Kap. 3: Hierarchische Zeichenverfahren
3.4 Kreuzungsminimierung ff

Prof. Dr. Petra Mutzel

Lehrstuhl für
Algorithm Engineering LS11
Universität Dortmund

9./10. VO WS07/08 12./13. November 2007

Überblick zu Kapitel 3

Überblick

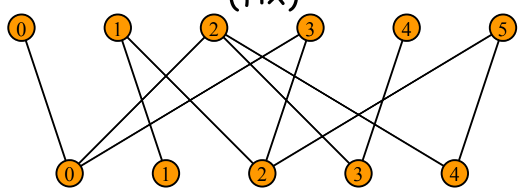
Literatur für diese VO

Originalliteratur:

- M.R. Garey und D.S. Johnson: Crossing Number is NP-complete, SIAM J. Algebraic and Discrete Methods 4, 312-316, 1983

- P. Eades und N.C. Wormald: Edge crossings in drawing bipartite graphs, Algorithmica 11, 379-403, 1994

2-Schichten Kreuzungsminimierung (fix)

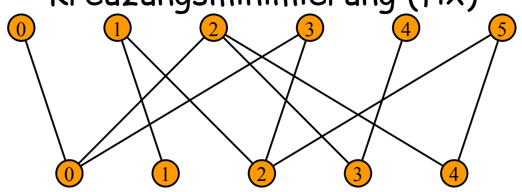


Geg: Bipartiter Graph auf 2 Schichten, Permutation der oberen Schicht ist fixiert

Gesucht: Permutation der Knoten der unteren Schicht mit kleinster Anzahl von Kreuzungen

---

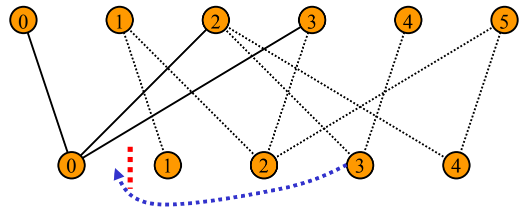
3.4.3 Heuristiken zur 2-Schichten Kreuzungsminimierung (fix)



- Greedy-Insert Heuristik
- Greedy Switch Heuristik
- Split Heuristik
- Barycenter Heuristik
- Median Heuristik
- es gibt viele viele mehr….

Greedy-Insert Heuristik

[Eades & Kelly, 1986]

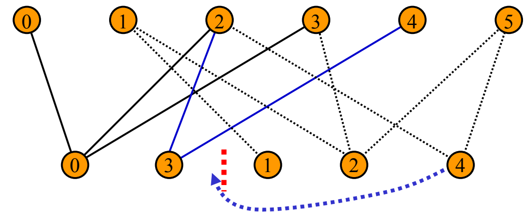


Fixiere von links nach rechts indem in jedem Schritt derjenige Knoten gewählt wird, der mit den bisher gesetzten Knoten am wenigsten Kreuzungen hat.

analog Selection Sort

Greedy-Insert Heuristik

[Eades & Kelly, 1986]

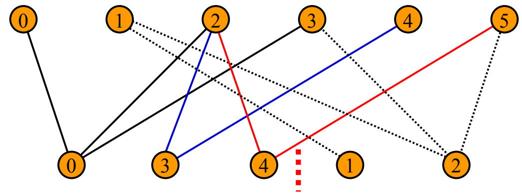


Fixiere von links nach rechts indem in jedem Schritt derjenige Knoten gewählt wird, der mit den bisher gesetzten Knoten am wenigsten Kreuzungen hat.

analog Selection Sort

Greedy-Insert Heuristik

[Eades & Kelly, 1986]

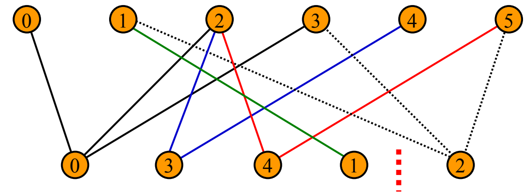


Fixiere von links nach rechts indem in jedem Schritt derjenige Knoten gewählt wird, der mit den bisher gesetzten Knoten am wenigsten Kreuzungen hat.

analog Selection Sort

Greedy-Insert Heuristik

[Eades & Kelly, 1986]



Fixiere von links nach rechts indem in jedem Schritt derjenige Knoten gewählt wird, der mit den bisher gesetzten Knoten am wenigsten Kreuzungen hat.

analog Selection Sort

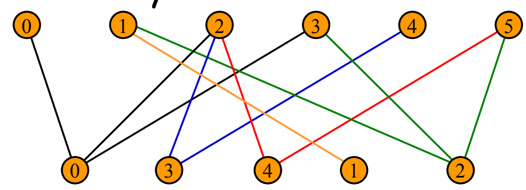
Analyse Greedy-Insert

- Greedy-Insert kann mit Hilfe einer Kreuzungsmatrix in Zeit $O(|V_2|^2)$ realisiert werden.
- Die Kreuzungsmatrix kann in Zeit $O(|V_2|\ |E|)$ berechnet werden.

Mit Merge($N(u),N(v)$): Eintrag (u,v) in Zeit: $\Delta(u)+\Delta(v)$
Summe aller Einträge $\sum\sum (\Delta(u)+\Delta(v)) = \sum 2|E| = 4|V_2||E|$ **NEU**

- Quadratische Kreuzungsmatrix: $|V_2|^2$ Einträge
  - c_{uv}: Anzahl der Kreuzungen zwischen den inzidenten Kanten falls u links von v ($\pi_2(u)<\pi_2(v)$)
  - c_{vu}: Anzahl der Kreuzungen zwischen den inzidenten Kanten falls u rechts von v
  - $c_{vv}=0$

Greedy-Insert Heuristik

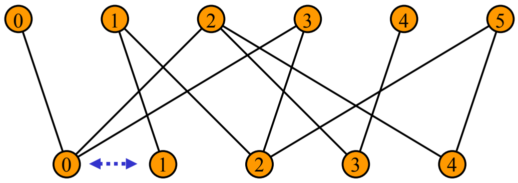


Anzahl der Kreuzungen, falls u vor v

	0	1	2	3	4
0	0	2	2	1	1
1	1	0	0	0	0
2	6	2	0	3	2
3	4	2	3	0	1
4	4	2	3	2	0

Greedy-Switch Heuristik

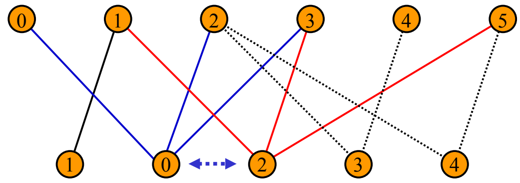
[Eades & Kelly, 1986]



Wiederhole bis keine Kreuzungsreduktion erfolgt:
Durchlaufe von links nach rechts:
Für alle benachbarten Knotenpaare (u,v): Falls es weniger Kreuzungen bringt, wenn man die Paare vertauscht, dann führe Vertauschung durch.

Greedy-Switch Heuristik

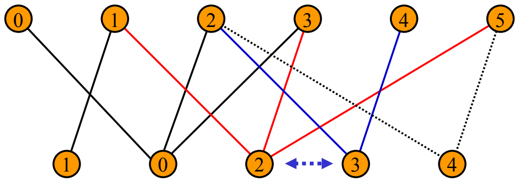
[Eades & Kelly, 1986]



Wiederhole bis keine Kreuzungsreduktion erfolgt:
Durchlaufe von links nach rechts:
Für alle benachbarten Knotenpaare (u,v): Falls es weniger Kreuzungen bringt, wenn man die Paare vertauscht, dann führe Vertauschung durch.

Greedy-Switch Heuristik

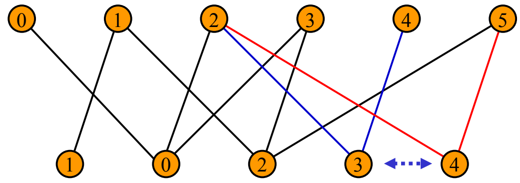
[Eades & Kelly, 1986]



Wiederhole bis keine Kreuzungsreduktion erfolgt:
Durchlaufe von links nach rechts:
Für alle benachbarten Knotenpaare (u,v): Falls es weniger Kreuzungen bringt, wenn man die Paare vertauscht, dann führe Vertauschung durch.

Greedy-Switch Heuristik

[Eades & Kelly, 1986]



Wiederhole bis keine Kreuzungsreduktion erfolgt:
Durchlaufe von links nach rechts:
Für alle benachbarten Knotenpaare (u,v): Falls es weniger Kreuzungen bringt, wenn man die Paare vertauscht, dann führe Vertauschung durch.

Analyse Greedy-Switch

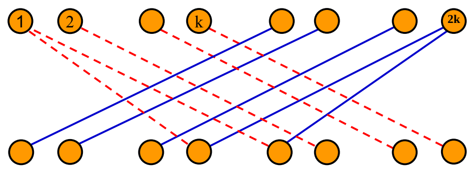
- Greedy-Switch kann mit Hilfe einer Kreuzungsmatrix in Zeit $O(|V_2|^3)$ realisiert werden, denn das Verfahren terminiert erst bei stabiler Reihenfolge (worst case: alle testen)
- bei max. $|V_2|$ Traversierungen: Zeit $O(|V_2|^2)$

  In jeder Traversierung wird mindestens 1 Paar vertauscht, es gibt also maximal $O(|V_2|^2)$ Traversierungen

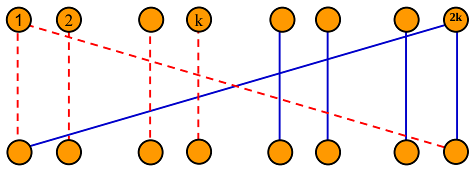
- Die Kreuzungsmatrix kann in Zeit $O(|V_2|\,|E|)$ berechnet werden,

Da Elemente keine lineare Ordnung beschreiben wie bei bubble sort

Pathologisches Beispiel

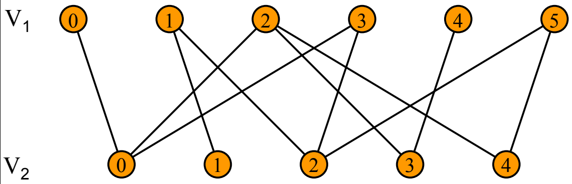


- $k^2+2(k-1)=O(k^2)=O(n^2)$ Kreuzungen



- $2(2k-2)+1=4k-3=O(k)=O(n)$ Kreuzungen

analog Quicksort Split Heuristik



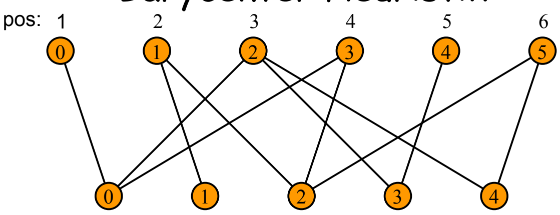
- Wähle ein Pivotelement $p \in V_2$, und platziere alle anderen Knoten entweder
  - links von p, falls $c_{up} < c_{pu}$
  - rechts von p, falls $c_{up} \geq c_{pu}$
- Führe dies rekursiv an L.S. und R.S. von p fort.

Analyse Split-Heuristik

- Die Split-Heuristik kann mit Hilfe einer Kreuzungsmatrix in Zeit $O(|V_2|^2)$ realisiert werden.
- Wie bei Quicksort ist die durchschnittliche Laufzeit unter Gleichverteilungsannahme $O(|V_2| \log |V_2|)$
- Die Kreuzungsmatrix kann in Zeit $O(|V_2|\,|E|)$ berechnet werden.

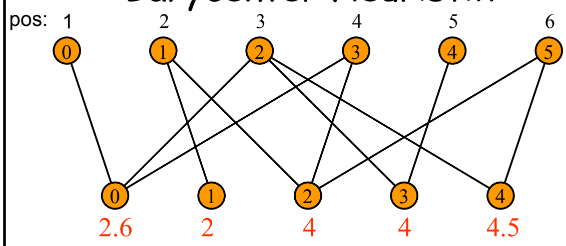
Die folgenden Heuristiken benutzen die Kreuzungsmatrix nicht.

Barycenter Heuristik



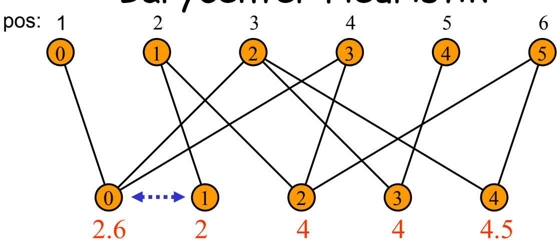
- Für jeden Knoten v auf der unteren Schicht:
  - barycenter(v) = durchschnittliche Position der Nachbarn auf der oberen Schicht

Barycenter Heuristik



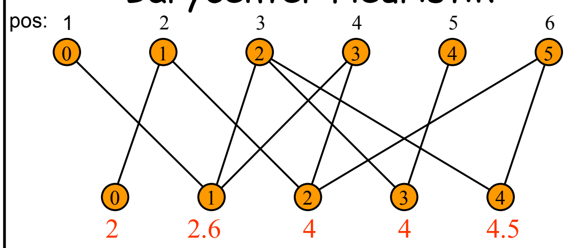
- Für jeden Knoten v auf der unteren Schicht:
  - barycenter(v) = durchschnittliche Position der Nachbarn auf der oberen Schicht

Barycenter Heuristik



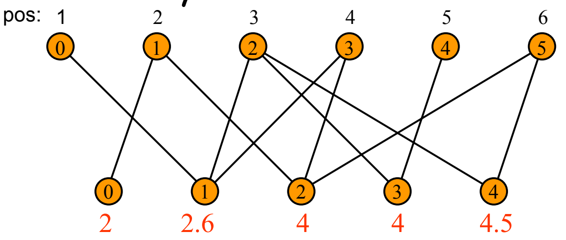
Sortiere die Knoten auf der unteren Schicht nach ihrem Barycenter Wert.

Barycenter Heuristik



Sortiere die Knoten auf der unteren Schicht nach ihrem Barycenter Wert.

Barycenter Heuristik



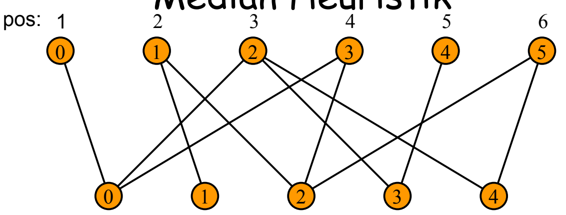
Sortiere die Knoten auf der unteren Schicht nach ihrem Barycenter Wert.

Eine Kreuzung weniger!

Analyse Barycenter-Heuristik

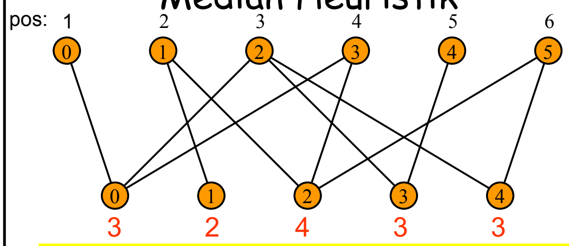
- Die Barycenter-Heuristik kann in Laufzeit $O(|E| + |V_2| \log |V_2|)$ realisiert werden.

Median Heuristik



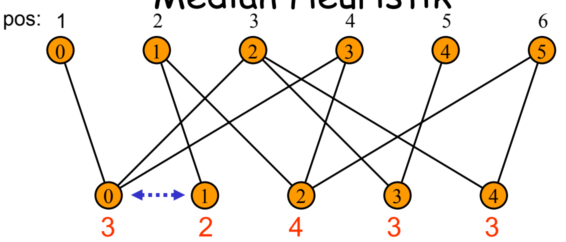
- Für jeden Knoten v auf der unteren Schicht:
  - median(v) = Median der Positionen der Nachbarn auf der oberen Schicht

Median Heuristik



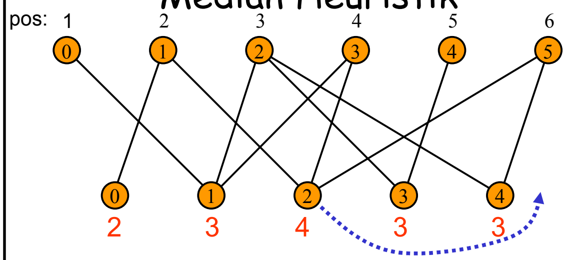
- Für jeden Knoten v auf der unteren Schicht:
  - median(v) = Median der Positionen der Nachbarn auf der oberen Schicht

Median Heuristik



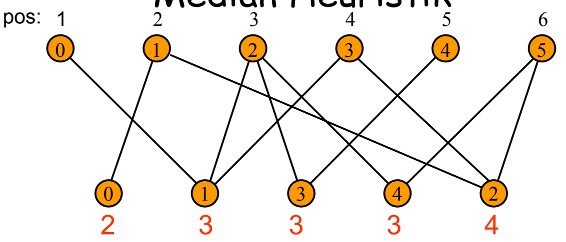
- Sortiere die Knoten auf der unteren Schicht nach ihrem Median Wert.

Median Heuristik



- Sortiere die Knoten auf der unteren Schicht nach ihrem Median Wert.

Median Heuristik



- Sortiere die Knoten auf der unteren Schicht nach ihrem Median Wert.

Analyse Median Heuristik

- Die Median-Heuristik kann in Laufzeit $O(|E|)$ realisiert werden (Vorr. keine singulären Knoten).
- Denn:
  - alle Median-Werte können in Zeit $O(|E|)$ berechnet werden, und
  - die Median-Werte liegen zwischen 1 und $|V_2|$, deshalb kann mittels Bucket-Sort sortiert werden.

Analyse (1)

- Sei **kreuz**(G,π_1,π_2) = Anzahl der Kreuzungen in einer Zeichnung von $G=(V_1,V_2,E)$ in der V_1 gemäß π_1 und V_2 gemäß π_2 permutiert ist
- **minkreuz**(G,π_1)= das Minimum von kreuz(G, π_1, π_2) über alle Permutationen π_2 von V_2 bei geg. Permutation π_1

Beobachtung:

- **kreuz**$(G,\pi_1,\pi_2) = \sum_{u,v\in V_2,\ \pi_2(u)<\pi_2(v)} c_{uv}$
- **minkreuz**$(G, \pi_1) \geq \sum_{u,v\in V_2} \min(c_{uv}, c_{vu})$

Beweis: ✓

Analyse Barycenter und Median

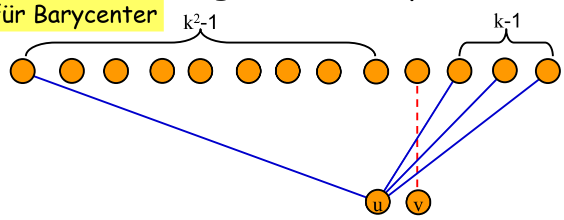
- Sei **barykreuz**(G, π_1)=Zahl der Kreuzungen mit Barycenter-Heuristik und
- **medkreuz**(G, π_1) = Zahl der Kreuzungen mit Median-Heuristik, wenn die Sortierung bei Knoten mit gleichen Median-Werten, alle Knoten mit ungeradem Grad links von denjenigen mit geradem Grad sortiert

- **Satz:** Falls minkreuz(G, π_1)=0, so gilt barykreuz(G, π_1)=0 und medkreuz(G, π_1) = 0.

Beweis: Übungen

Pathologische Beispiele

für Barycenter



barycenter(u) = $1/k(1+(k^2+1)+(k^2+2)+\ldots+(k^2+k-1))$ =

- $=1/k\ (1+(k-1)k^2+\sum i\ (i=1,\ldots,k-1))$ =
- $=1/k+k^2-k/2-1/2 < k^2$ für $k\geq 2$

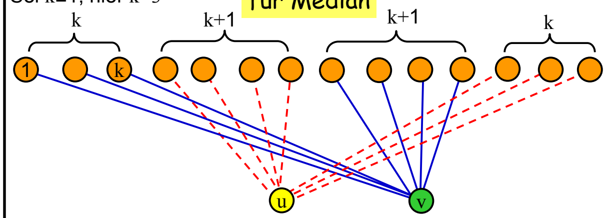
barycenter(v) = k^2

→ barykreuz(G,π_1) = $k-1$ aber minkreuz(G,π_1)=1

Pathologische Beispiele

Sei $k\geq 1$, hier k=3

für Median



median(u) = $2k+1$

median(v) = $2k+2$

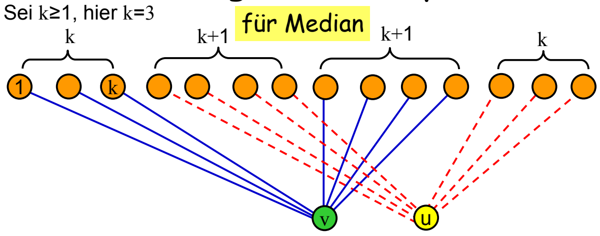
medkreuz(G,π_1) = $2k(k+1)+k^2$ Kreuzungen aber

minkreuz(G,π_1)=$(k+1)^2$ Kreuzungen

Pathologische Beispiele

Sei k≥1, hier k=3

für Median



median(u) = 2k+1

median(v) = 2k+2

medkreuz(G,π_1) = 2k(k+1)+k² Kreuzungen aber

minkreuz(G,π_1)=(k+1)² Kreuzungen

Analyse Barycenter und Median

Lemma:

- Zu jedem k≥2 gibt es einen Digraphen $G=(V_1,V_2,E)$ mit $|V_1|=k^2+k-1$ und $|V_2|=2$ sowie eine Permutation π_1 von V_1, so dass barykreuz(G,π_1) / minkreuz(G,π_1) = $\Omega(\sqrt{|V_1|})$
- Zu jedem k≥1 gibt es einen Digraphen $G=(V_1,V_2,E)$ mit $|V_1|=4k+2$ und $|V_2|=2$ sowie eine Permutation π_1 von V_1, so dass medkreuz(G,π_1) / minkreuz(G,π_1) ≥ 3-O(1/$|V_1|$)

Beweis: wegen pathologischer Beispiele:

- $k=\theta(\sqrt{|V_1|})$
- medkreuz(G,π_1) / minkreuz(G,π_1) = $(2k(k+1)+k^2)/(k+1)^2 = (3k^2+2k)/(k^2+2k+1) \geq (3k+9)/(k+3)-7/(k+3) = 3-O(1/k)$

für Median ist Schranke scharf!

Analyse Barycenter und Median

Satz:

- Für alle $G=(V_1,V_2,E)$ und alle Permutationen π_1 von V_1 gilt medkreuz(G,π_1) ≤ 3 minkreuz(G,π_1)

Beweis Analyse

Lemma: Seien $u,v\in V_2$ und MEDIAN platziere u links von v. Dann gilt: $c_{uv} \leq 3c_{vu}$

Beweis des Lemmas: s. nächste Folien

Beweis des Satzes unter Verwendung von Lemma:

Also haben wir: $c_{uv} \leq 3c_{vu}$ für alle Paare $u,v\in V_2$ in der Median Sortierung

$\Rightarrow c_{uv} \leq 3 \min(c_{uv}, c_{vu})$

$\Rightarrow$ medkreuz$(G,\pi 1) = \sum_{\pi_2(u)<\pi_2(v)} c_{uv} \leq 3 \sum_{\pi_2(u)<\pi_2(v)} \min(c_{uv},c_{vu})$

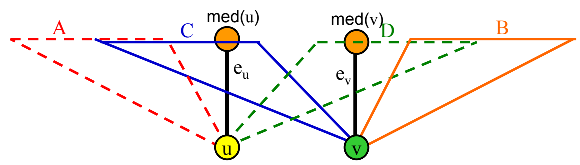
≤ 3 minkreuz(G,π_1)

Beweis Lemma

Beweis: Seien $u,v\in V_2$ und MEDIAN platziere u links von v.

A = $\{(w,u)\in E \mid \pi_1(w) < \text{med}(u)\}$, B = $\{(w,v)\in E \mid \pi_1(w) > \text{med}(v)\}$

C = $\{(w,v)\in E \mid \pi_1(w) < \text{med}(v)\}$, D = $\{(w,u)\in E \mid \pi_1(w) > \text{med}(u)\}$



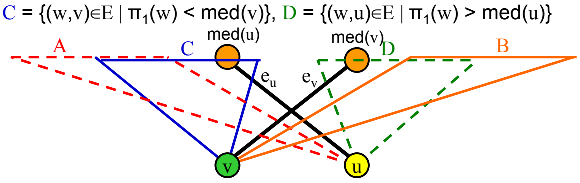
Sei a=|A|, b=|B|, c=|C|, d=|D|, dann gilt $c_{uv} \leq ac+cd+bd+c+d$

Beweis Lemma ff

Beweis: Seien $u,v\in V_2$ und MEDIAN platziere u links von v.

A = $\{(w,u)\in E \mid \pi_1(w) < \text{med}(u)\}$, B = $\{(w,v)\in E \mid \pi_1(w) > \text{med}(v)\}$

C = $\{(w,v)\in E \mid \pi_1(w) < \text{med}(v)\}$, D = $\{(w,u)\in E \mid \pi_1(w) > \text{med}(u)\}$



Sei a=|A|, b=|B|, c=|C|, d=|D| und

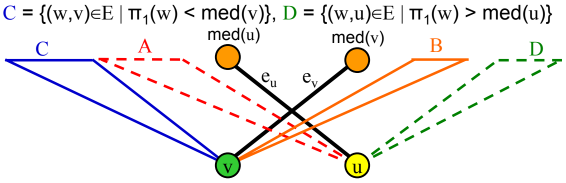
sei $\varepsilon = \begin{cases} 0, \text{ falls med(u)=med(v)} \\ 1, \text{ sonst} \end{cases}$ dann gilt: $c_{vu} \geq ab+a+b+\varepsilon$

Beweis Lemma ff

Beweis: Seien $u,v \in V_2$ und MEDIAN platziere u links von v.

A = $\{(w,u)\in E \mid \pi_1(w) <$ med(u)$\}$, B = $\{(w,v)\in E \mid \pi_1(w) >$ med(v)$\}$

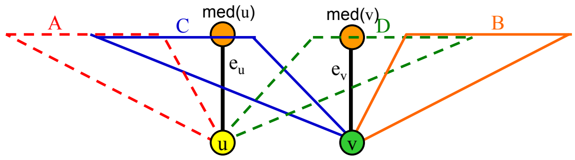
C = $\{(w,v)\in E \mid \pi_1(w) <$ med(u)$\}$, D = $\{(w,u)\in E \mid \pi_1(w) >$ med(u)$\}$



Sei $a=|A|$, $b=|B|$, $c=|C|$, $d=|D|$ und

sei $\varepsilon = \begin{cases} 0, \text{ falls med(u)=med(v)} \\ 1, \text{ sonst} \end{cases}$ dann gilt: $\boxed{c_{vu} \geq ab+a+b+\varepsilon}$ ★

Beweis Lemma ff



Weiterhin gilt:

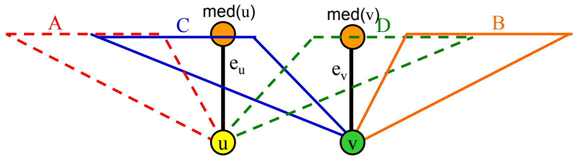
$a=d$, falls $\Delta(u)$ ungerade und $a+1=d$, falls $\Delta(u)$ gerade,

$c=b$, falls $\Delta(v)$ ungerade und $c+1=b$, falls $\Delta(v)$ gerade

$\Rightarrow \boxed{d \leq a+1 \text{ und } c \leq b}$

★ $\Rightarrow c_{uv} \leq ab+b(a+1)+b(a+1)+b+a+1 =$ $\boxed{3ab+a+3b+1}$ ★

Beweis Lemma ff



Jetzt zeigen wir: $c_{uv} \leq 3c_{vu}$

Annahme: $c_{uv} > 3c_{vu}$ ★ ★

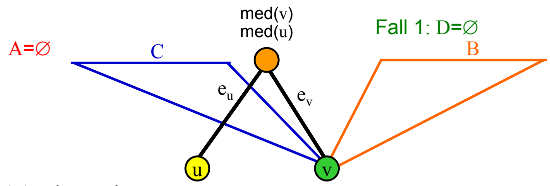
Wir haben $3ab+a+3b+1 \geq c_{uv} > 3c_{vu} \geq 3ab+3a+3b+3\varepsilon$

$\Rightarrow 0 > 3a-a+3\varepsilon-1 = 2a+3\varepsilon-1$ R.S. ganzzahlig:

$\Rightarrow \boxed{a = \varepsilon = 0}$

$\Rightarrow d \leq 1 \quad \Rightarrow A = \emptyset, |D| \leq 1 \quad \Rightarrow \Delta(u) \leq 2$

Beweis Lemma ff



Jetzt zeigen wir: $c_{uv} \leq 3c_{vu}$

Annahme: $c_{uv} > 3c_{vu}$

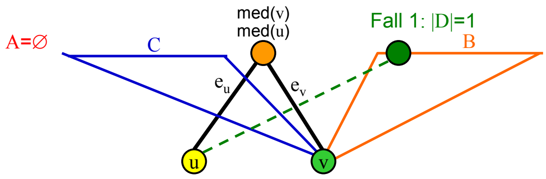
Fall 1: $\Delta(u)=1$: Dann ist $d=0$. Einsetzen in ★:

$c_{uv} \leq ac+cd+bd+c+d = c$

und ★: $c_{vu} \geq ab+a+b+\varepsilon = b$

Also: $b \geq c \geq c_{uv} > 3c_{vu} \geq 3b$ ⚡

Beweis Lemma ff



Fall 2: $\Delta(u)=2$: Dann ist $d=1$. Wegen $\varepsilon=0$ gilt med(u)=med(v)

Regel: Falls $\Delta(u)$ gerade und $\Delta(v)$ ungerade, so gilt $\pi_2(v)<\pi_2(u)$.

Also ist $\Delta(v)$ gerade und deshalb $\boxed{c=b-1}$

Einsetzen in ★: $c_{uv} \leq ac+cd+bd+c+d = b-1+b+b-1+1 = 3b-1$

und ★: $c_{vu} \geq ab+a+b+\varepsilon = b$

Also: $3b-1 \geq c_{uv} > 3c_{vu} \geq 3b$ ⚡

Ende des Beweises

Praktische Experimente für 2-Schichten Kreuzungsminimerung

Experimental Setup

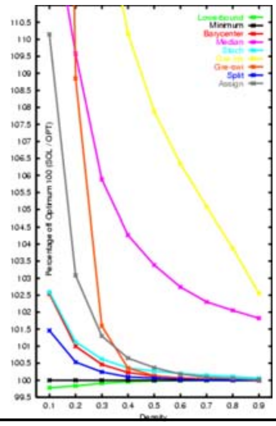
- Wiederholung der Experimente von Eades und Kelly 1986 um sicherzustellen, dass wir gleiche Resultate erhalten:
  - je 100 zufällig generierte Graphen mit |L|=|R|=20 und steigender Dichte
  - Ergebnis: „All heuristics behave well for moderate and high density“

Experimental Setup

- Random Graph Generator:
- random_bigraph von Stanford GraphBase von Knuth, denn diese Generatoren sind
  - Hardware unabhängig
  - jederzeit von allen überall reproduzierbar

- SUN Sparcstation 10, Linux

Relative Qualität



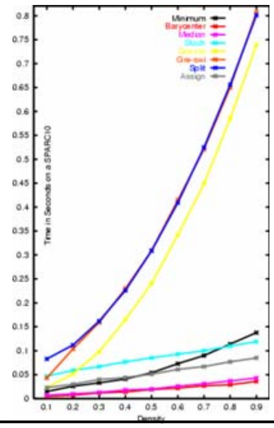
100 (SOL / OPT)

Sieger: Split Heuristik Barycenter und Stochastik auch gut

Median scheint schlecht zu sein

je 100 Instanzen
20+20 Knoten
mit steigender Dichte

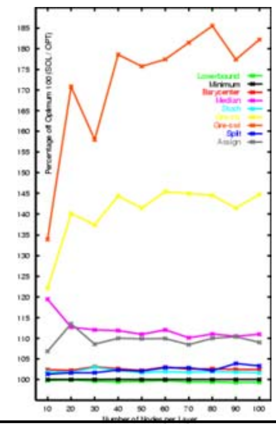
Laufzeiten in Sekunden



Exakter Algorithmus schneller als viele Heuristiken

je 100 Instanzen
20+20 Knoten
mit steigender Dichte

Relative Qualität



100 (SOL / OPT)

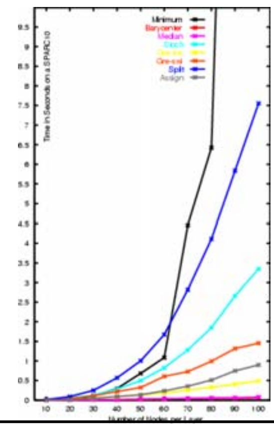
Sieger: Split Heuristik Barycenter und Stochastik

Verlierer: Greedy-Switch und Greedy-Insert

Untere Schranke ist sehr nah am Optimum

je 10 Instanzen
dünne Graphen mit aufsteigender Größe
10+10 bis 100+100

Laufzeiten in Sekunden



Split Heuristik bis 60+60 Langsamer als exakter Algorithmus

Sieger: Barycenter Heuristik und Stochastik

je 10 Instanzen
dünne Graphen mit aufsteigender Größe
10+10 bis 100+100

Eigentlich interessiert uns k-Schichten Kreuzungsminimierung

- Das betrachtete wird im Graphlayout nur als Teilproblem für die k-Schichten Kreuzungsminimierung benutzt.

- Uns interessiert also eigentlich, wie gut diese Heuristiken für das Originalproblem geeignet sind.

- Wiederhole Experimente für zwei Schichten, bei denen nichts fixiert ist

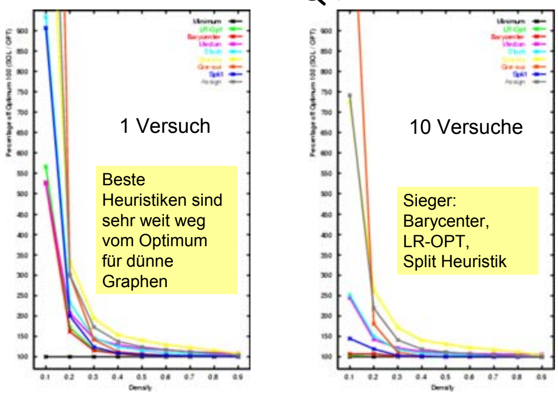
2-Schichten Kreuzungsminimierung (frei)

- Führe abwechselnd die folgenden Schritte durch:
- Fixiere obere Schicht, berechne untere Schicht
- Fixiere untere Schicht, berechne obere Schicht,
- Bis keine Verbesserung mehr möglich

- Gleiche Heuristiken für fixierte Probleme wie oben plus LR-OPT: jeweils der exakte Algorithmus für eine fixierte Schicht

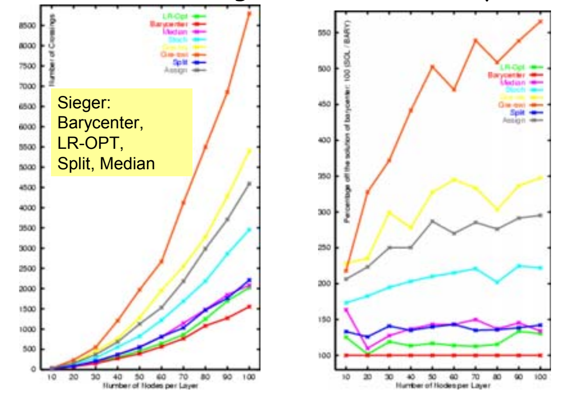
- Optimale Berechnung damals nur für sehr kleine Instanzen möglich (bis ca. 15 Knoten auf kleinerer Schicht): Kombination von Enumeration mit exaktem fixierten Algorithmus

Relative Qualität



je 100 Instanzen 10+10 Knoten mit steigender Dichte

Anzahl an Kreuzungen Relativ zu Barycenter



je 10 Instanzen dünne Graphen von 10+10 bis 100+100

Computational Results

- Exakter Algorithmus kann optimale Kreuzungszahl (fix) berechnen für ca. 60 Knoten auf beweglicher Schicht
- Barycenter und Median sind die besten Heuristiken für Originalproblem
- Laufzeit der meisten Heuristiken wird dominiert durch das Kreuzungszählen (Experimente waren vor Entwicklung von BJM)

und jetzt: exakter Algorithmus