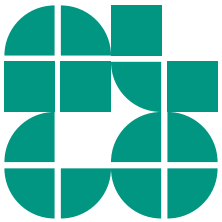
Vorlesung Algorithmische Kartografie

Randbeschriftungen Teil 2

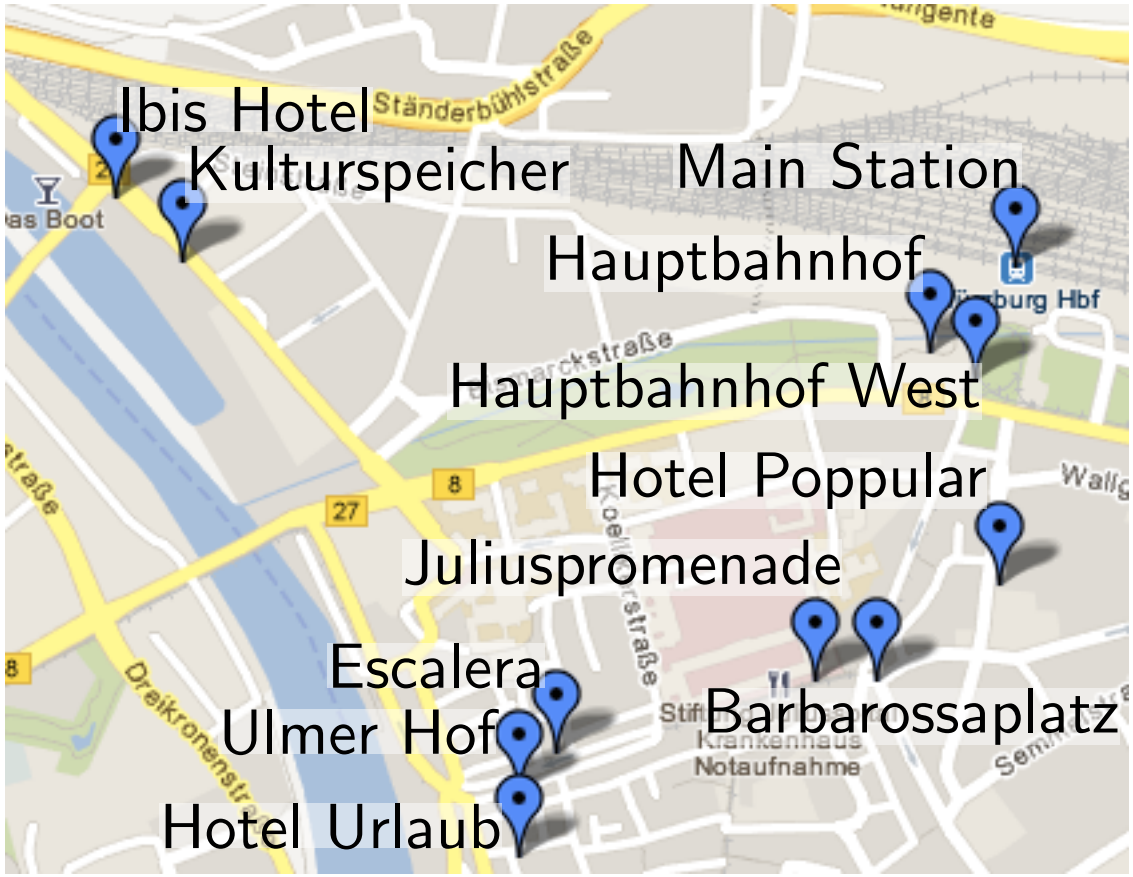
LEHRSTUHL FÜR ALGORITHMIK I · INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Martin Nöllenburg

28.05.2013



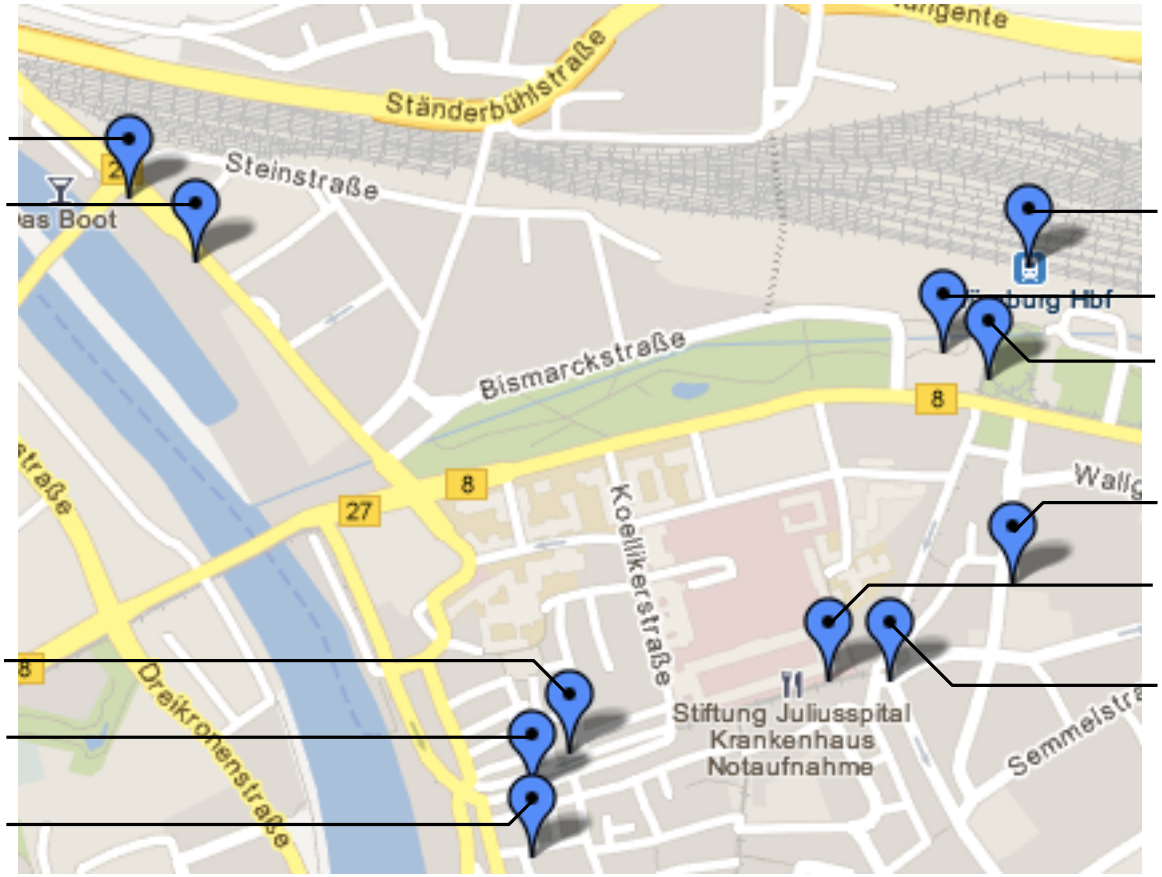
Randbeschriftungen (Wdh.)



Nachteile von internen Beschriftungen

- funktioniert nur für „dünne“ Punktmengen gut
- Label verdecken unterliegende Topographie
- manche Label müssen ausgeblendet werden
- visuelles *cluttering*

Randbeschriftungen (Wdh.)



Ibis Hotel
Kulturspeicher
Escalera
Ulmer Hof
Hotel Urlaub
Main Station
Hauptbahnhof
Hauptbahnhof West
Hotel Poppular
Juliuspromenade
Barbarossaplatz

Vorteile von Randbeschriftungen

- dichte Punktmengen lassen sich beschriften
- weniger Überdeckungen der Karte
- schnelleres Finden Name → Ort
- besser geeignet für Label mit viel Text

Das Randbeschriftungs-Problem

Geg: n Punkte in Rechteck R und für jeden Punkt ein Label repräsentiert durch seine bounding box

Ges:
- **zulässige Labelplatzierung** am Rand von R
- **zulässiger Leader** zw. jedem Punkt und seinem Label
- optimale **Beschriftungsqualität**

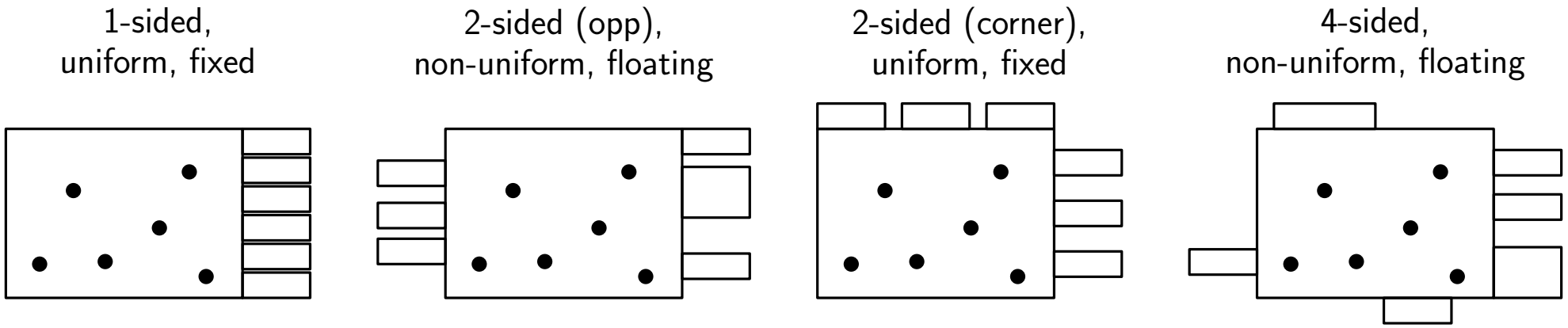
Das Randbeschriftungs-Problem

Geg: n Punkte in Rechteck R und für jeden Punkt ein Label repräsentiert durch seine bounding box

Ges:
- **zulässige Labelplatzierung** am Rand von R
- **zulässiger Leader** zw. jedem Punkt und seinem Label
- optimale **Beschriftungsqualität**

(1) zulässige Labelplatzierung

- disjunkte Label
- wo auf dem Rand?
- (nicht-) uniforme Labelgröße?
- fixed oder floating Position?



Das Randbeschriftungs-Problem

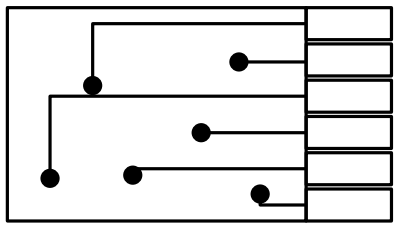
Geg: n Punkte in Rechteck R und für jeden Punkt ein Label repräsentiert durch seine bounding box

Ges:
- **zulässige Labelplatzierung** am Rand von R
- **zulässiger Leader** zw. jedem Punkt und seinem Label
- optimale **Beschriftungsqualität**

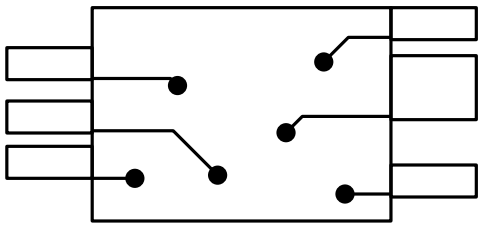
(1) zulässige Labelplatzierung

(2) zulässige Leader

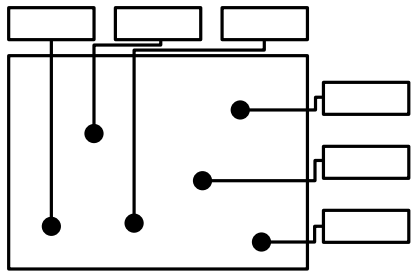
- kreuzungsfrei?
- Form (**p**arallel, **o**rthogonal, **d**iagonal, **s**traight, . . .)
- Hindernisse vermeiden?
- fixed oder sliding Ports?



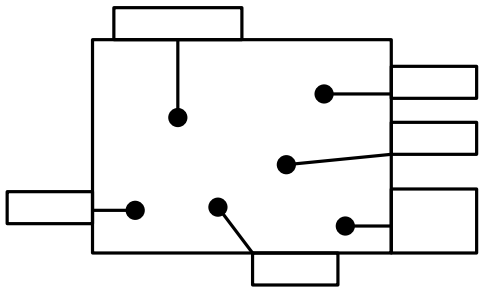
po-leaders, fixed ports



do-leaders, sliding ports



opo-leaders, fixed ports



s-leaders, sliding ports

Das Randbeschriftungs-Problem

Geg: n Punkte in Rechteck R und für jeden Punkt ein Label repräsentiert durch seine bounding box

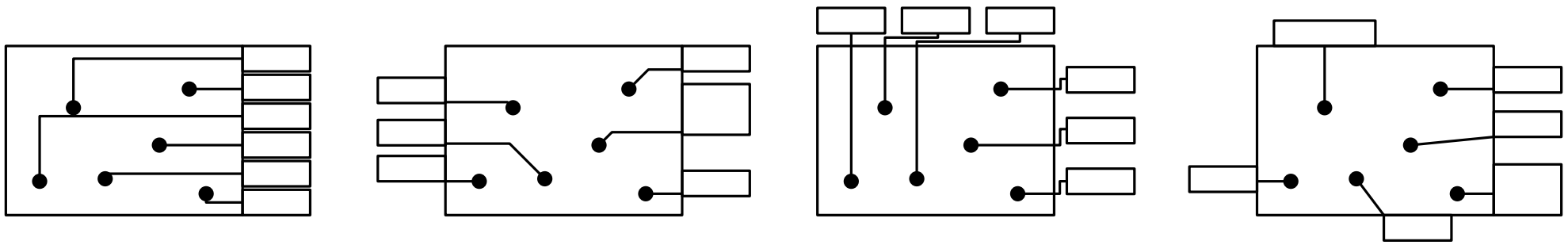
Ges:
- **zulässige Labelplatzierung** am Rand von R
- **zulässiger Leader** zw. jedem Punkt und seinem Label
- optimale **Beschriftungsqualität**

(1) zulässige Labelplatzierung

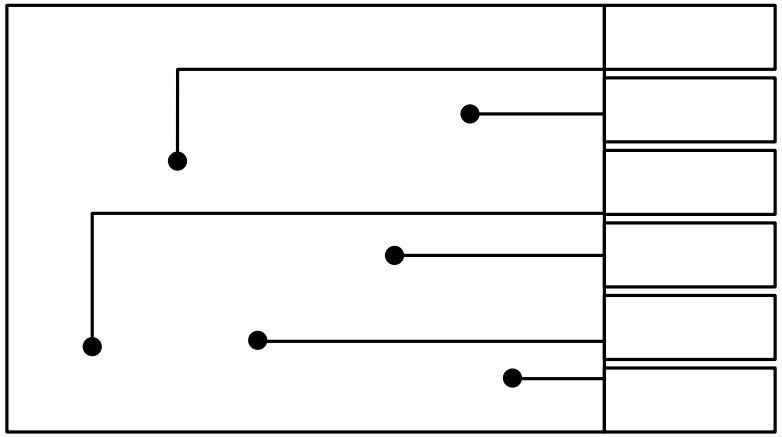
(2) zulässige Leader

(3) Beschriftungsqualität

- minimale Gesamtleaderlänge (= minimum ink)
- minimale Knickzahl
- beliebiger Leader-abhängiger Wert



Teil 1: Allgemeine Bewertungsfunktion

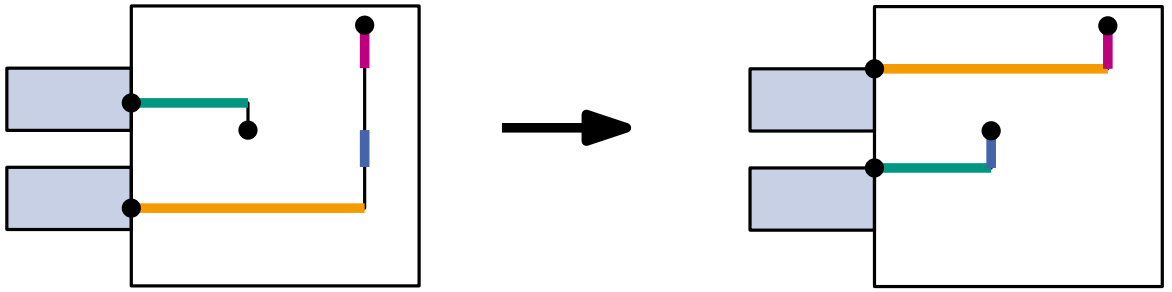


- einseitig
- uniforme Label
- feste Labelpositionen
- kreuzungsfrei
- po-Leader
- sliding Ports
- **allgemeine Bewertungsfunktion**

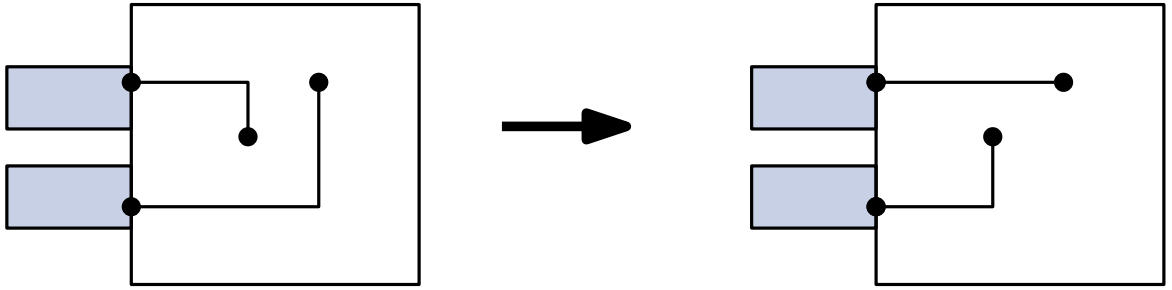
Allgemeine Bewertungsfunktionen

Beispiele aus letzter Vorlesung:

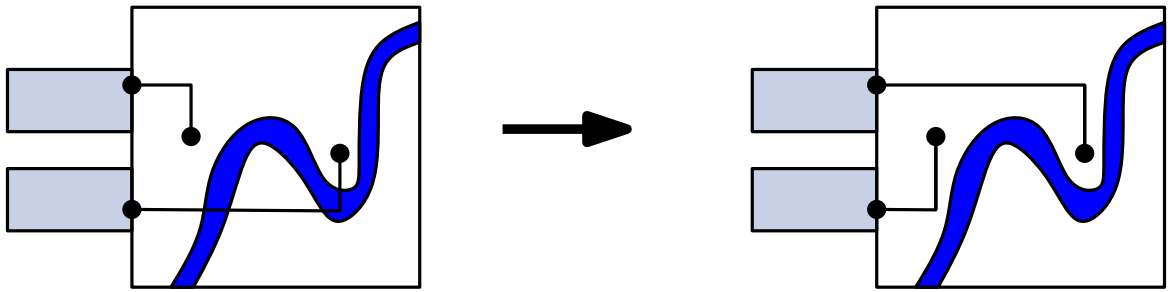
Längenminimierung:



Knickminimierung:



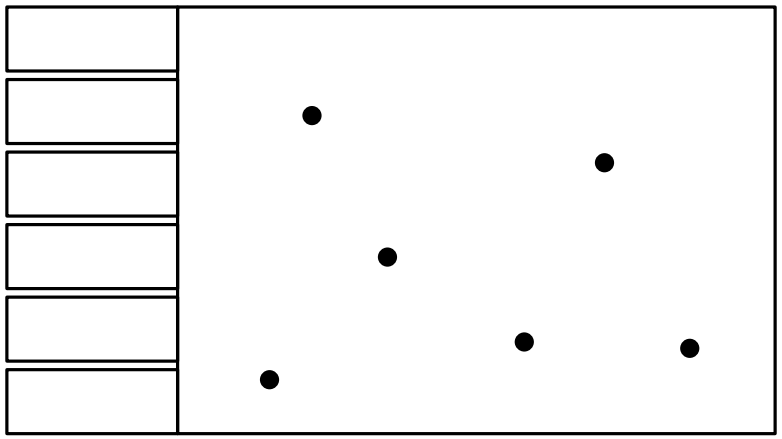
Interferenzminimierung mit Karte



Rekursives Zerlegen

Spezielle Eigenschaften längenminimaler Lösungen ließen sich ausnutzen um einen schnellen Sweep-Line Algorithmus einzusetzen.

Bei allgemeinen (Leader-basierten) Bewertungsfunktionen kann man einen generischen Ansatz verfolgen und versuchen das Problem rekursiv zu zerlegen.

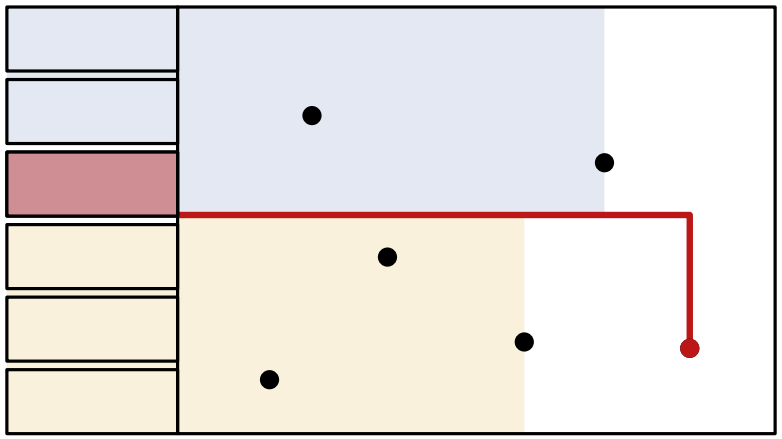


Wie?

Rekursives Zerlegen

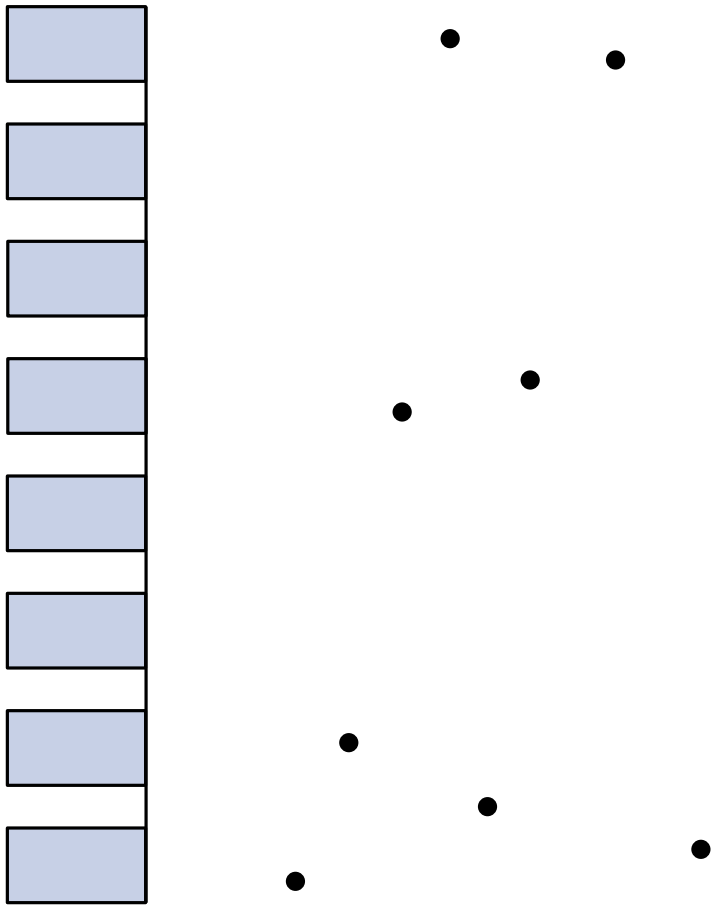
Spezielle Eigenschaften längenminimaler Lösungen ließen sich ausnutzen um einen schnellen Sweep-Line Algorithmus einzusetzen.

Bei allgemeinen (Leader-basierten) Bewertungsfunktionen kann man einen generischen Ansatz verfolgen und versuchen das Problem rekursiv zu zerlegen.

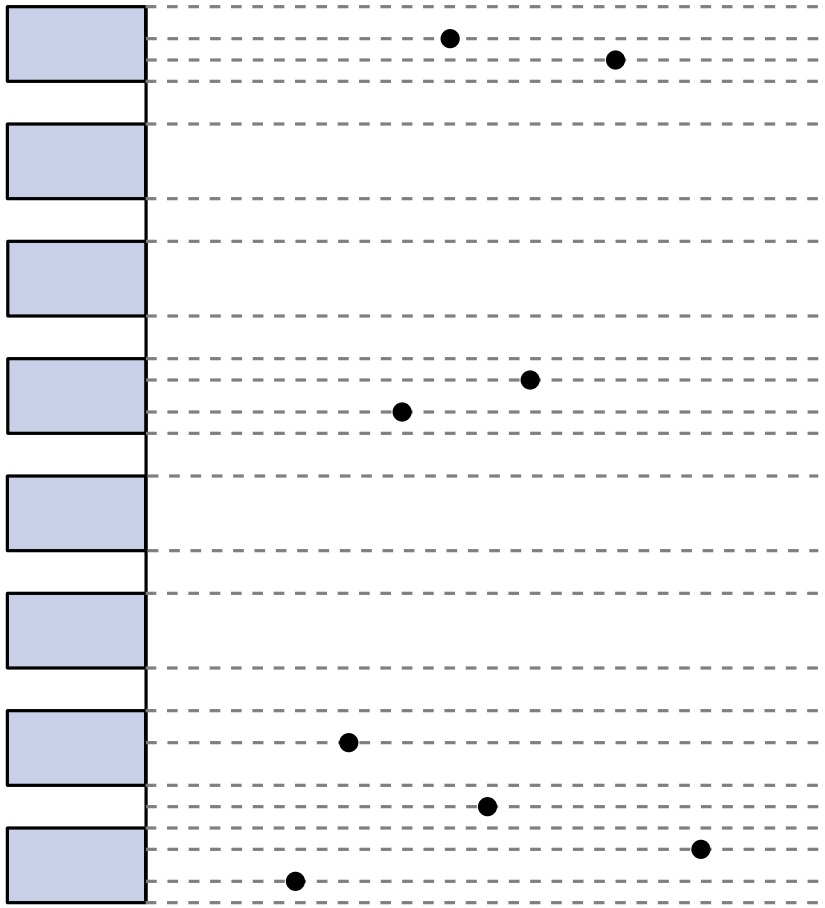


Leader des rechtesten Punktes zerlegt Instanz in zwei unabhängige Teile.

Ein dynamisches Programm (Benkert et al. '09)

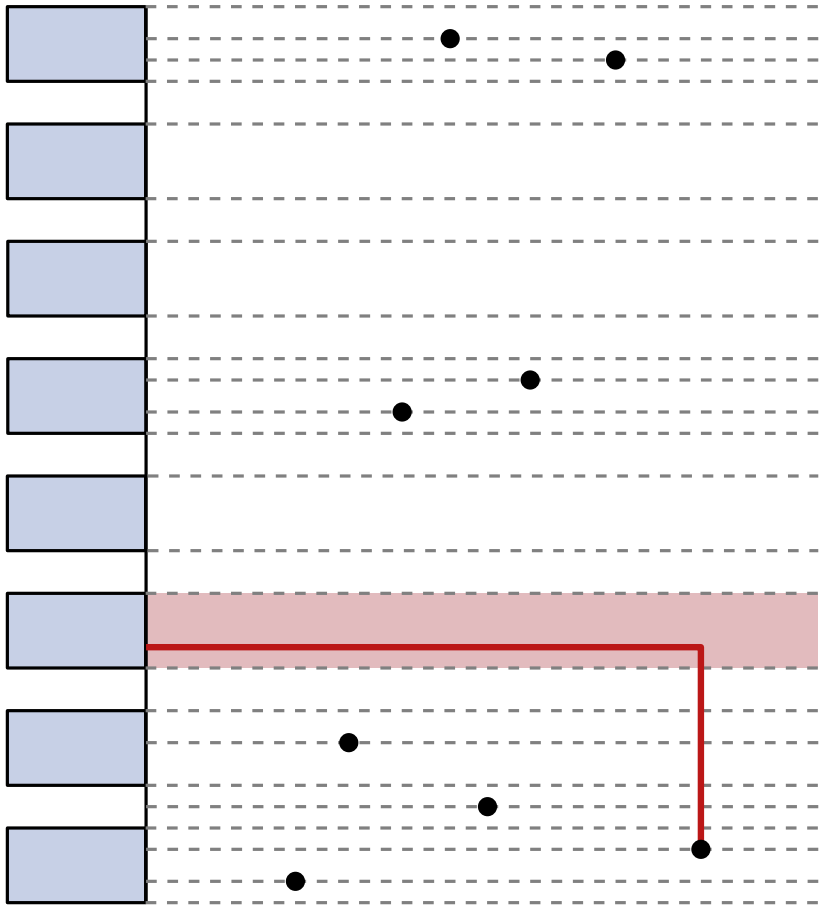


Ein dynamisches Programm (Benkert et al. '09)



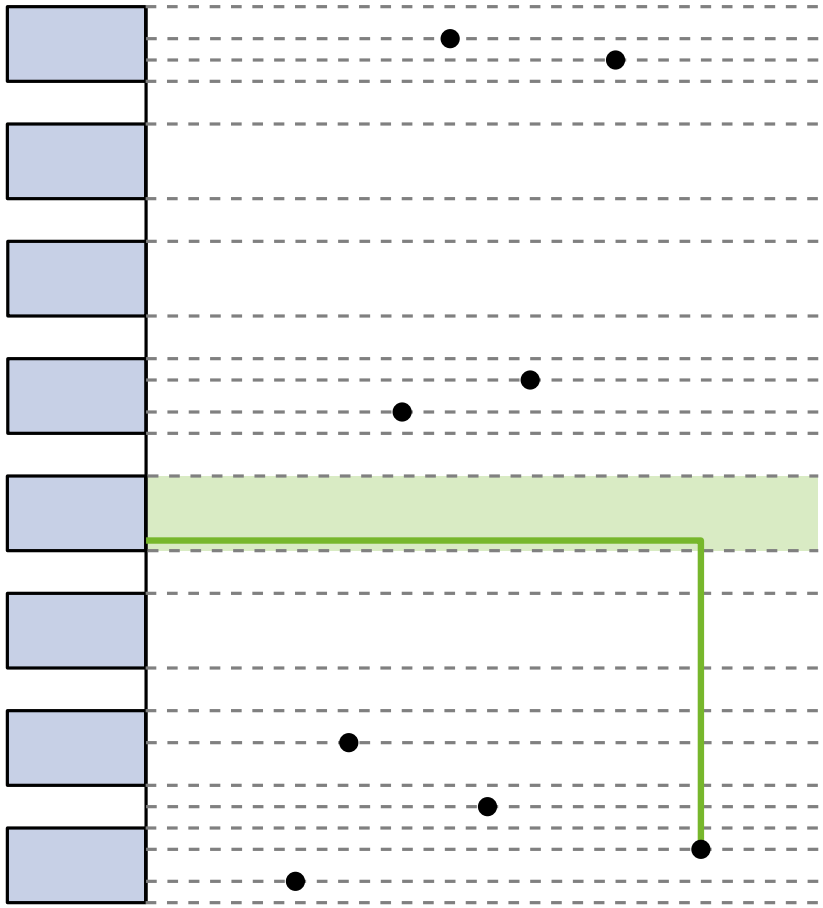
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen

Ein dynamisches Programm (Benkert et al. '09)



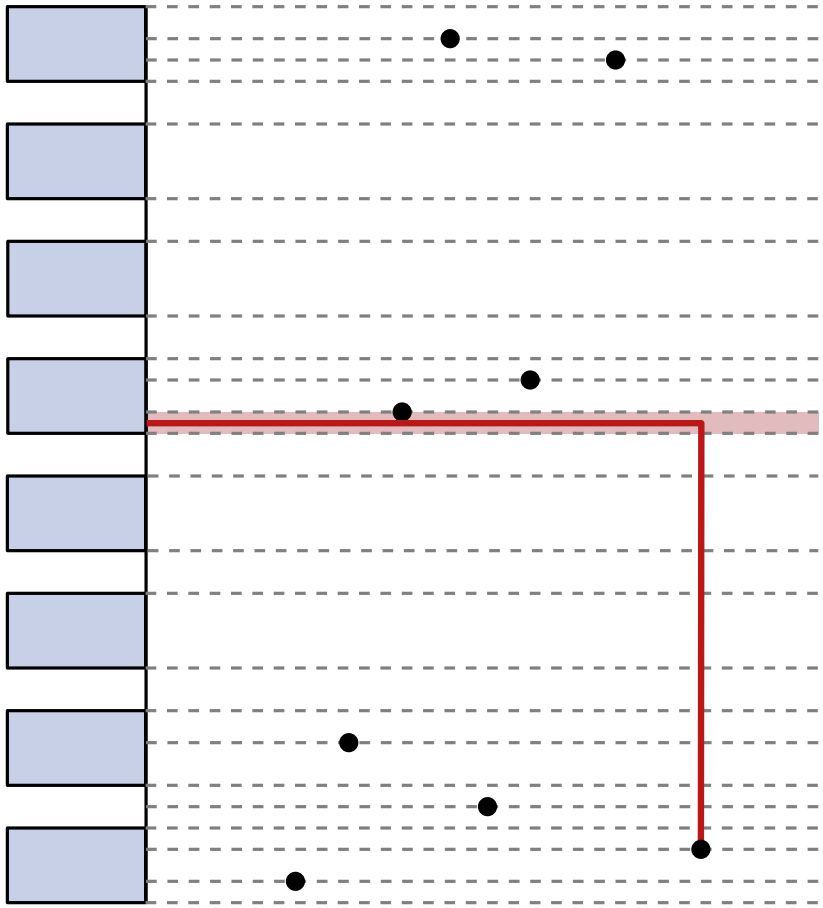
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. '09)



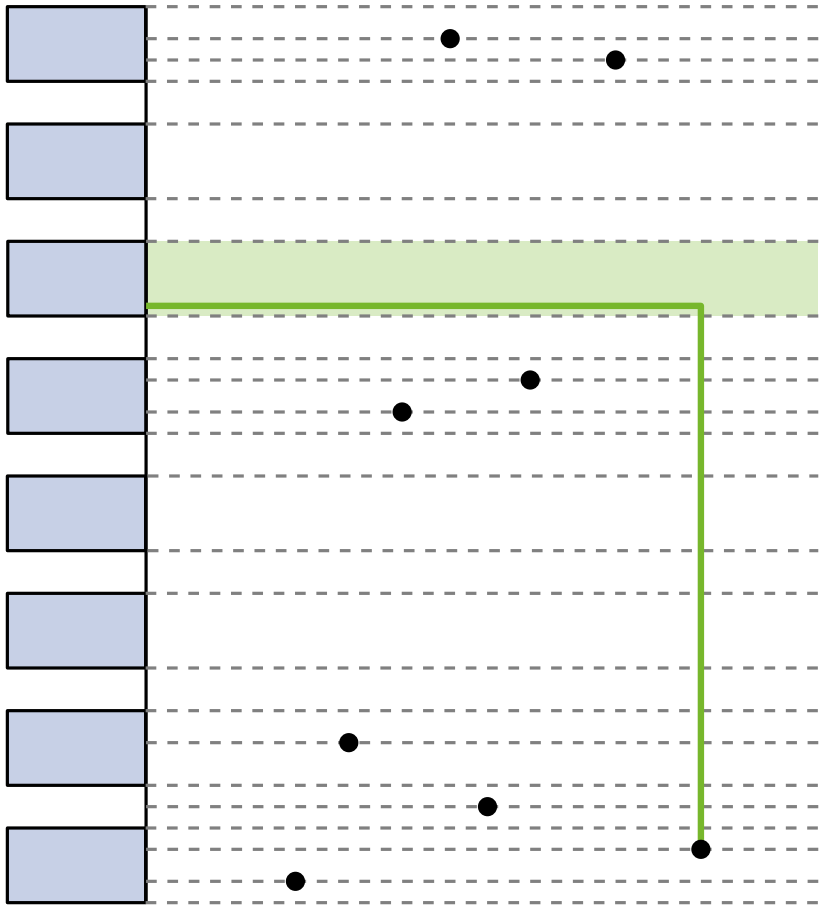
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. '09)



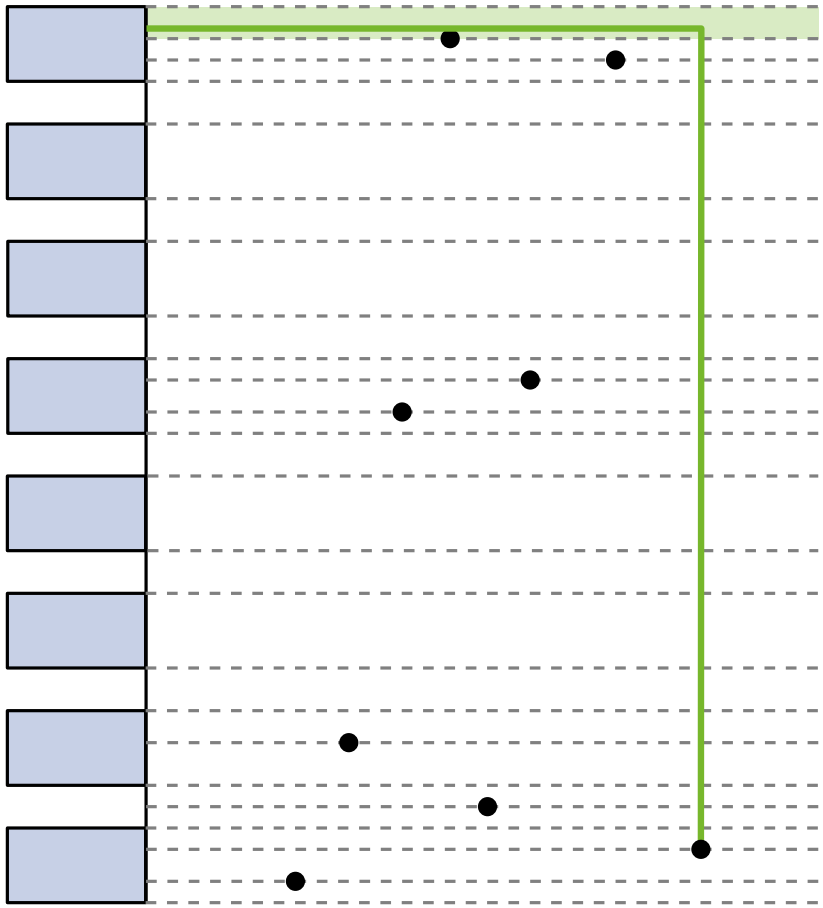
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. '09)



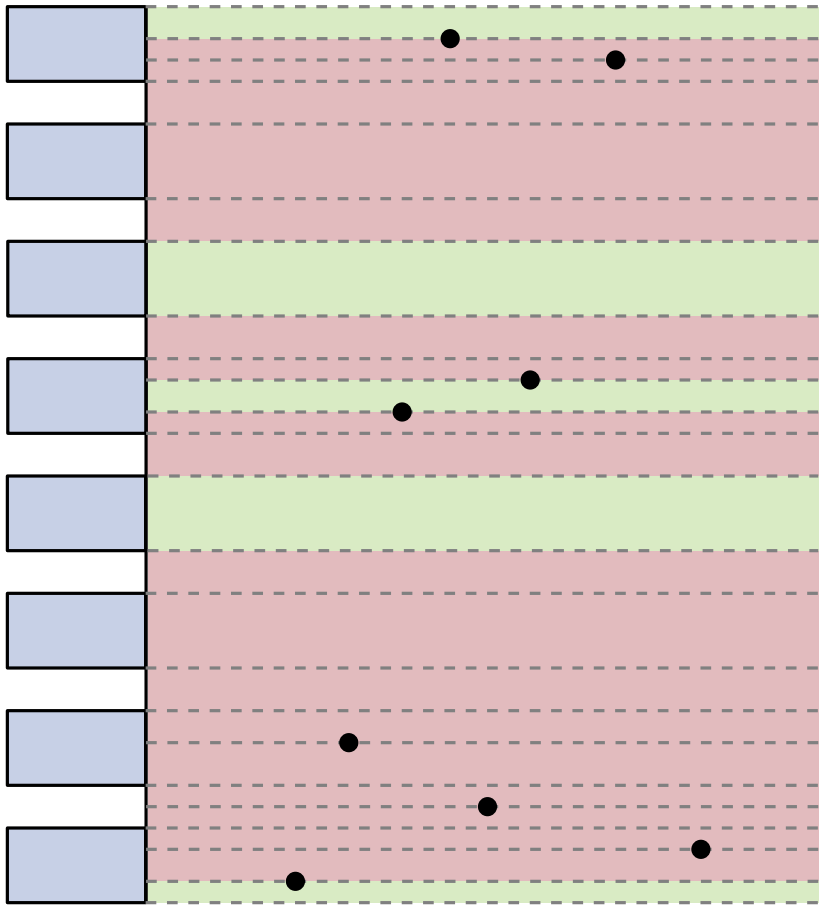
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. '09)



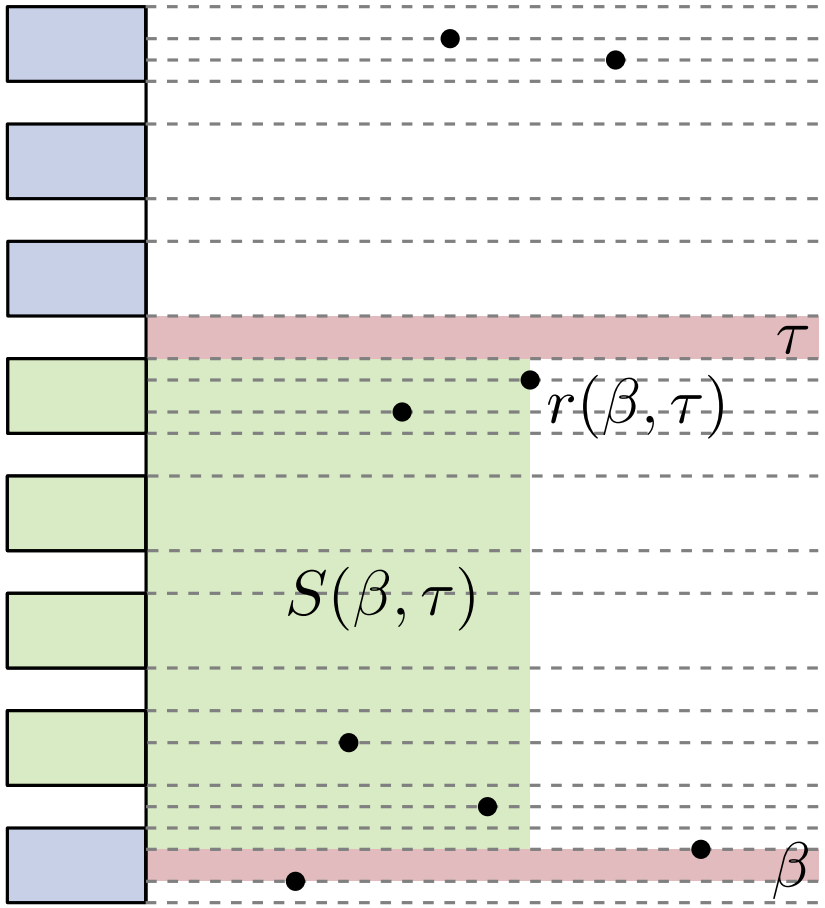
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. '09)



- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r, d.h. # Punkte und # Label in Teilinstanzen passen

Ein dynamisches Programm (Benkert et al. '09)

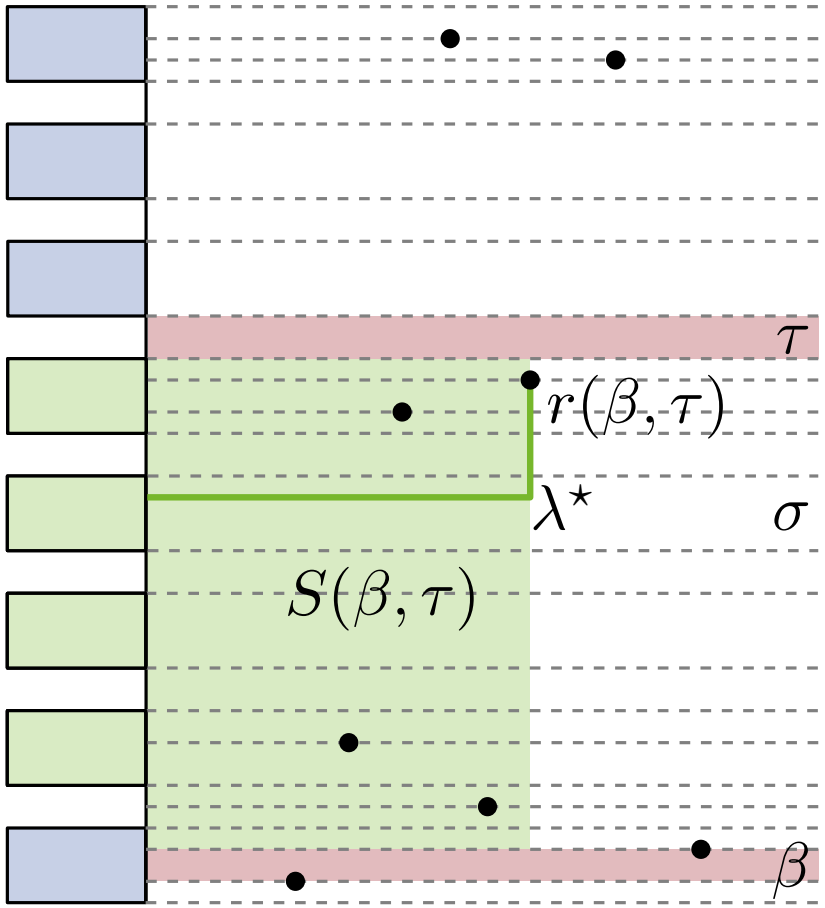


- Labelgrenzen und Punkte induzieren $3n+1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r, d.h. # Punkte und # Label in Teilinstanzen passen

Instanz $S(\beta,\tau)$ für Streifen β und τ mit

- alle Streifen zwischen β und τ
- alle k vollständigen Label
- k linkeste Punkte
- davon rechtester Punkt $r(\beta,\tau)$

Ein dynamisches Programm (Benkert et al. '09)



- Labelgrenzen und Punkte induzieren $3n+1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r, d.h. # Punkte und # Label in Teilinstanzen passen

Instanz $S(\beta,\tau)$ für Streifen β und τ mit

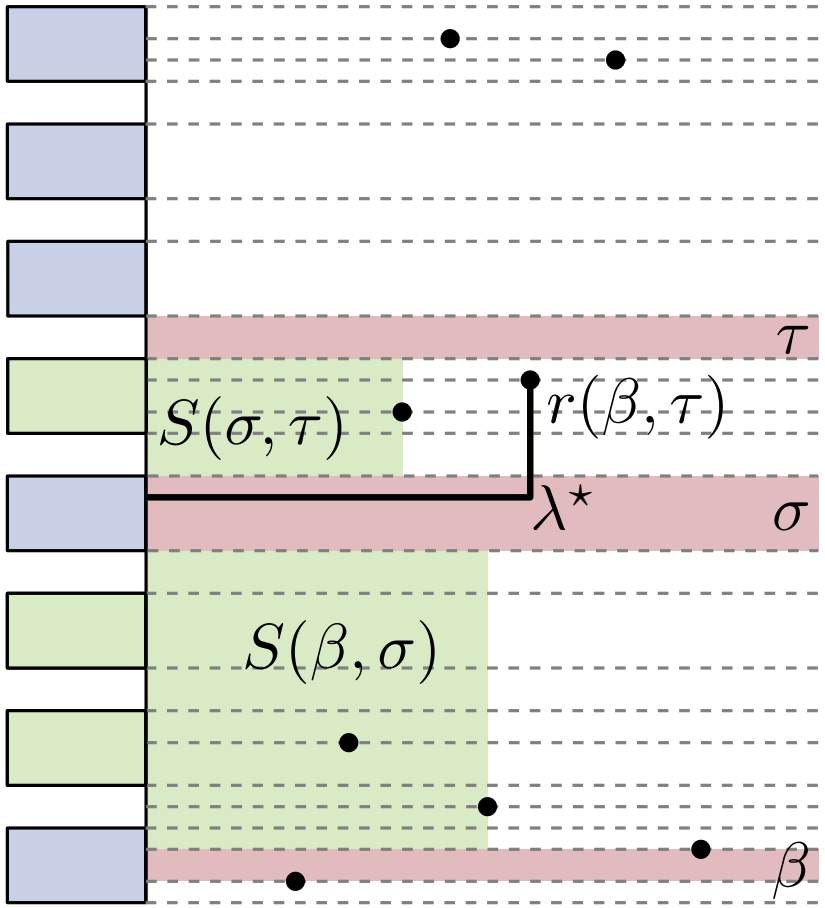
- alle Streifen zwischen β und τ
- alle k vollständigen Label
- k linkeste Punkte
- davon rechtester Punkt $r(\beta,\tau)$

Rekurrenzgleichung:

$$T[\beta,\tau] = \min_{\text{zul. } \sigma \in S(\beta,\tau)} \text{bad}(\lambda^\star(r(\beta,\tau),\sigma)) + T[\beta,\sigma] + T[\sigma,\tau]$$

$\lambda^\star(r,\sigma)$: optimaler Leader für Punkt r in Streifen σ

Ein dynamisches Programm (Benkert et al. '09)



- Labelgrenzen und Punkte induzieren $3n+1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r, d.h. # Punkte und # Label in Teilinstanzen passen

Instanz $S(\beta,\tau)$ für Streifen β und τ mit

- alle Streifen zwischen β und τ
- alle k vollständigen Label
- k linkeste Punkte
- davon rechtester Punkt $r(\beta,\tau)$

Rekurrenzgleichung:

$$T[\beta,\tau] = \min_{\text{zul. } \sigma \in S(\beta,\tau)} \text{bad}(\lambda^\star(r(\beta,\tau),\sigma)) + T[\beta,\sigma] + T[\sigma,\tau]$$

$\lambda^\star(r,\sigma)$: optimaler Leader für Punkt r in Streifen σ

Ergebnis

Satz 1: Gegeben n Punkte in einem Rechteck R und n Label auf einer Seite von R kann in ? Zeit und ? Platz eine zulässige po-Randbeschriftung mit minimalen Kosten berechnet werden.

Rekurrenzgleichung:

$$T[\beta, \tau] = \min_{\text{zul. } \sigma \in S(\beta,\tau)} \mathsf{bad}(\lambda^\star(r(\beta, \tau), \sigma)) + T[\beta, \sigma] + T[\sigma, \tau]$$

$\lambda^\star(r, \sigma)$: optimaler Leader für Punkt r in Streifen σ

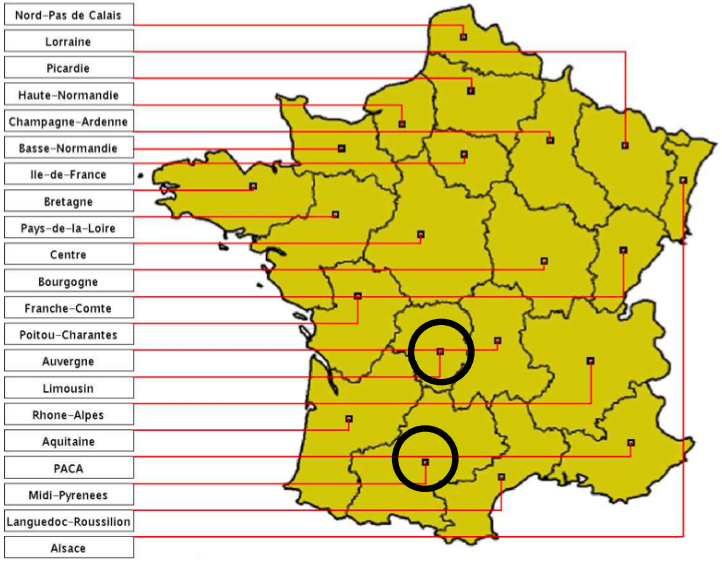
Ergebnis

Satz 1: Gegeben n Punkte in einem Rechteck R und n Label auf einer Seite von R kann in $O(n^3)$ Zeit und $O(n^2)$ Platz eine zulässige po-Randbeschriftung mit minimalen Kosten berechnet werden.

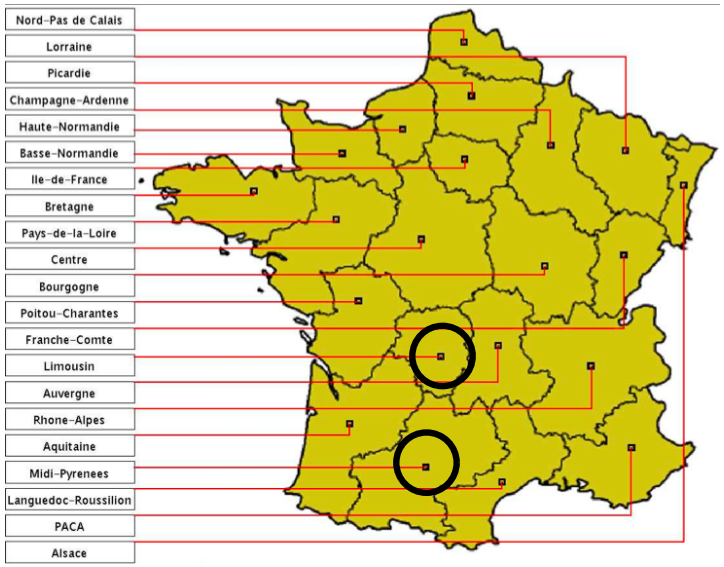
Ergebnis

Satz 1: Gegeben n Punkte in einem Rechteck R und n Label auf einer Seite von R kann in $O(n^3)$ Zeit und $O(n^2)$ Platz eine zulässige po-Randbeschriftung mit minimalen Kosten berechnet werden.

Beispiel & Demo



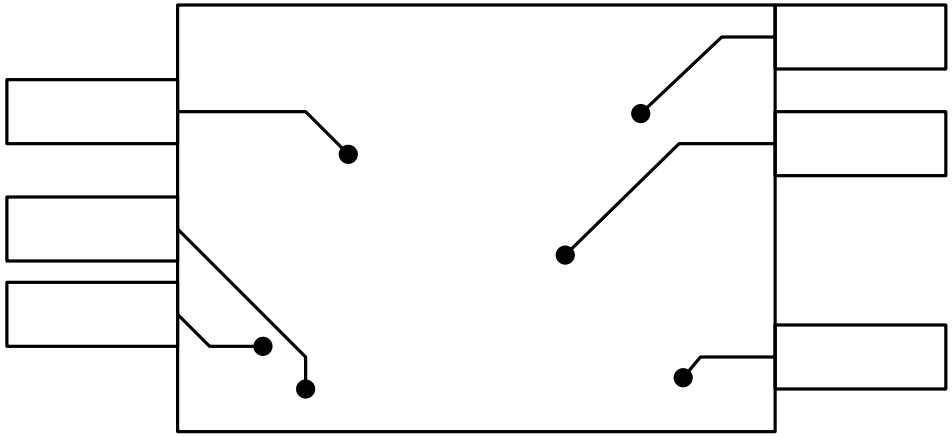
Längenminimierung



Länge & Leader-Punkt-Abstände

Java-Applet s. `i11www.iti.kit.edu/extra/labeling`

Teil 2: Längenminimierung allgemein



- k-seitig
- uniforme Label
- feste Labelpositionen
- kreuzungsfrei
- verschiedene Leadertypen
- fixed Ports
- Längenminimierung

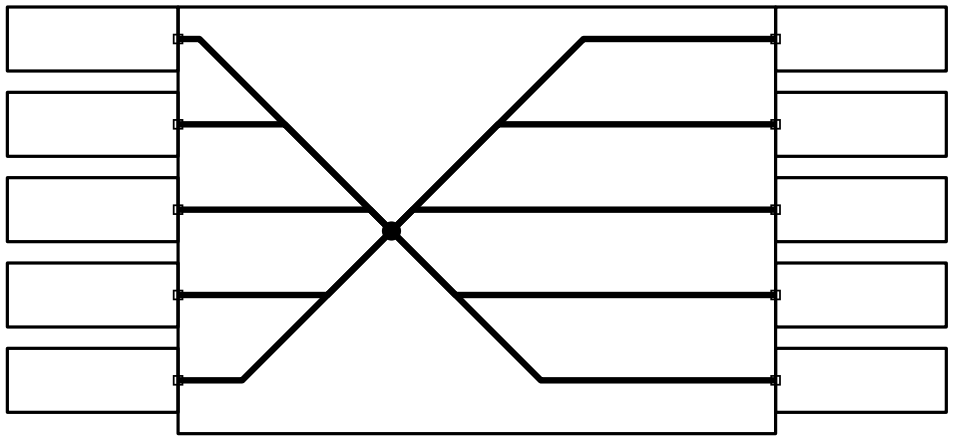
Generischer Algorithmus (Bekos et al. '10)

A) erstelle gewichteten vollständigen bipartiten Graphen $G = (P \cup L, E, w)$ für Punktmenge P, Labelmenge L, Punkt-Label-Kanten E und Kantenlängenfunktion w
B) bestimme bipartites Matching mit minimalem Gewicht in G
C) transformiere Matching in längenminimale Beschriftung
D) löse Kreuzungen längenneutral auf

Generischer Algorithmus (Bekos et al. '10)

A) erstelle gewichteten vollständigen bipartiten Graphen $G = (P \cup L, E, w)$ für Punktmenge P, Labelmenge L, Punkt-Label-Kanten E und Kantenlängenfunktion w
B) bestimme bipartites Matching mit minimalem Gewicht in G
C) transformiere Matching in längenminimale Beschriftung
D) löse Kreuzungen längenneutral auf

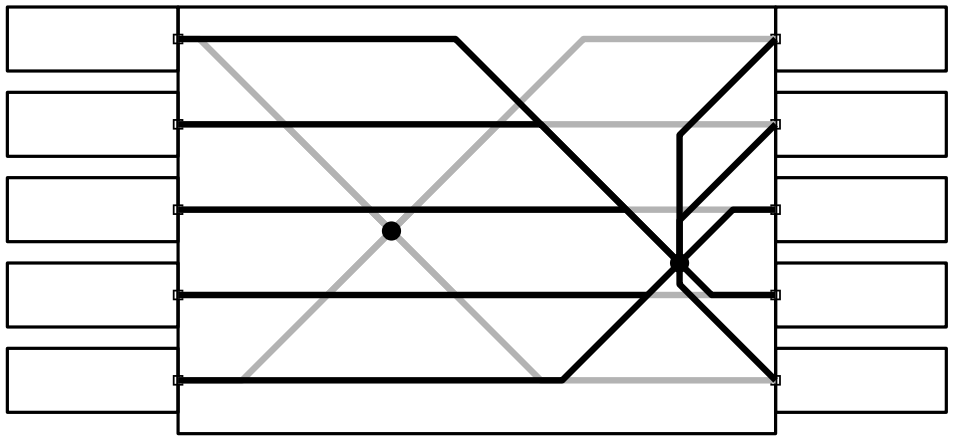
Beispiel: 2-seitig, do- und pd-Leader



Generischer Algorithmus (Bekos et al. '10)

A) erstelle gewichteten vollständigen bipartiten Graphen $G = (P \cup L, E, w)$ für Punktmenge P, Labelmenge L, Punkt-Label-Kanten E und Kantenlängenfunktion w
B) bestimme bipartites Matching mit minimalem Gewicht in G
C) transformiere Matching in längenminimale Beschriftung
D) löse Kreuzungen längenneutral auf

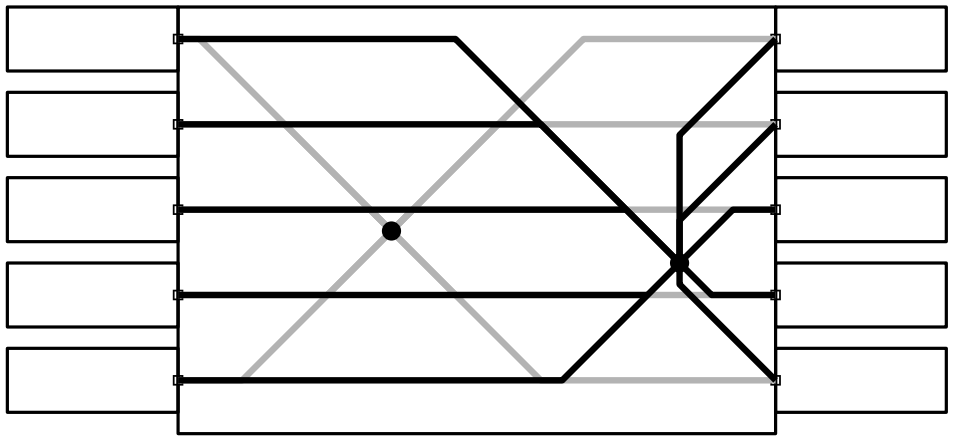
Beispiel: 2-seitig, do- und pd-Leader



Generischer Algorithmus (Bekos et al. '10)

A) erstelle gewichteten vollständigen bipartiten Graphen $G = (P \cup L, E, w)$ für Punktmenge P, Labelmenge L, Punkt-Label-Kanten E und Kantenlängenfunktion w
B) bestimme bipartites Matching mit minimalem Gewicht in G
C) transformiere Matching in längenminimale Beschriftung
D) löse Kreuzungen längenneutral auf

Beispiel: 2-seitig, do- und pd-Leader



Schritte A–C:

- Matching garantiert gültige Beschriftung
- Matchinggewicht entspricht Gesamtleaderlänge
- Schritt B in $O(n^3)$ Zeit [Kuhn '55]
- Kreuzungen werden ignoriert

Eigenschaften

Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

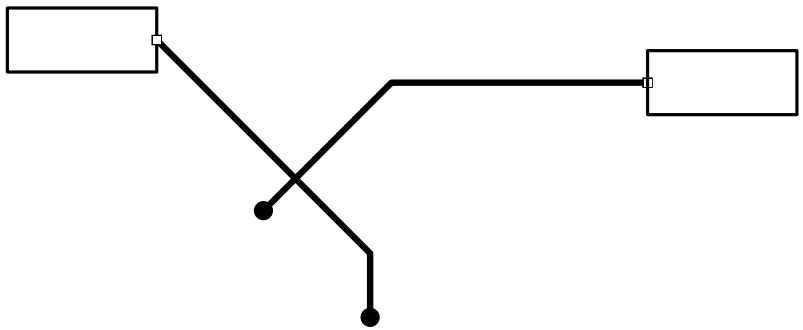
Eigenschaften

Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

Beweis:

1.) Fallunterscheidung, z.B.



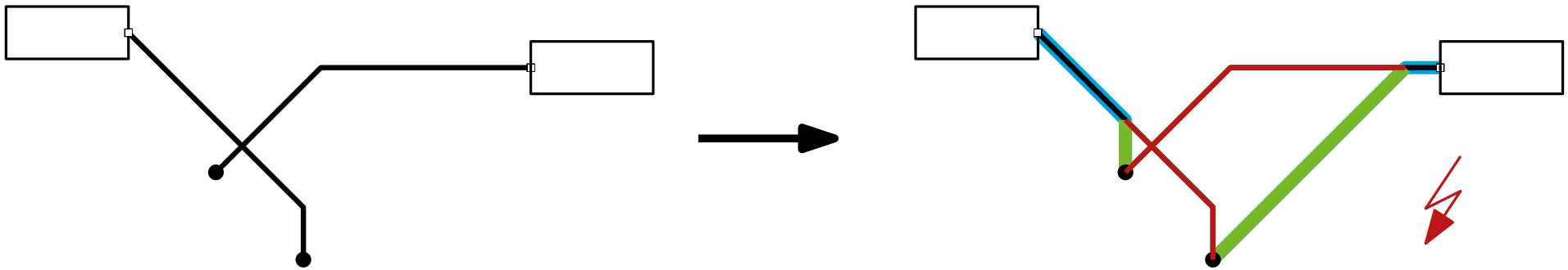
Eigenschaften

Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

Beweis:

1.) Fallunterscheidung, z.B.



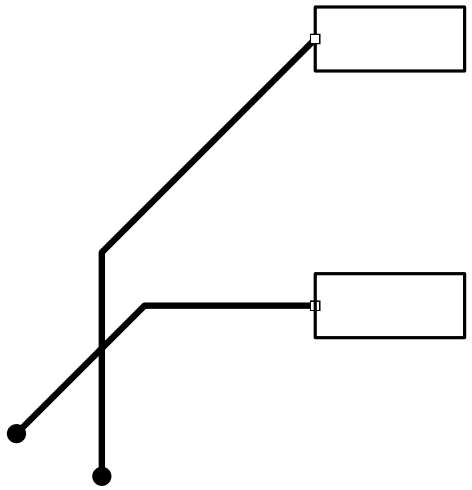
Eigenschaften

Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

Beweis:

2.) Fallunterscheidung, z.B.



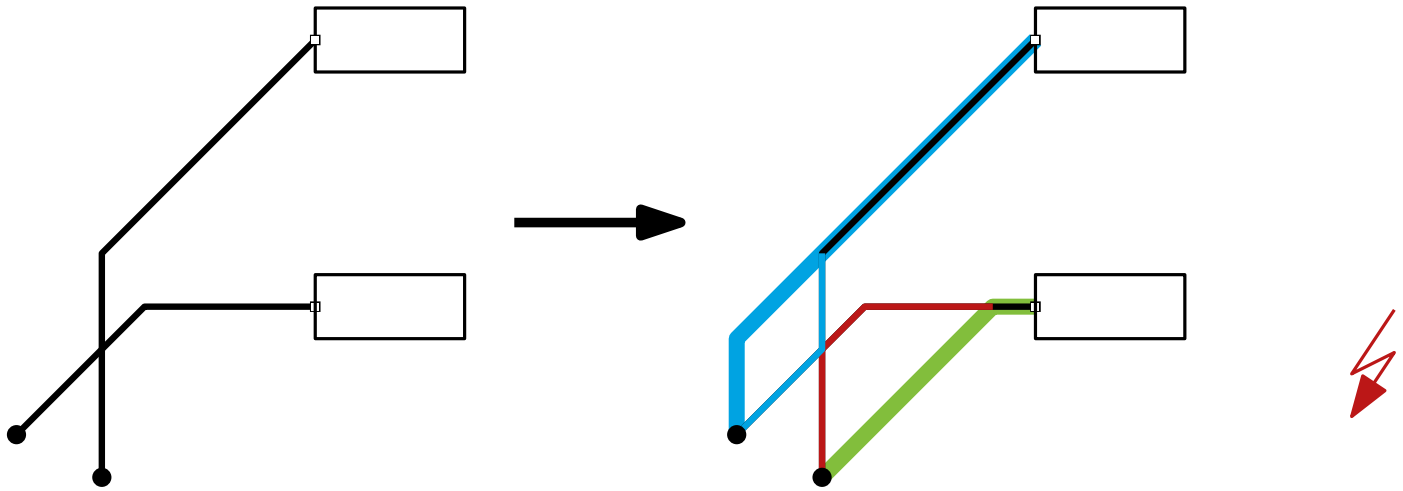
Eigenschaften

Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

Beweis:

2.) Fallunterscheidung, z.B.



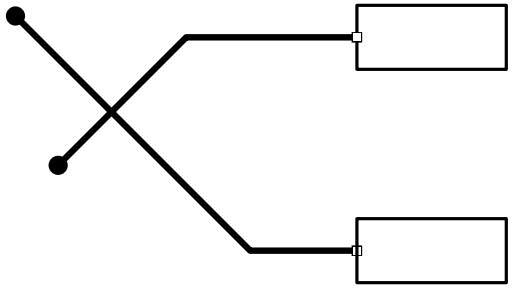
Eigenschaften

Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

Beweis:

3.) Fallunterscheidung, z.B.



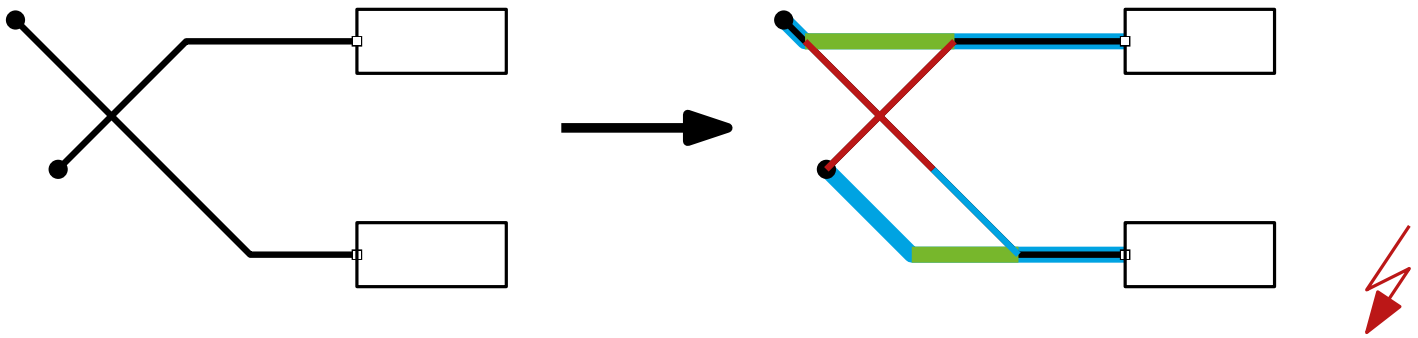
Eigenschaften

Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

Beweis:

3.) Fallunterscheidung, z.B.



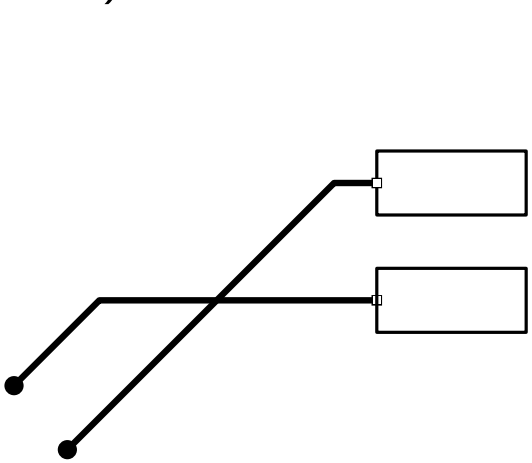
Eigenschaften

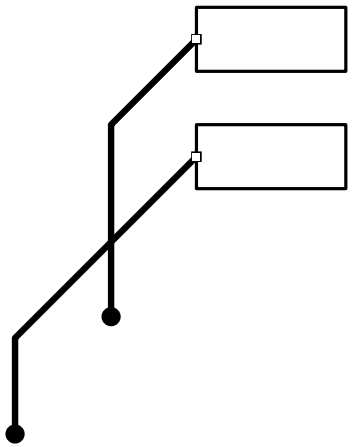
Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

Beweis:

4.) Fallunterscheidung





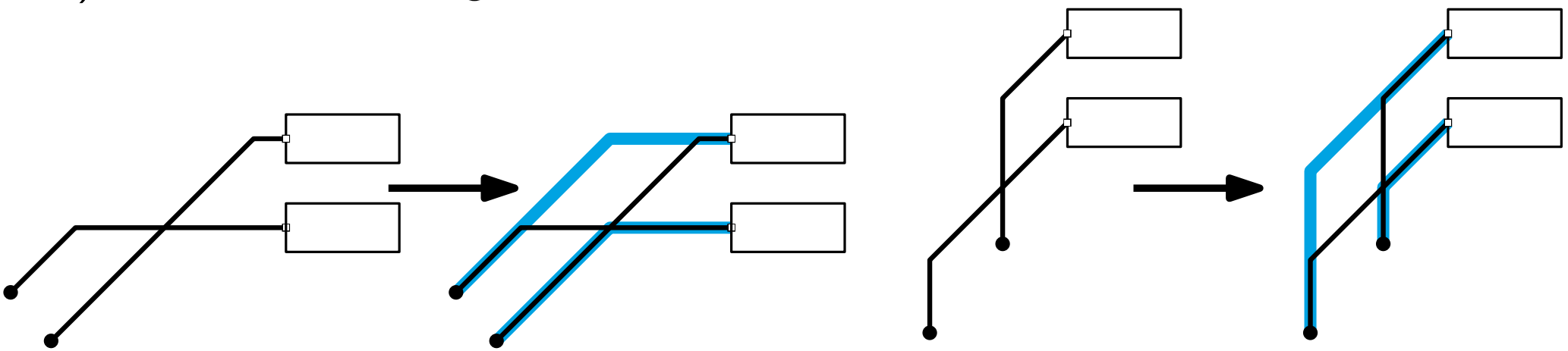
Eigenschaften

Lemma 1: Sei M die Beschriftung aus Schritt C des Algorithmus. Dann gilt für sich kreuzende Leader

1. beide Leader gehen zur gleichen Seite von R
2. beide Leader sind vom gleichen Typ
3. beide Leader zeigen in die gleiche Richtung
4. durch Tauschen der Zuordnung löst sich die Kreuzung auf, die Gesamtlänge bleibt gleich, Typ und Orientierung bleiben gleich.

Beweis:

4.) Fallunterscheidung



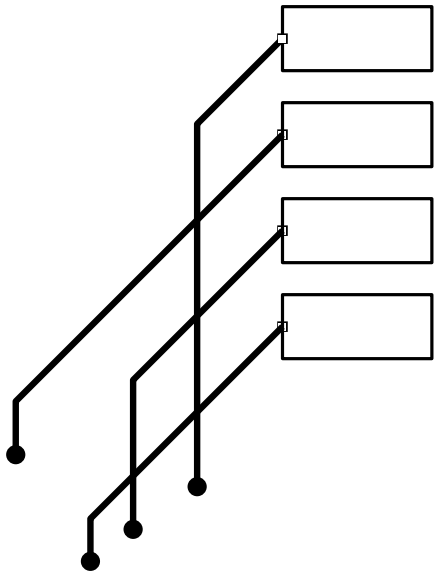
Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

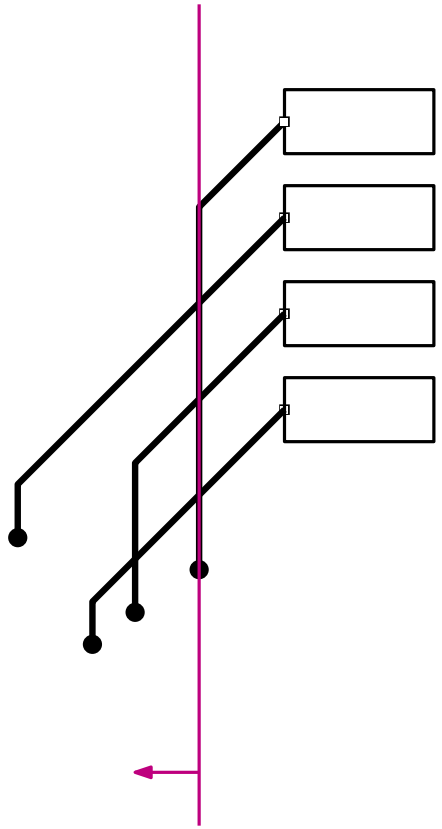
Beweisskizze:



Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

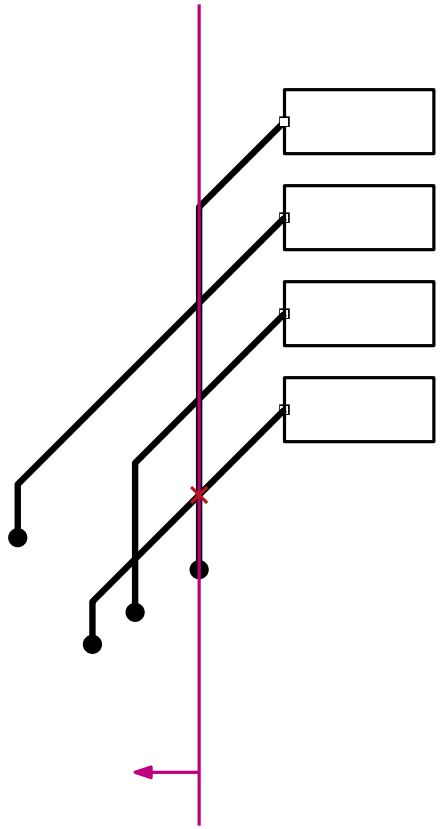
Beweisskizze:



Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

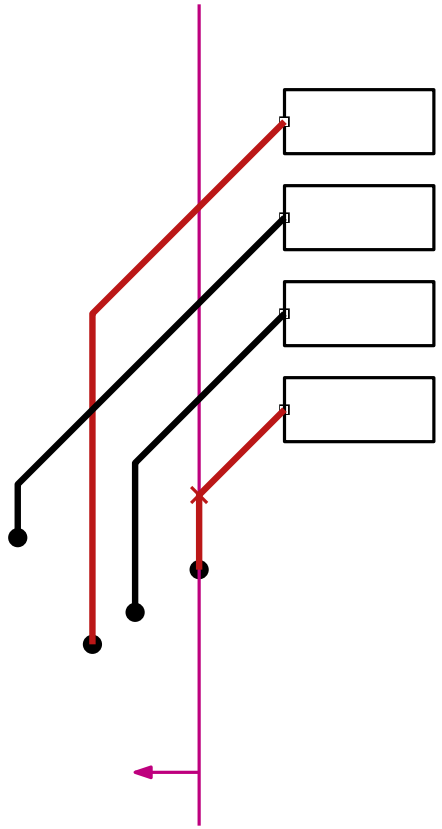
Beweisskizze:



Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

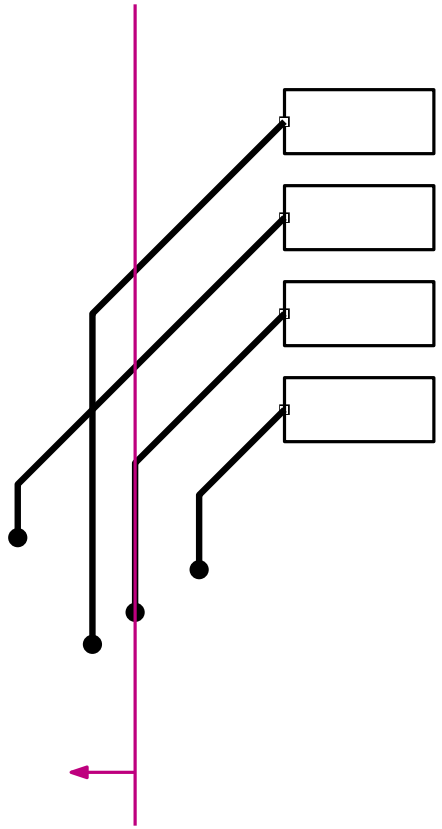
Beweisskizze:



Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

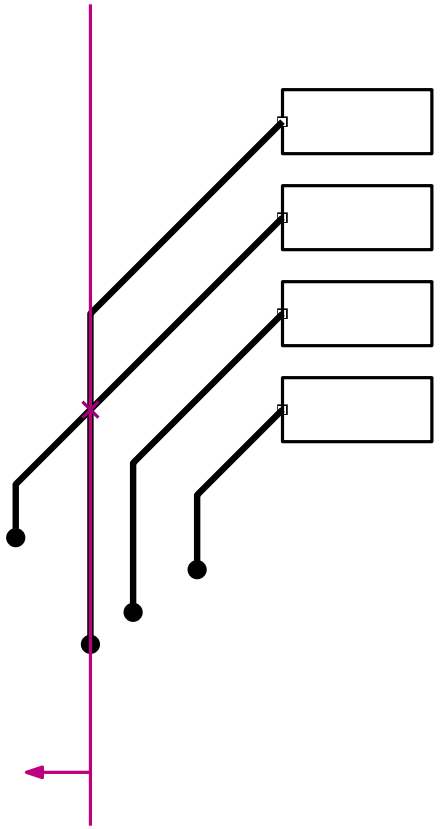
Beweisskizze:



Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

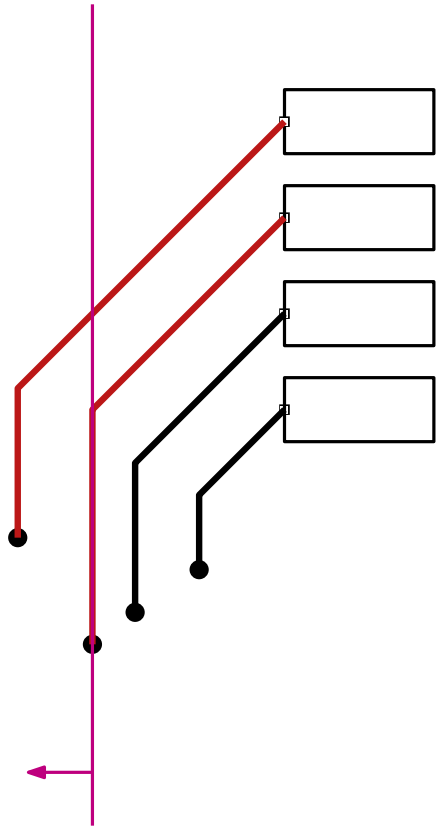
Beweisskizze:



Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

Beweisskizze:



Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

Satz 2: Gegeben n Punkte in einem Rechteck R und n Label uniformer Größe auf zwei gegenüberliegenden Seiten von R kann in $O(n^3)$ Zeit eine längenminimale do-/pd-Beschriftung bestimmt werden.

Kreuzungsauflösung

Lemma 2: Schritt D ist korrekt, d.h. die Kreuzungen der Beschriftung aus Schritt C lassen sich alle auflösen, und dies benötigt $O(n^2)$ Zeit.

Satz 2: Gegeben n Punkte in einem Rechteck R und n Label uniformer Größe auf zwei gegenüberliegenden Seiten von R kann in $O(n^3)$ Zeit eine längenminimale do-/pd-Beschriftung bestimmt werden.

ohne Beweis:

Satz 3: Gegeben n Punkte in einem Rechteck R und n Label uniformer Größe auf allen vier Seiten von R kann in $O(n^3)$ Zeit eine längenminimale **od-/pd-**Beschriftung bestimmt werden.