten- und Kanten/Cluster-Bedingungen

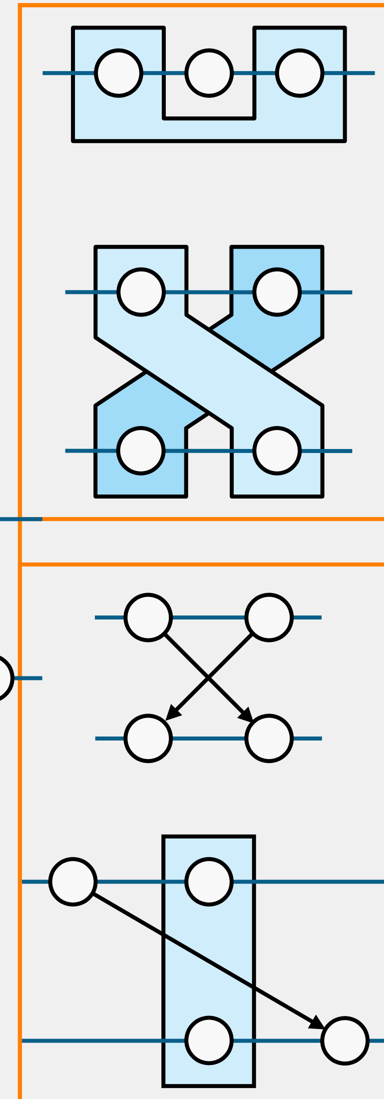




- Cluster-Level-Planarität
  - Test
    - Gegeben: Cluster-Level-Graph G
    - Frage: Lässt sich G ohne Kreuzungen zeichnen?
      - Erfülle alle Bedingungen
  - Einbettung
    - Gegeben: Planarer Cluster-Level-Graph G
    - Gesucht: Planare Einbettung von G
      - Erfülle alle Bedingungen







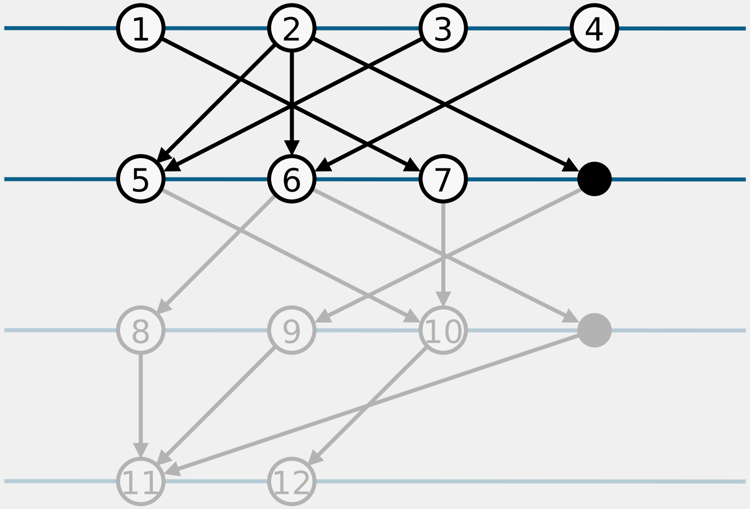
Kreuzungsreduzierung

Wie hält man die Anzahl an Kreuzungen gering?

[GD 2002]

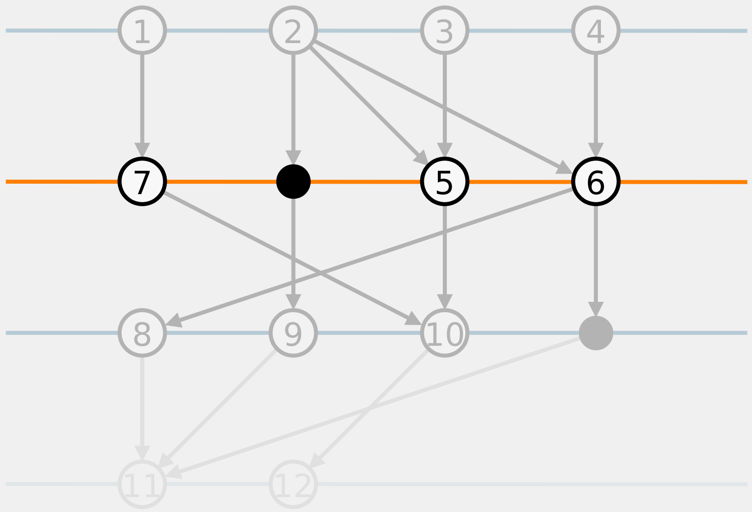
Überblick Kreuzungsreduzierung

- Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart
  - Meist: 2-Level-Kreuzungsreduzierung („Level Sweep“)



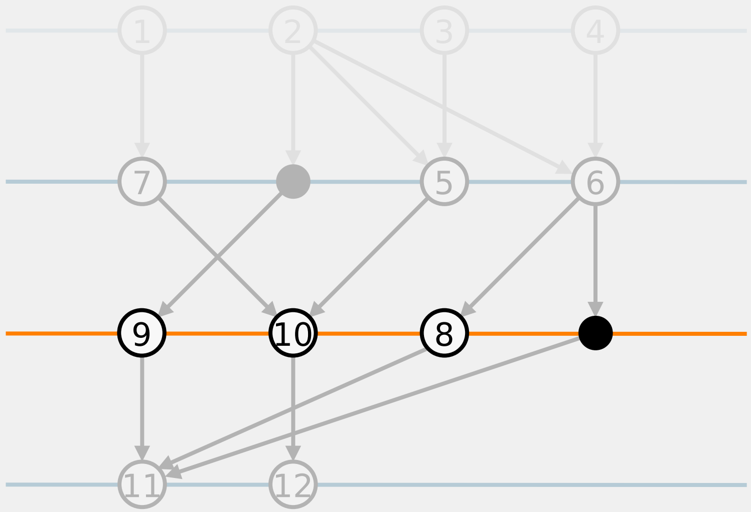
Überblick Kreuzungsreduzierung

- Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart
  - Meist: 2-Level-Kreuzungsreduzierung („Level Sweep“)



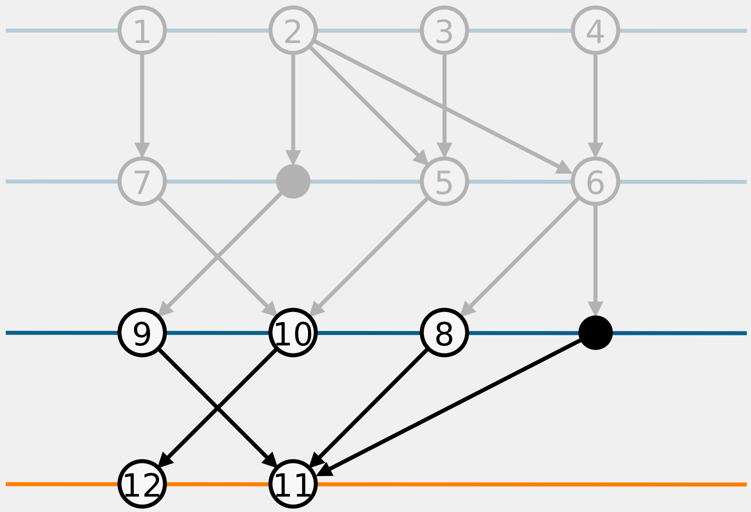
Überblick Kreuzungsreduzierung

- Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart
  - Meist: 2-Level-Kreuzungsreduzierung („Level Sweep“)



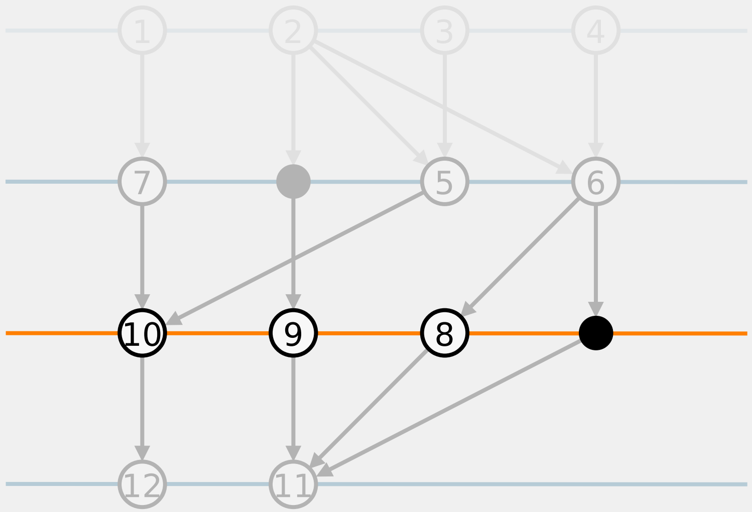
Überblick Kreuzungsreduzierung

- Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart
  - Meist: 2-Level-Kreuzungsreduzierung („Level Sweep“)



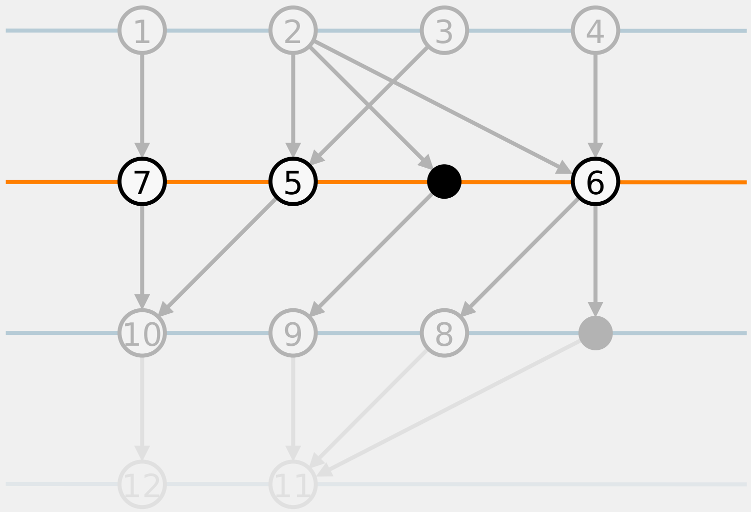
Überblick Kreuzungsreduzierung

- Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart
  - Meist: 2-Level-Kreuzungsreduzierung („Level Sweep“)



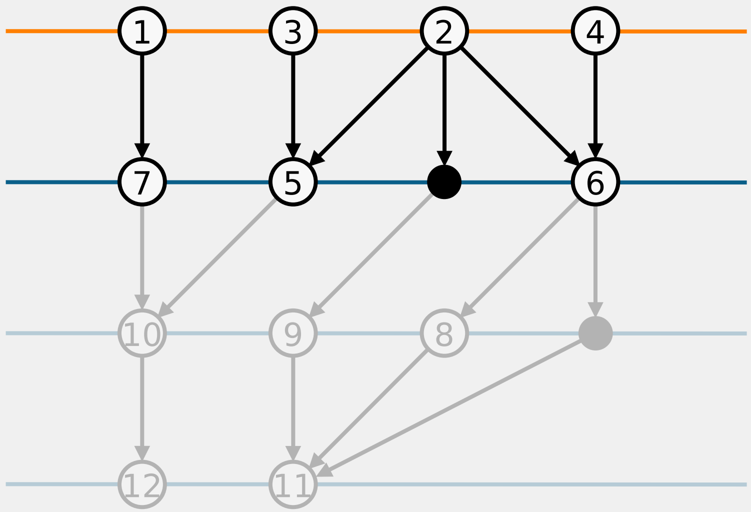
Überblick Kreuzungsreduzierung

- Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart
  - Meist: 2-Level-Kreuzungsreduzierung („Level Sweep“)



Überblick Kreuzungsreduzierung

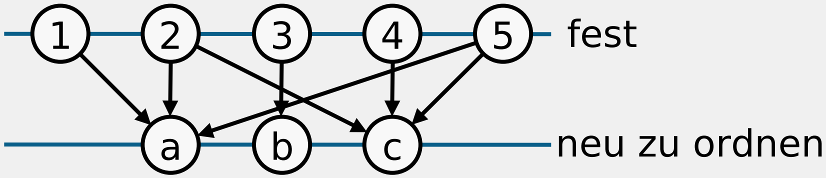
- Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart
  - Meist: 2-Level-Kreuzungsreduzierung („Level Sweep“)



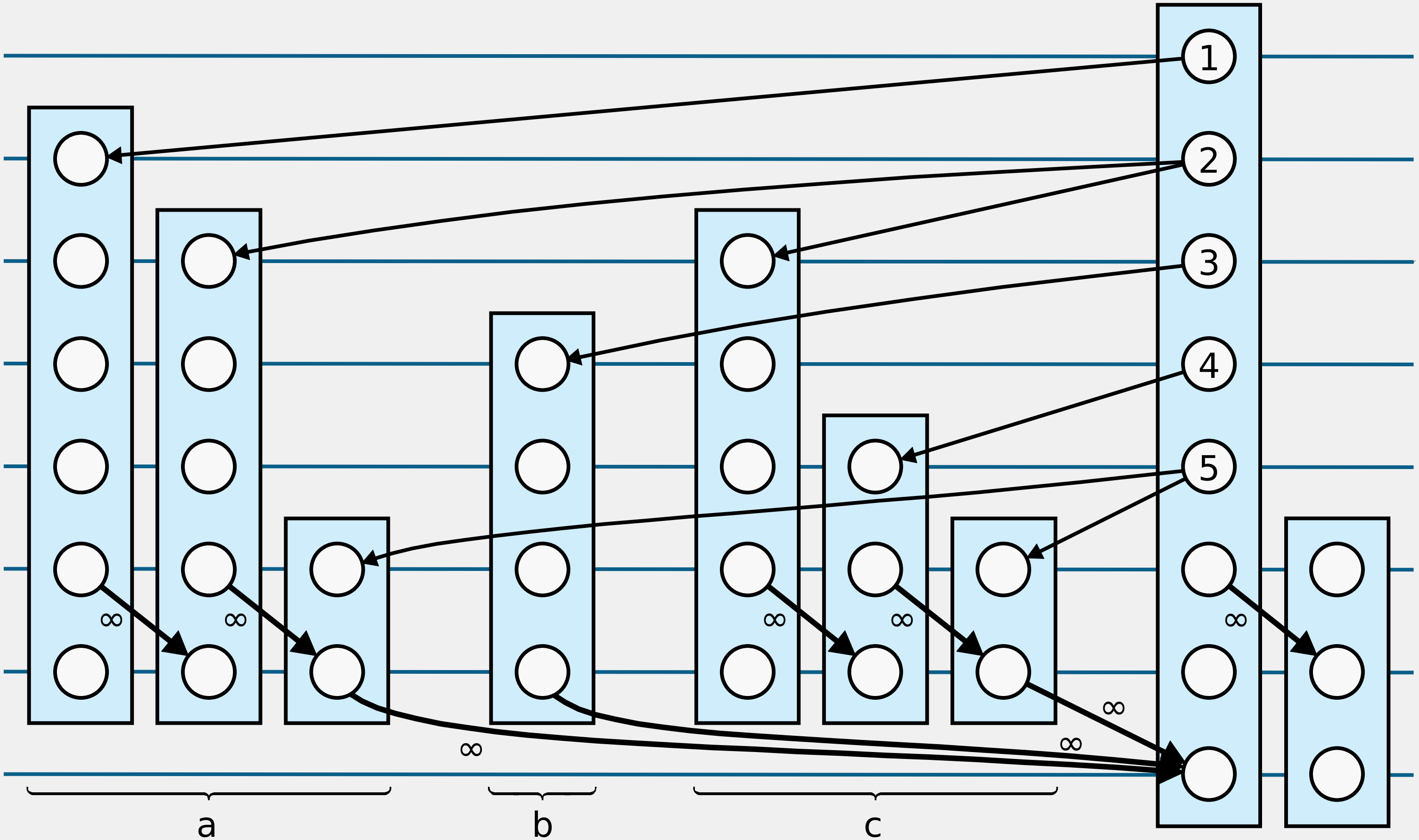
Überblick Kreuzungsreduzierung

- Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart
  - Meist: 2-Level-Kreuzungsreduzierung („Level Sweep“)
    - Selbst das ist schon NP-hart
    - Sogar wenn man die Reihenfolge eines Levels fixiert
    - Sehr viele existierende Heuristiken
- Cluster-Level-Graphen
  - Minimierung von Kantenkreuzungen ist NP-hart (trivialerweise)
  - Minimierung von Cluster/Kanten-Kreuzungen ist NP-hart

Cluster/Kanten-Kreuzungen sind NP-hart



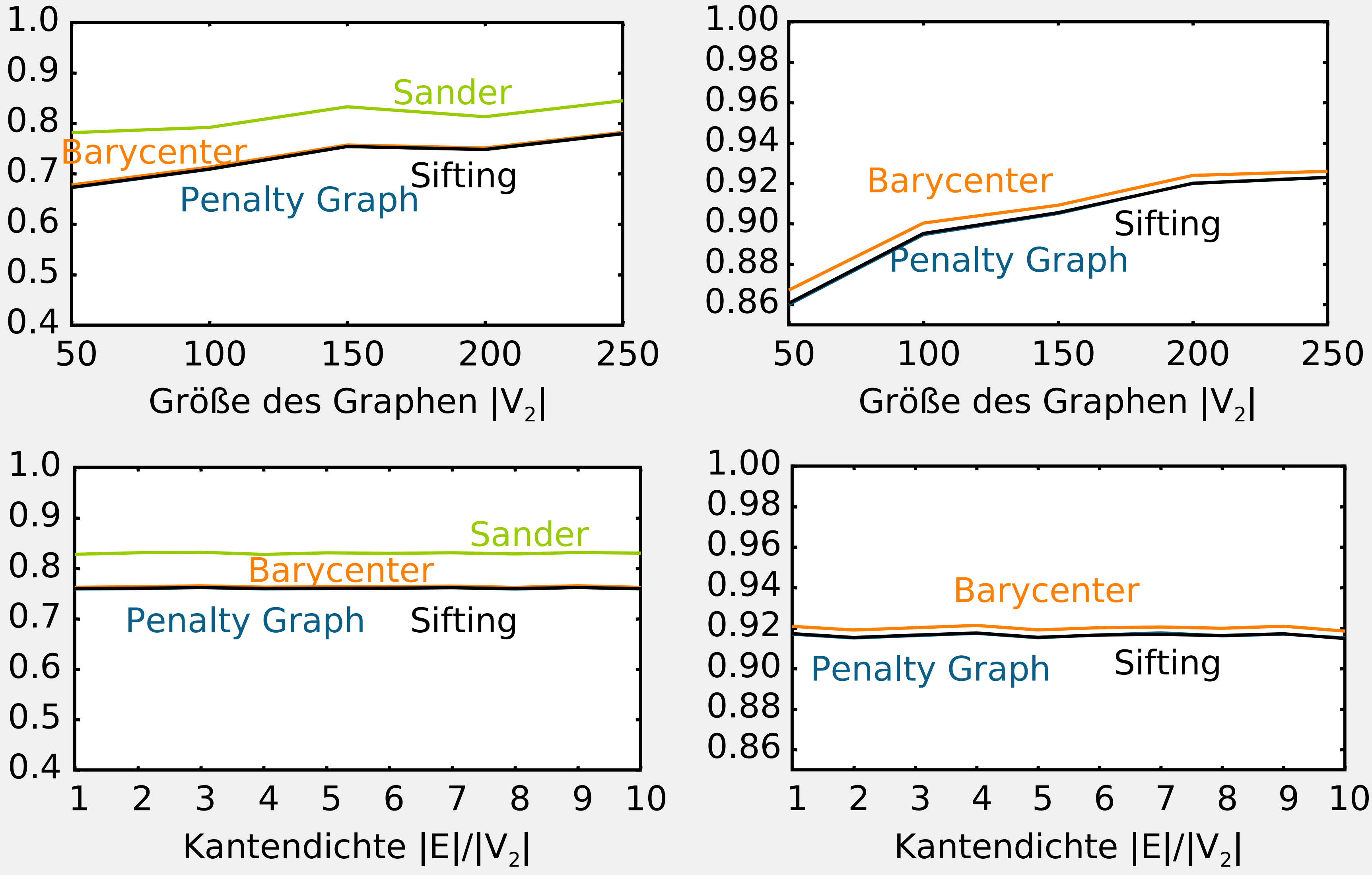
- Reduktion von einseitiger 2-Level-Kreuzungsreduzierung



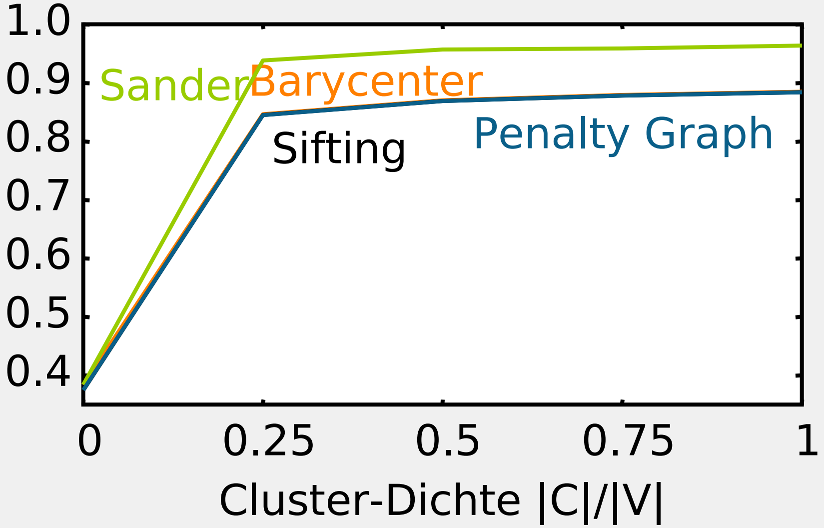
Heuristik

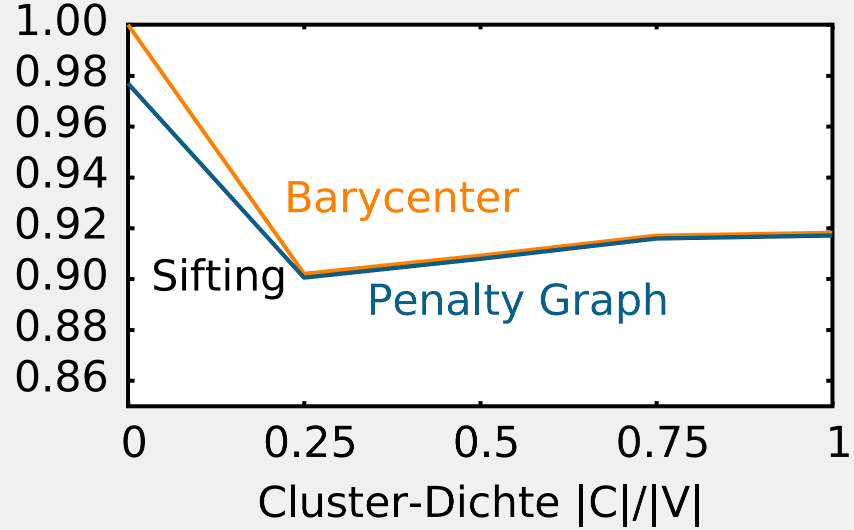
- Existierende Heuristik von Sander
  - Vorgehensweise
    1. Ignoriere Cluster-Bedingungen → Keine gültige Einbettung
    2. Korrigiere die Einbettung durch Umsortieren
  - Ergebnis
    - Cluster/Kanten-Kreuzungen werden komplett ignoriert
    - Unnötige Kreuzungen durch das Umsortieren
- Neue Heuristik
  - Vorgehensweise
    - Allgemeines Schema zur Anwendung bestehender 2-Level-Heuristiken auf Cluster-Level-Graphen
    - Integrierte Berücksichtigung der Cluster-Bedingungen
  - Ergebnis
    - Cluster/Level-Kreuzungen werden berücksichtigt
    - Keine unnötigen zusätzlichen Kreuzungen
    - Vergleichbare Laufzeit

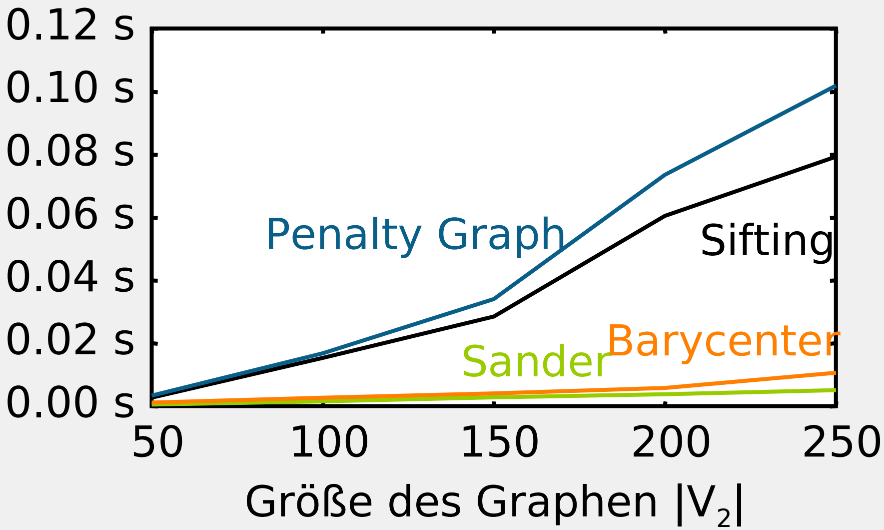
Experimentelle Ergebnisse

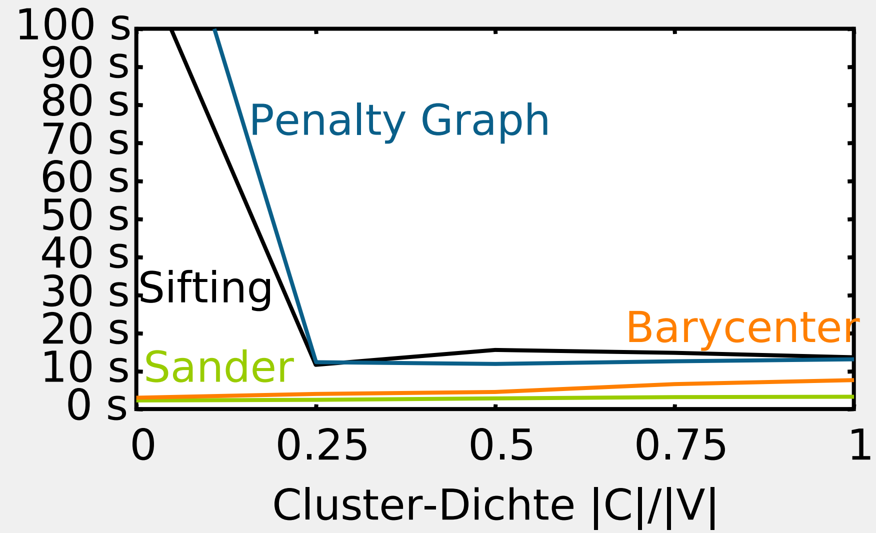


Experimentelle Ergebnisse









Ergebnis

- Kreuzungsreduzierung für Cluster-Level-Graphen
  - NP-hartes Problem
- Neue Heuristik
  - Anwendung von 2-Level-Kreuzungsreduzierung auf Cluster-Level-Graphen
- Ergebnisse
  - Deutlich weniger Kreuzungen als existierende Heuristiken
  - Etwas höhere Laufzeit
  - Bei k=2: Optimum wird erreicht
  - Bei beschränktem Grad der Hierarchie: Optimale Lösung in Polynomialzeit O(kn)
- Benötigt als Zwischenschritt:
  - Kreuzungsreduzierung in 2-Level-Graphen mit Constraints

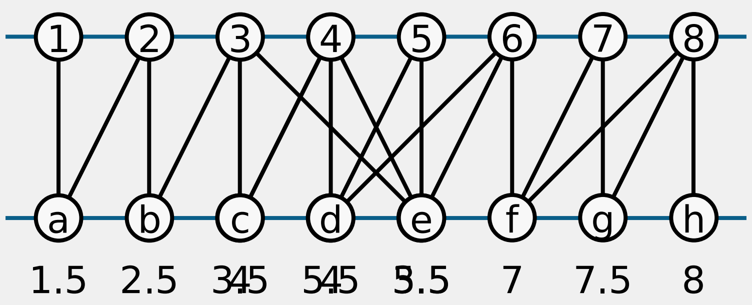
Kreuzungsreduzierung in Level-Graphen mit Constraints

Erfüllung von Nebenbedingungen

[GD 2004]

Einseitige 2-Level Kreuzungsreduzierung

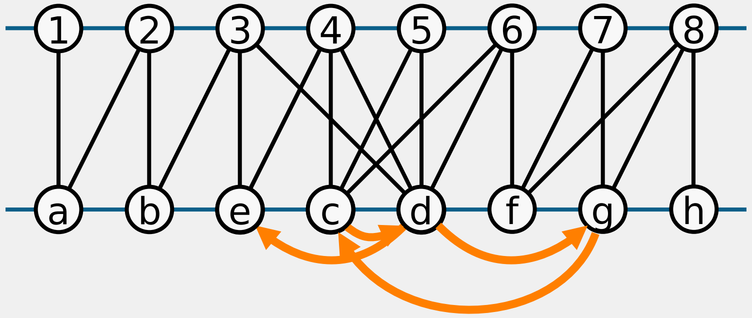
- Wichtiges und gut untersuchtes Problem
  - Gegeben: 2-Level-Graph, feste Reihenfolge des ersten Levels
  - Gesucht: Reihenfolge des zweiten Levels mit wenig Kreuzungen
- NP-hart → Heuristiken
- Barycenter Heuristik
  - Sortiere den zweiten Level nach dem Schwerpunkt der Nachbarn

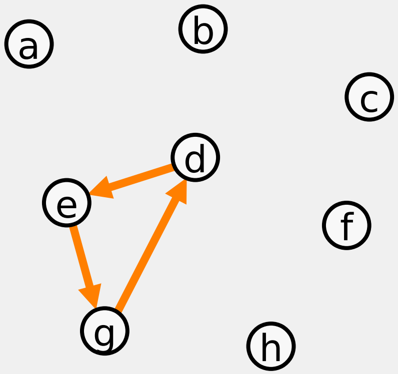


Kreuzungsreduzierung mit Constraints

- Zusätzlich: Constraints
  - Vorgegebene Reihenfolge
  - Verletzt / erfüllt
- Anwendungen
  - Cluster-Level-Graphen
  - Gegeben vom Benutzer
  - Inkrementelle Verfahren („Mental Map“)

- Ziel
  - Erfülle alle Constraints
  - Wenig Kantenkreuzungen
- Constraint-Graph
  - Muss azyklisch sein
  - Wichtiger Spezialfall: Einzelner Pfad + isolierte Knoten



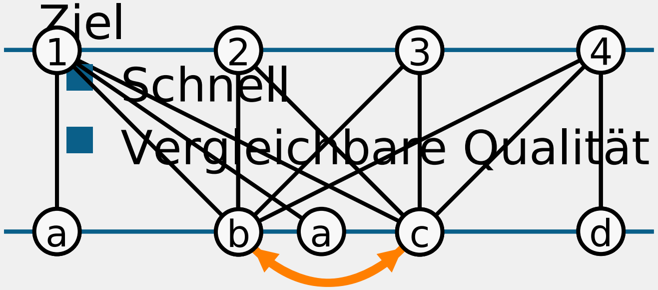


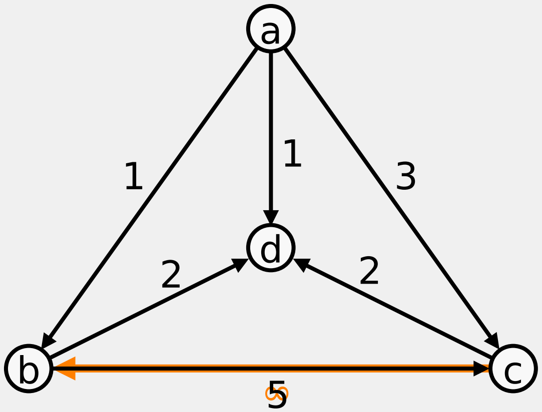
Einfache Heuristiken

- Sander (1996)
  - Erweiterung beliebiger iterativer 2-Level-Kreuzungsreduzierung
  - Beginne mit beliebiger zulässiger Reihenfolge
  - Lasse einen Schritt weg, falls Constraints verletzt würden
- Waddle (2000)
  - Berechne Barycenter-Werte
  - Bei verletztem Constraint: Ändere einen Barycenter-Wert
- Bewertung
  - Sehr schnell
  - Mittelmäßige Qualität
  - Besonders schlecht bei vielen Constraints

Penalty-Graph Heuristik

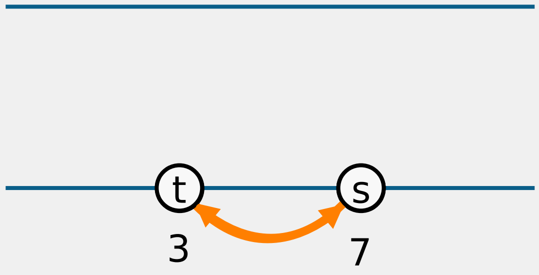
- Schreiber (2001), Finocchi (2001)
  - Berechne den Penalty-Graphen
  - Behandle Constraints als ∞-Kanten
  - Mache den Penalty Graphen azyklisch durch Entfernen von Kanten mit minimalem Gewicht („weighted feedback arc set“, NP-hart)
  - Topologische Sortierung der Knoten
- Bewertung
  - Sehr gute Qualität (bis zu 15% weniger Kreuzungen)
  - Signifikant langsamer
- Ziel
  - Schnell
  - Vergleichbare Qualität





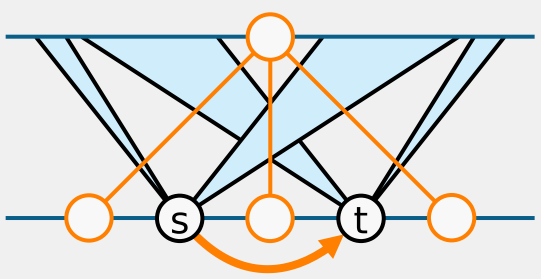
Idee

- Neue Erweiterung der Barycenter-Heuristik
- Erster Schritt: Berechnen Barycenter-Werte b(v) für alle Knoten
- Betrachte ein Constraint (s, t)
  - b(s) > b(t) → wird durch Sortierung verletzt
  - b(s) < b(t) → wird durch Sortierung erfüllt
- Verletzung muss verhindert werden
  - Sander: Verbiete das Vertauschen von s und t
  - Waddle: Ändere den Barycenter-Wert von s

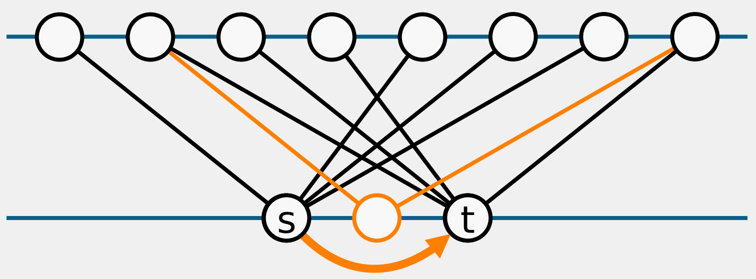


Idee

- Annahme
  - Verletztes Constraint → Kein anderer Knoten sollte dazwischen liegen
- Stimmt das?
  - In einigen Spezialfällen: Ja
  - Im Allgemeinen: Nein
- Aber: Gerechtfertigt durch gute experimentelle Ergebnisse (siehe später)

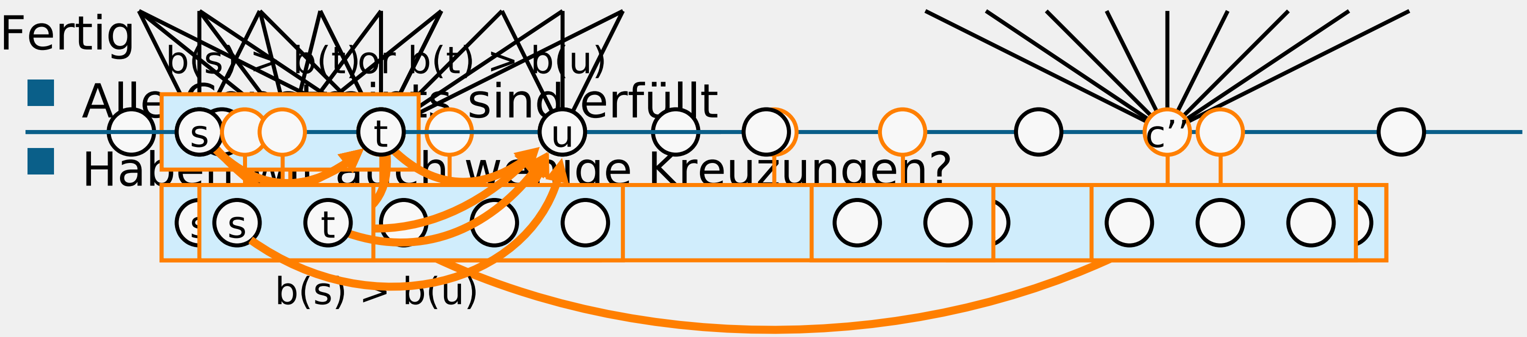


b(s) > b(t)

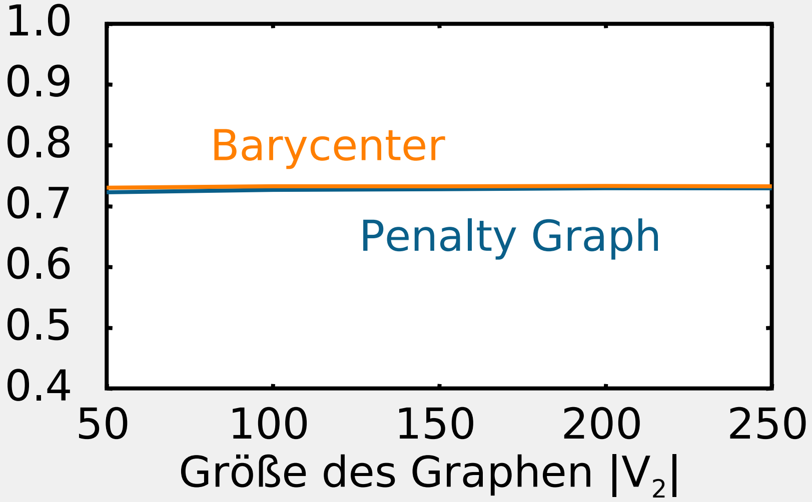


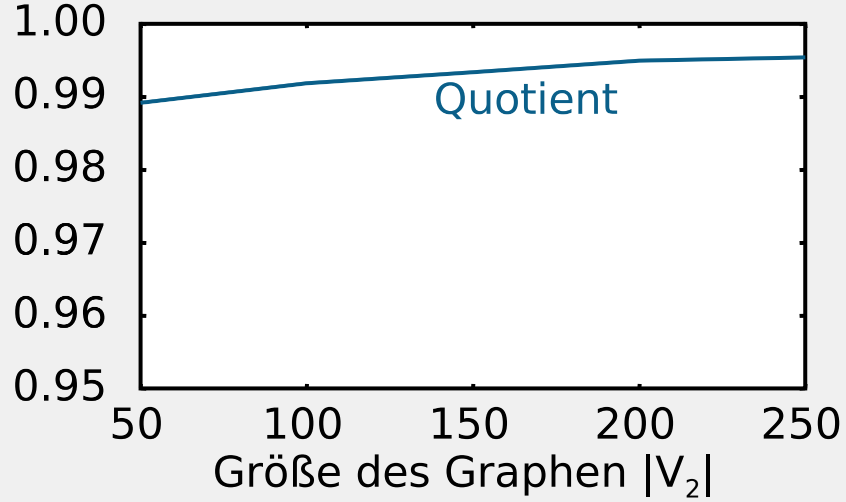
Algorithmus

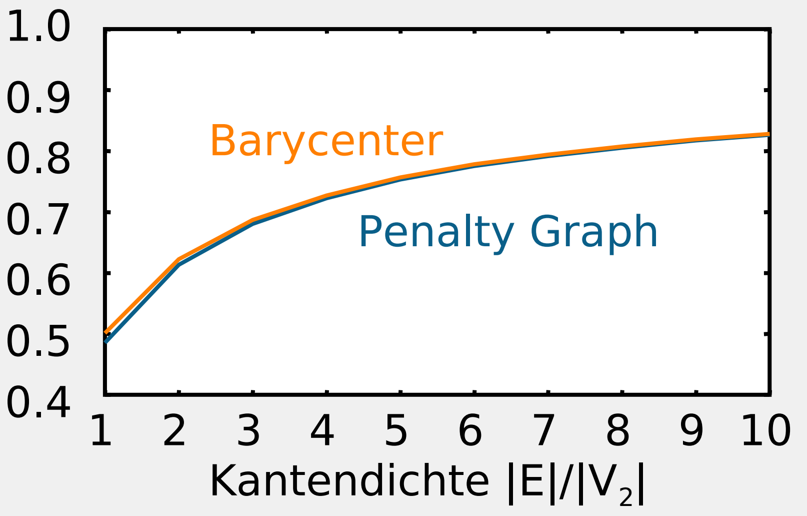
- Für jedes verletzte Constraint
  - Ersetze betroffene Knoten durch einen einzelnen Knoten
  - Berechne dessen Barycenter-Wert in O(1): $b(c) = \dfrac{b(s)\cdot deg(s) + b(t)\cdot deg(t)}{deg(s) + deg(t)}$
- Es dürfen keine Constraint-Zyklen entstehen
  - Constraints müssen in der richtigen Reihenfolge berücksichtigt werden
  - Modifizierte topologische Sortierung
- Wenn es keine verletzten Constraints mehr gibt
  - Sortiere verbliebene Knoten nach dem Barycenter-Wert
  - Füge entfernte Knoten wieder ein
- Fertig
  - Alle Constraints sind erfüllt
  - Haben wir auch wenige Kreuzungen?

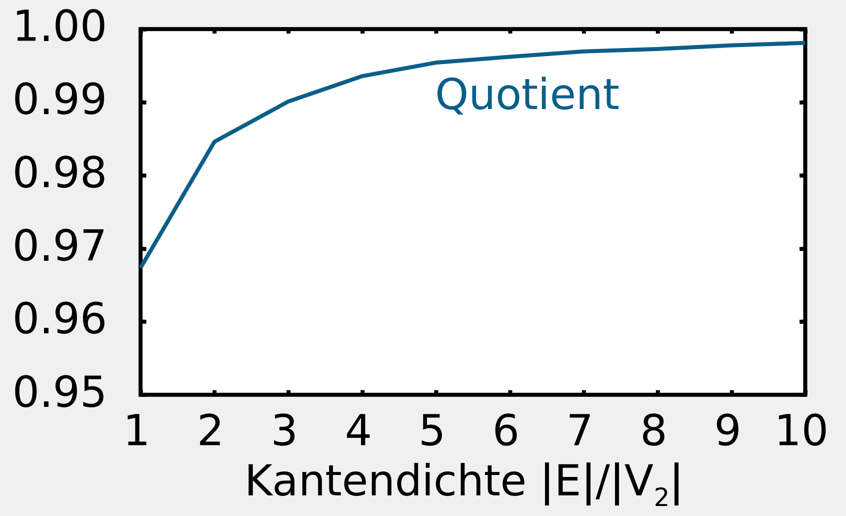


Experimentelle Ergebnisse

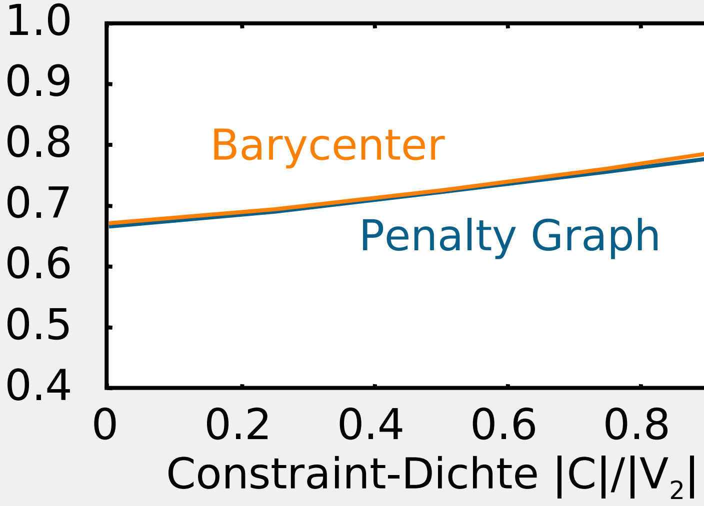


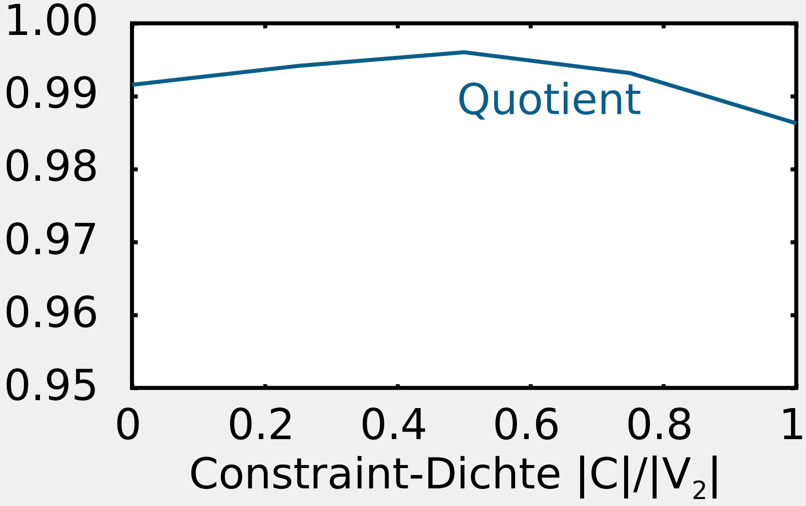


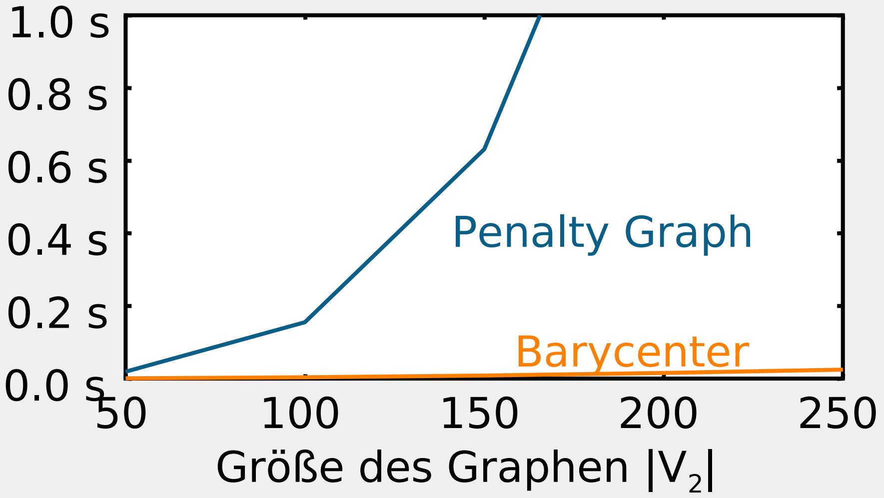




Experimentelle Ergebnisse







Ergebnis

- Einseitige 2-Level-Kreuzungsreduzierung mit Constraints
  - Neue Heuristik
  - Basis: Barycenter-Heuristik
- Qualität
  - Besser als bisherige einfache Erweiterungen
  - Geringere Abhängigkeit von der Anzahl von Constraints
  - Ähnlich zur Penalty-Graph-Heuristik
- Laufzeit
  - $O(|V_2| \log |V_2| + |E| + |C|^2)$
  - Wesentlich schneller als Penalty-Graph Heuristik
- Einfache Implementierung (~ 100 LOC)

Cluster-Level-Planarität

Kommt man ohne Kreuzungen aus?

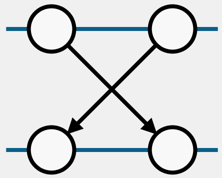
[SOFSEM 2004]

Cluster-Level-Planarität: Überblick

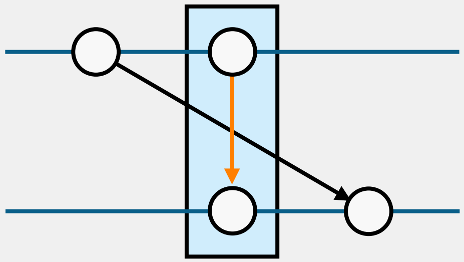
- Kann man einen Cluster-Level-Graph ohne Kreuzungen zeichnen?
  - Bisher nicht betrachtetes Problem
- Verwandte Probleme
  - Cluster-Planarität (c-Planarity)
    - Kann man einen Cluster-Graphen ohne Kreuzungen zeichnen?
    - Komplexität: offen für den allgemeinen Fall
    - $\Theta(n)$, falls das Innere jedes Clusters zusammenhängend ist
  - Level-Planarität
    - Kann man einen Level-Graphen ohne Kreuzungen zeichnen?
    - Komplexität: $\Theta(n)$
- Ergebnis
  - Erweiterung eines existierenden Algorithmus für Level-Planarität
  - Komplexität: offen für den allgemeinen Fall
  - Lösung für eingeschränkte Problemstellung: „ele

Cluster-Level-Planarität: Bedingungen

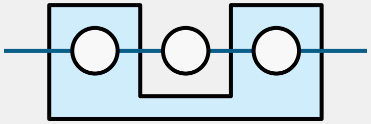
- Kanten/Kanten-Bedingung



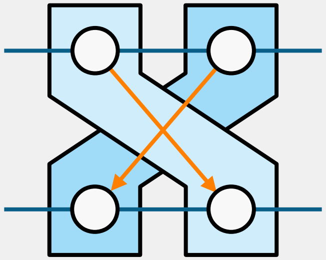
- Kanten/Cluster-Bedingung



- Cluster/Level-Bedingung



- Cluster/Cluster-Bedingung



- Forderung an den Graphen: Level-Zusammenhang
  - Je zwei aufeinander folgende Level in einem Cluster müssen durch mindestens eine Kante verbunden sein
  - Abgeschwächte Form von Cluster-Zusammenhang (c-Connectivity)
  - Ähnlich zu Cluster-Planarität
- Zusätzlich: Proper und nur eine Quelle

Überblick

	Eigenschaften	Komplexität
Cluster-Planarität	beliebig	offen
	Cluster-Zusammenhang	$\Theta(\|V\|)$
Level-Planarität	beliebig	$\Theta(\|V\|)$

Zusammenfassung
Ausblick

Ergebnisse und offene Probleme

Zusammenfassung

- Charakterisierung von Kreuzungen in Cluster-Level-Graphen
- Kreuzungsreduzierung in Cluster-Level-Graphen
  - Einordnung bzgl. der Komplexität
  - Neue Heuristik
- Kreuzungsreduzierung in Level-Graphen mit Constraints
  - Neue Heuristik
- Neues Problem: Cluster-Level-Planarität
  - Für eine eingeschränkte Problemstellung
    - Planaritätstest und Berechnung einer Einbettung
    - Laufzeit $O(k \cdot |V|)$
  - Im allgemeinen Fall: offen
  - Ähnlich zu Cluster-Planarität (c-Planarity)

Fragen?

Vielen Dank für Ihre Aufmerksamkeit!