Vorlesung Algorithmische Kartografie

Randbeschriftungen Teil 2

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

Benjamin Niedermann · Martin Nöllenburg

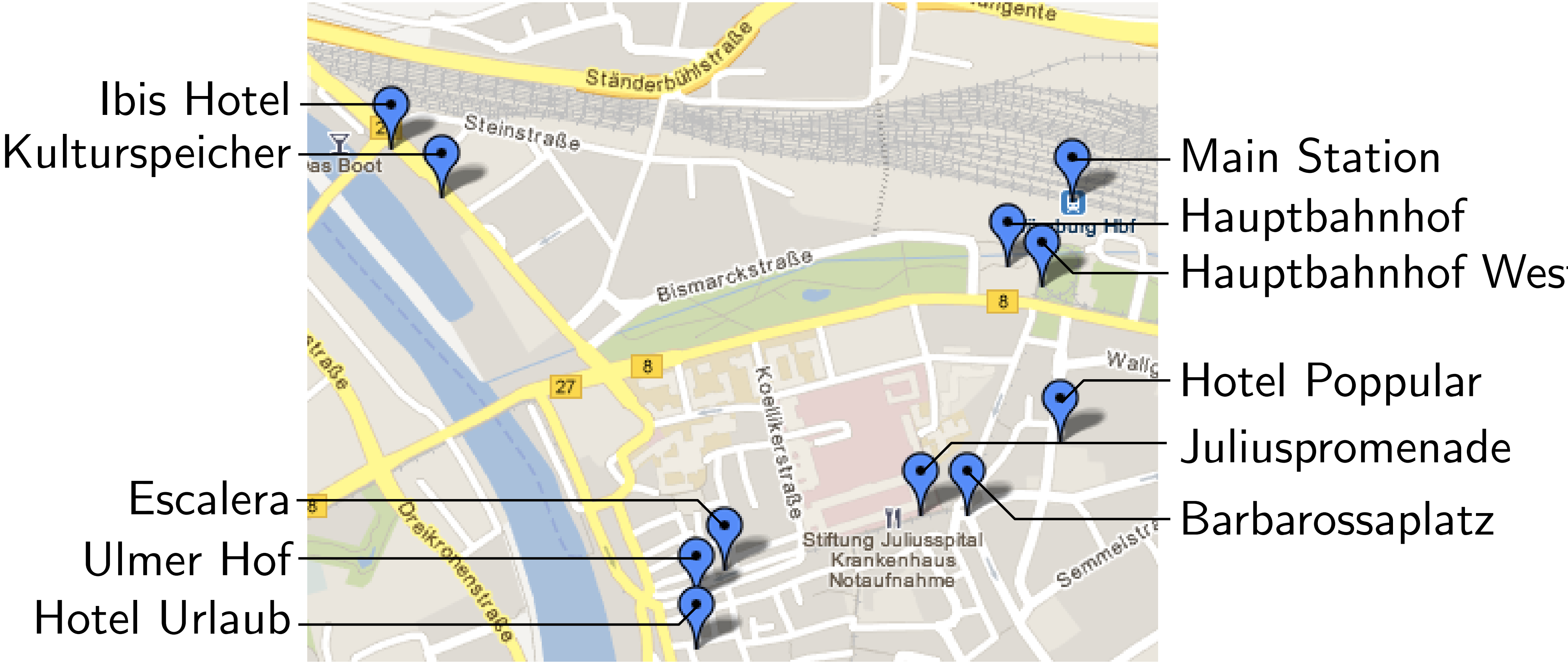
25.06.2015

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.)



Vorteile von Randbeschriftungen

- dichte Punktmengen lassen sich beschriften
- weniger Überdeckungen der Karte
- schnelleres Finden Name $\rightarrow$ 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**

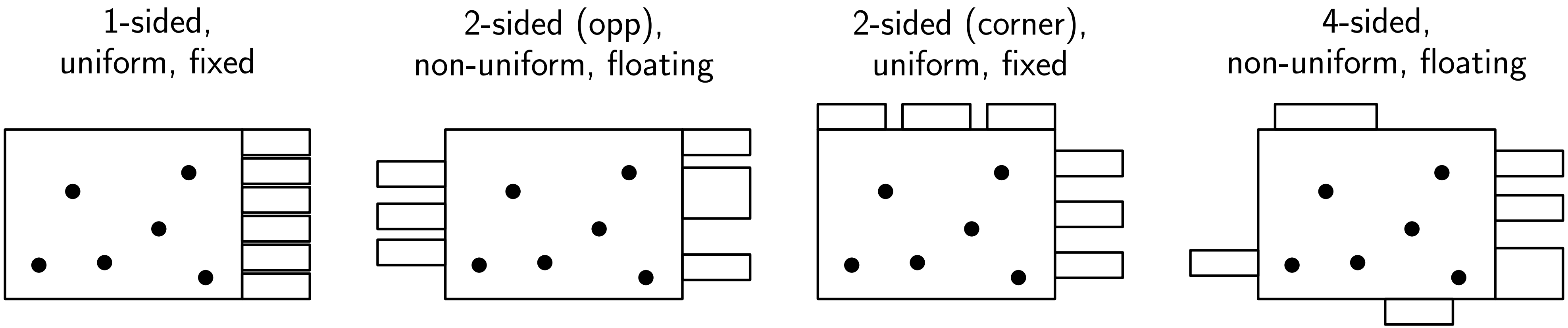
Das Randbeschriftungs-Problem

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

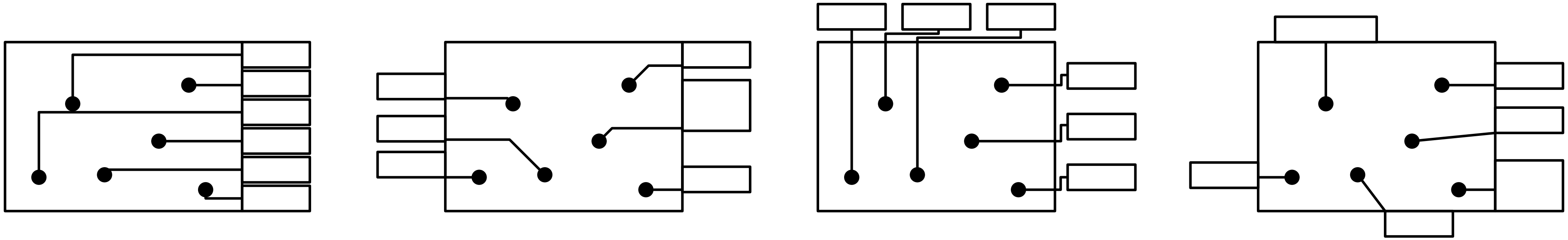
Geg: n Punkte in Rechteck R und für jeden Punkt ein Label repräsentiert durch seine bounding box

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

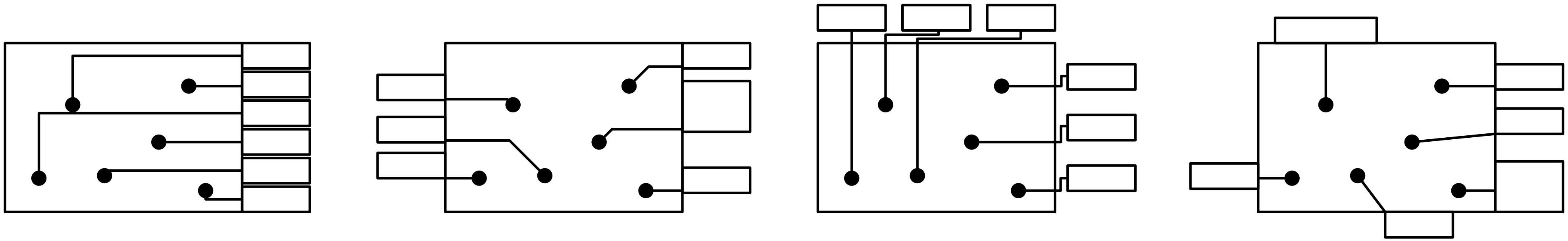
Ges:
- **zulässige Labelplatzierung** am Rand von R
- **zulässiger Leader** zw. jedem Punkt und seinem Label
- optimale **Beschriftungsqualität**

(1) zulässige Labelplatzierung

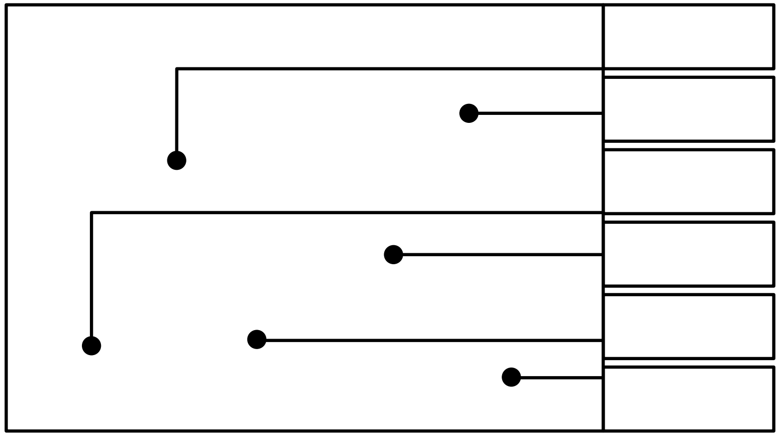
(2) zulässige Leader

(3) Beschriftungsqualität

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



Teil 1: Allgemeine Bewertungsfunktion

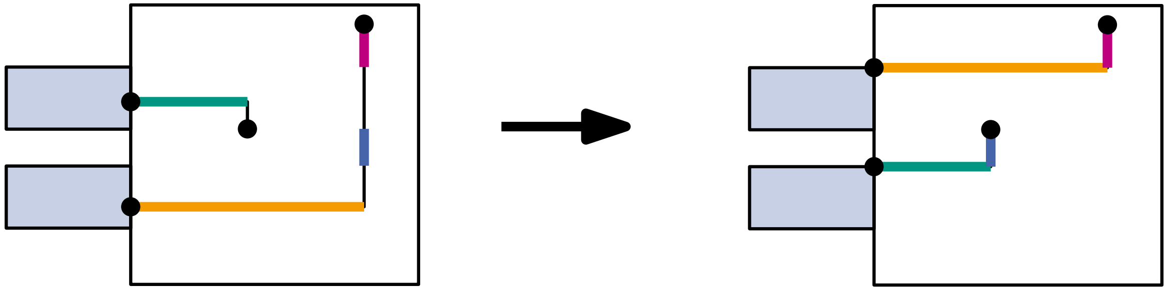


- einseitig
- uniforme Label
- feste Labelpositionen
- kreuzungsfrei
- po-Leader
- sliding Ports
- **allgemeine Bewertungsfunktion**

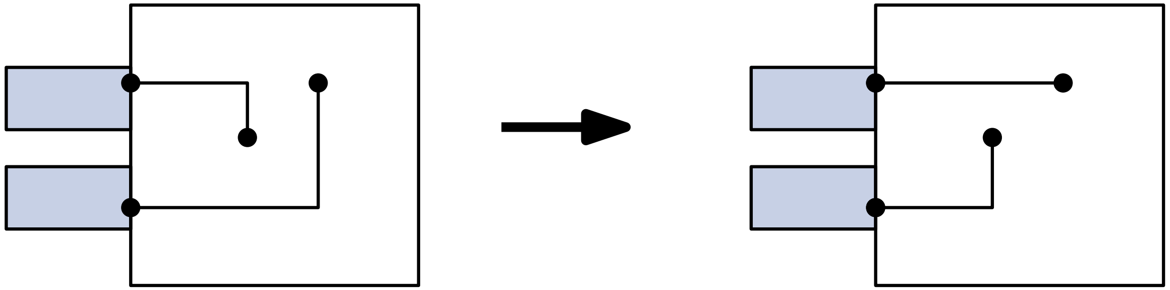
Allgemeine Bewertungsfunktionen

Beispiele aus letzter Vorlesung:

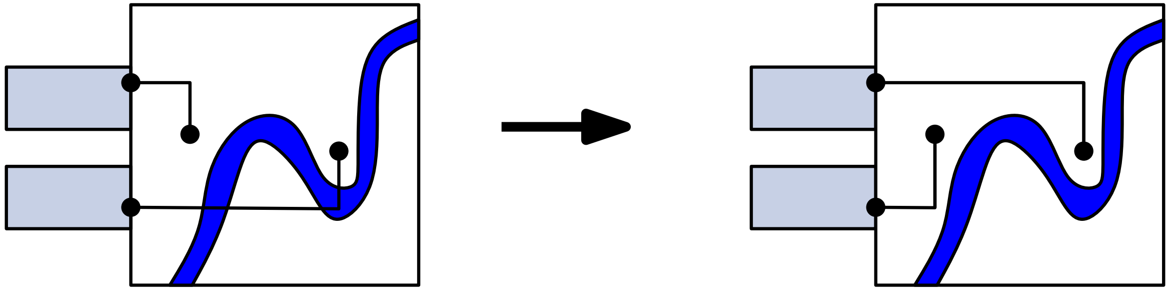
Längenminimierung:



Knickminimierung:



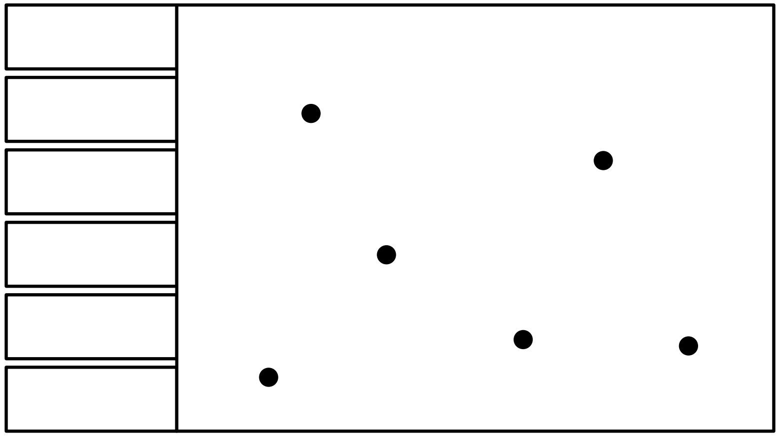
Interferenzminimierung mit Karte



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.

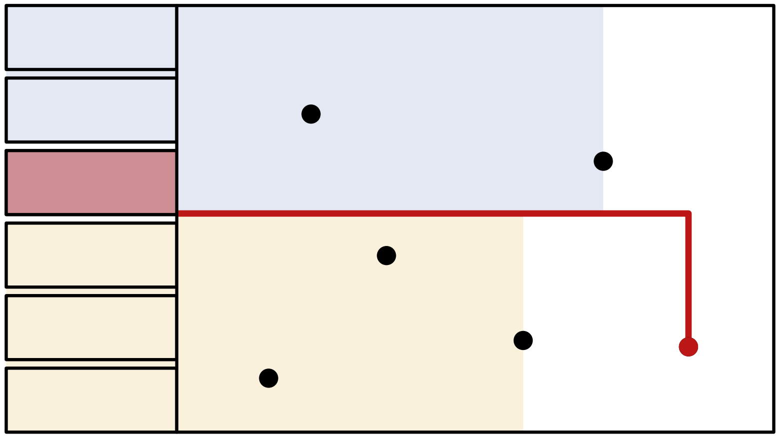


Wie?

Rekursives Zerlegen

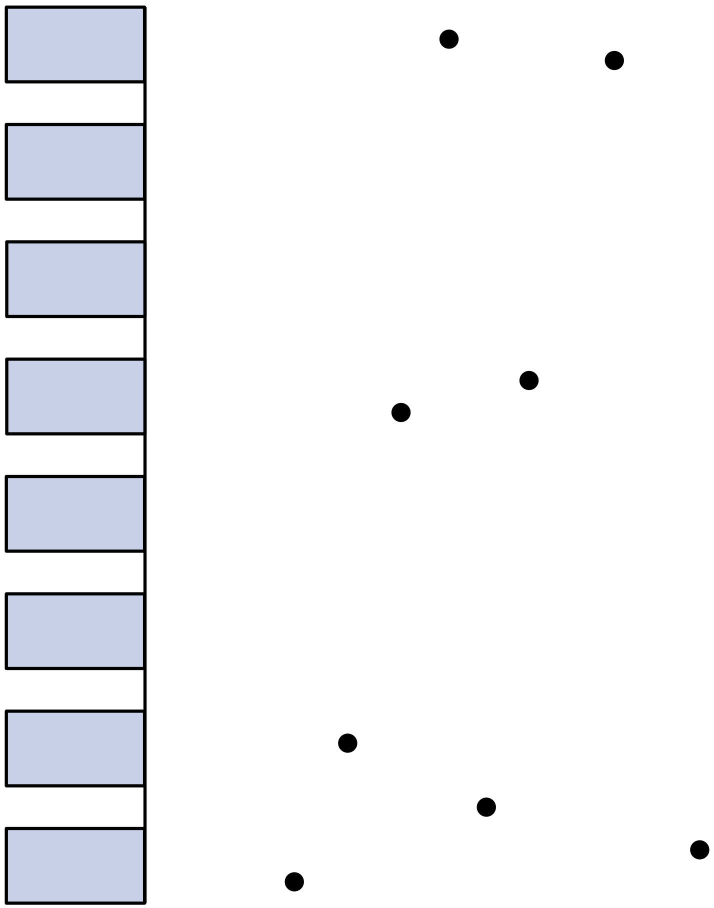
Spezielle Eigenschaften längenminimaler Lösungen ließen sich ausnutzen um einen schnellen Sweep-Line Algorithmus einzusetzen.

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

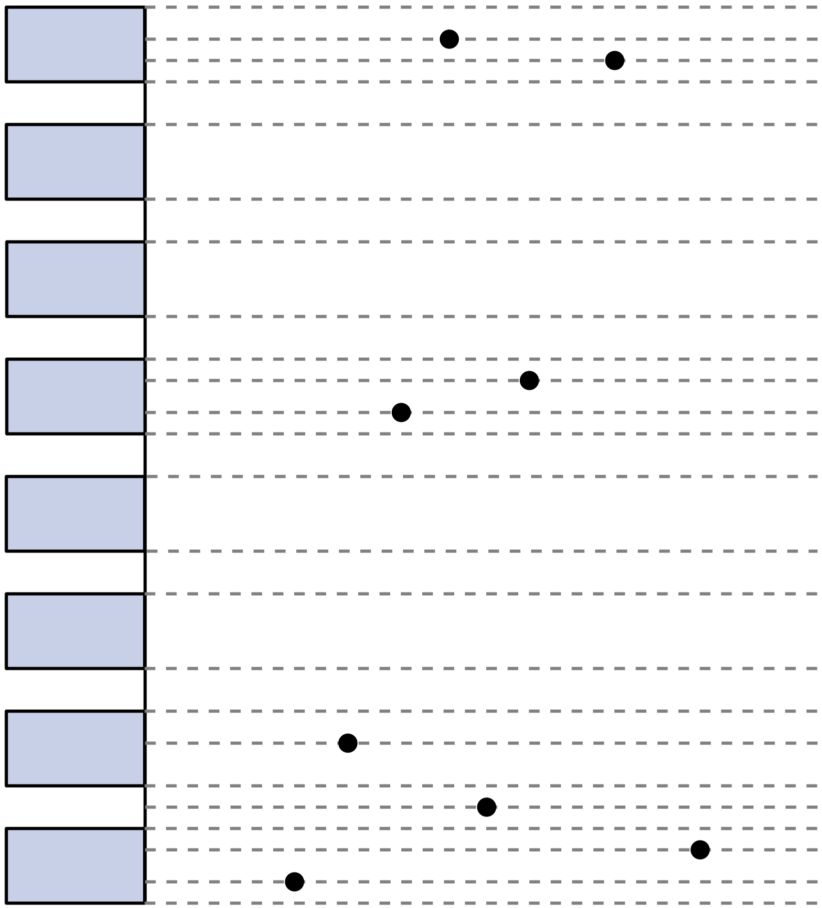


Leader des rechtesten Punktes zerlegt Instanz in zwei unabhängige Teile.

Ein dynamisches Programm (Benkert et al. '09)

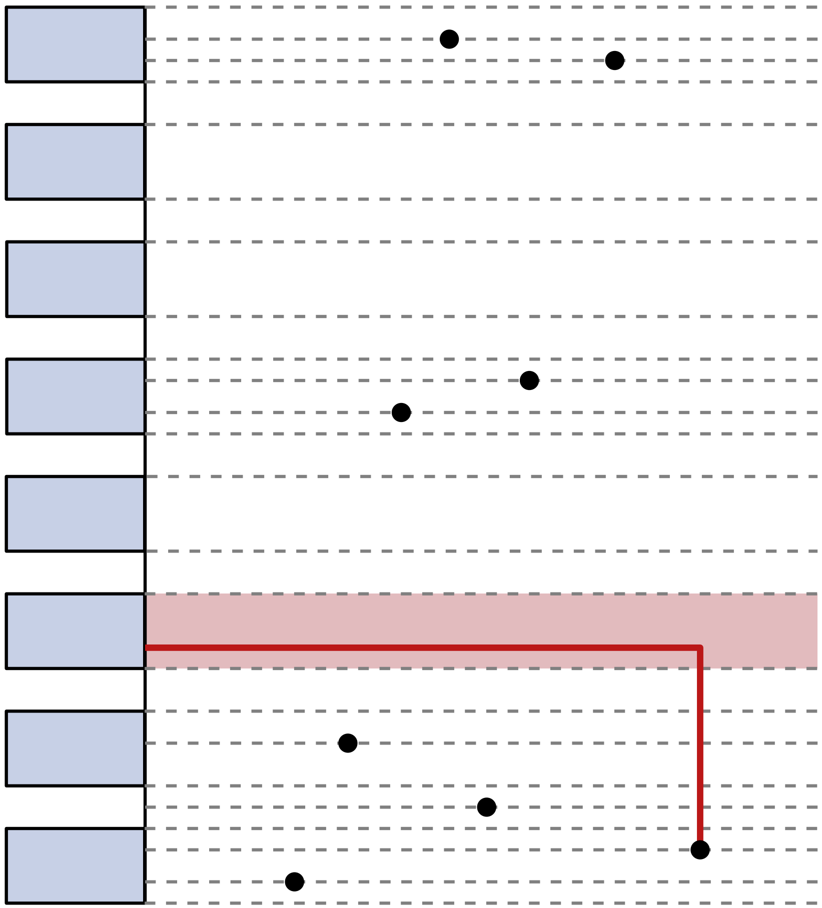


Ein dynamisches Programm (Benkert et al. '09)



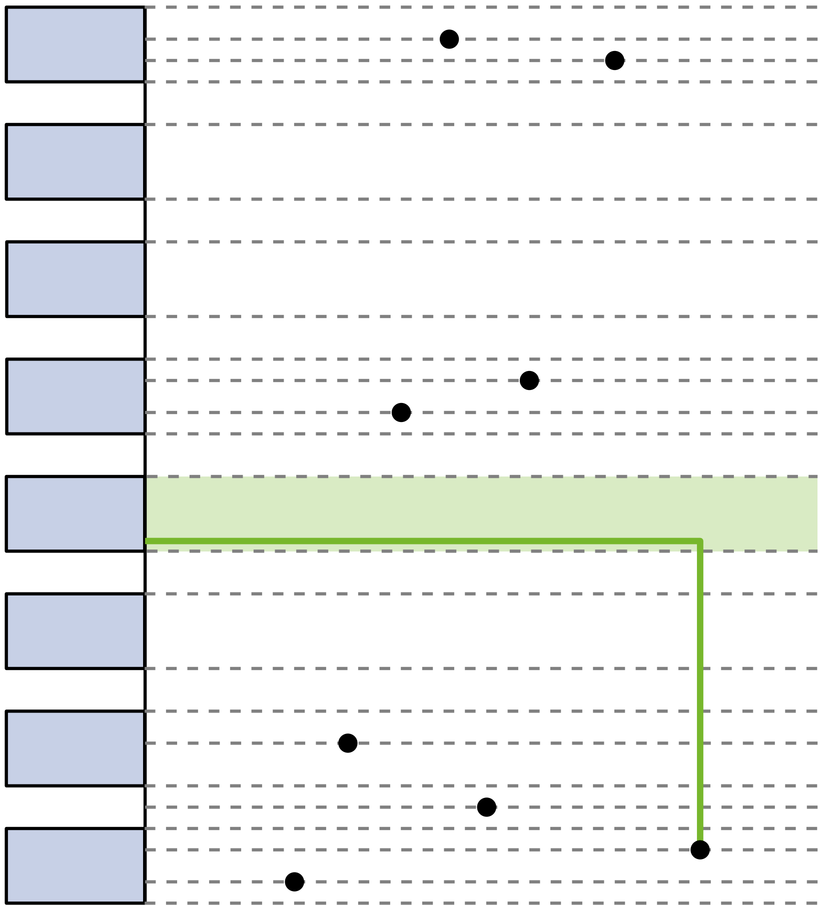
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen

Ein dynamisches Programm (Benkert et al. ’09)



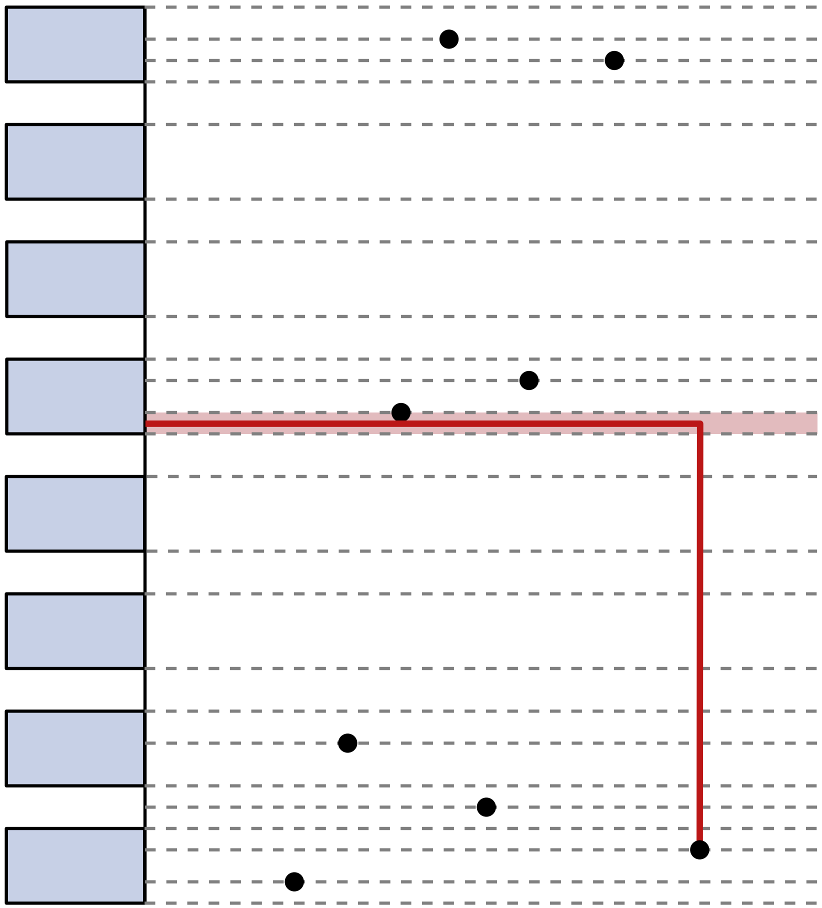
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. '09)



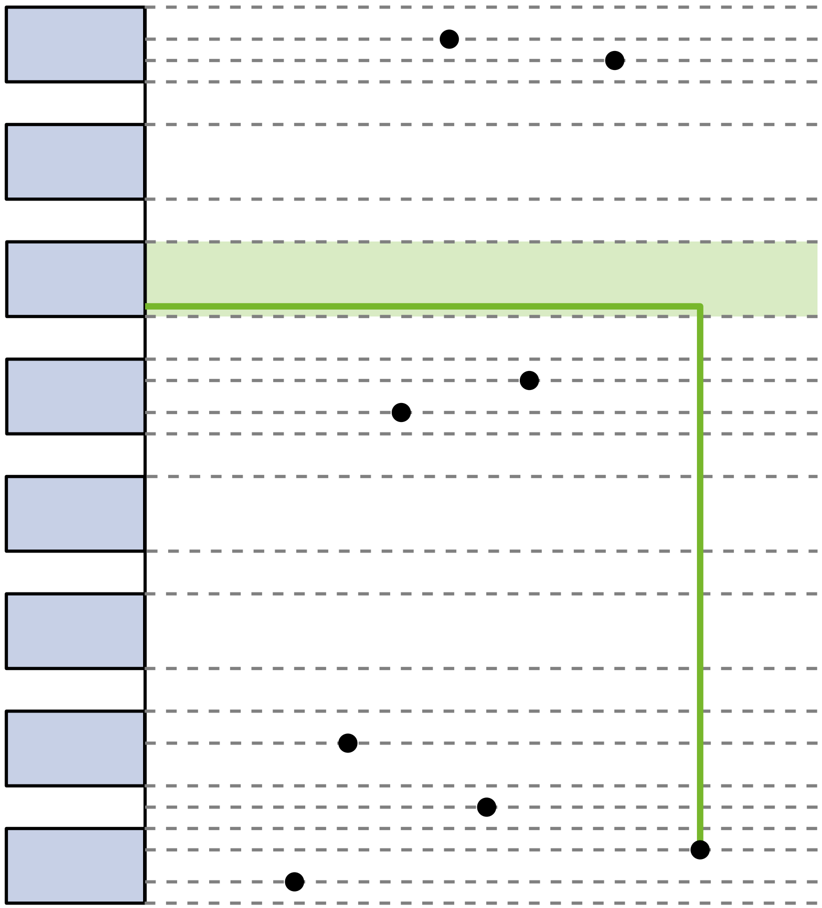
- Labelgrenzen und Punkte induzieren $3n+1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. '09)



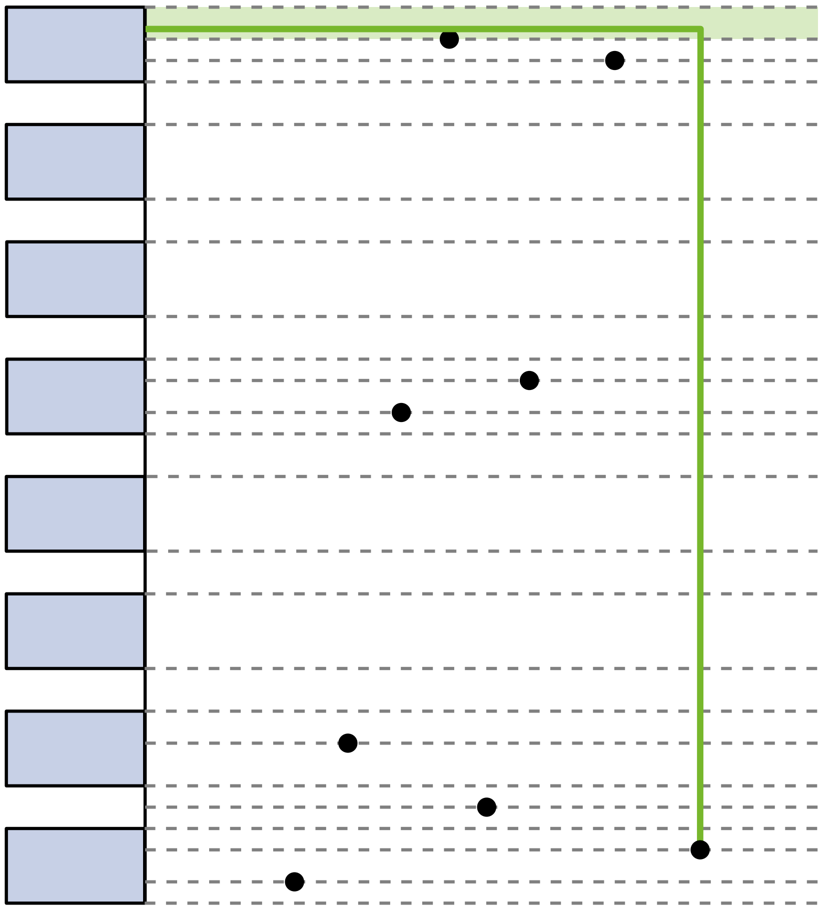
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. ’09)



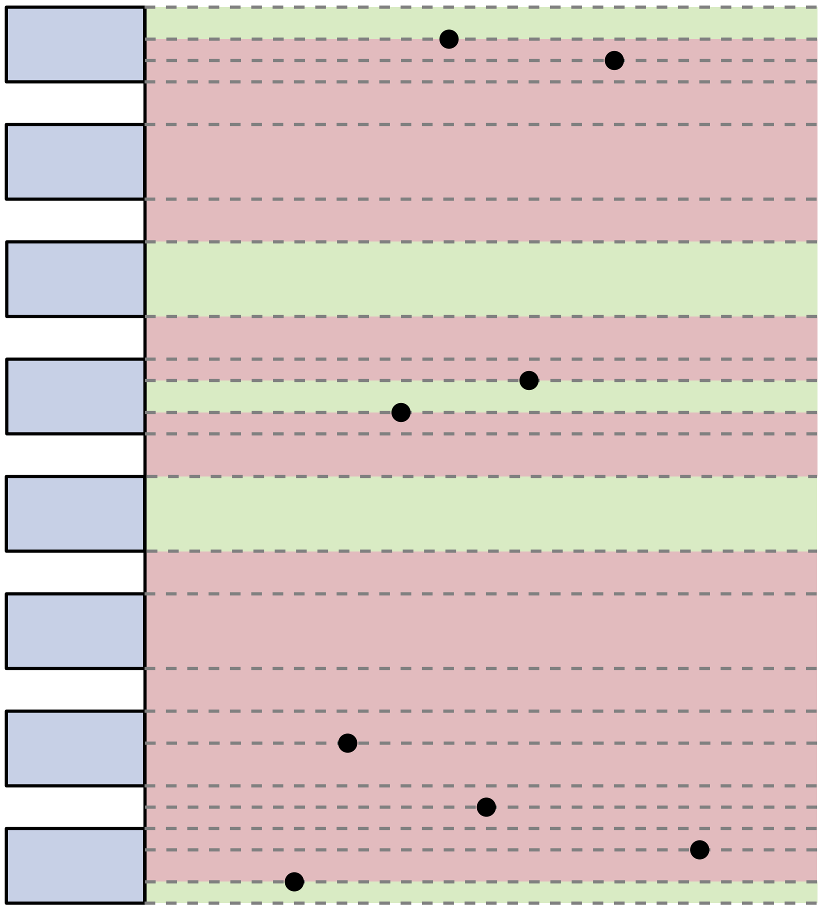
- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

Ein dynamisches Programm (Benkert et al. ’09)



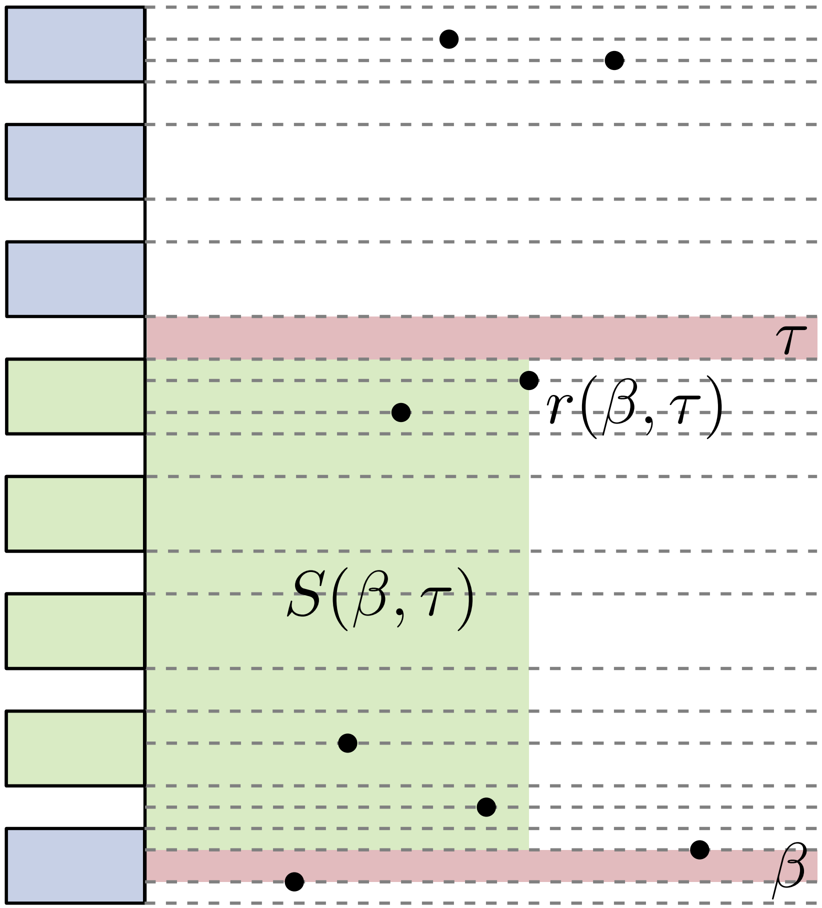
- Labelgrenzen und Punkte induzieren $3n+1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r

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)

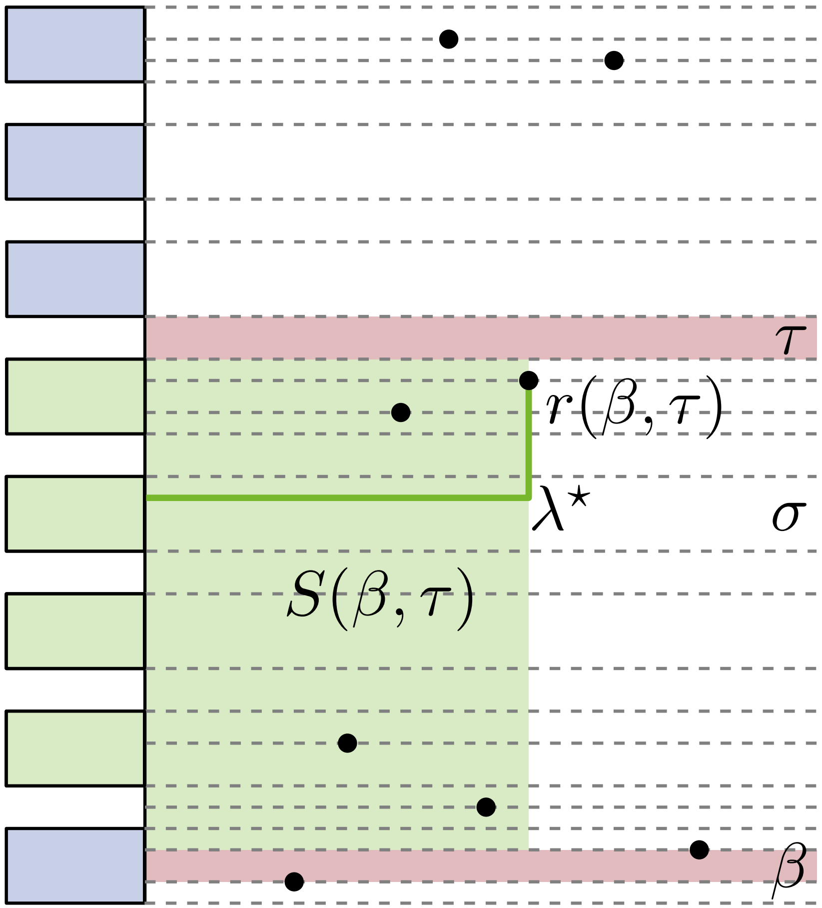


- Labelgrenzen und Punkte induzieren $3n + 1$ Streifen
- rechtester Punkt r und Leader-Streifen σ definieren zwei Teilprobleme
- optimiere über alle **zulässigen** Leader-Streifen für r, d.h. # Punkte und # Label in Teilinstanzen passen

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

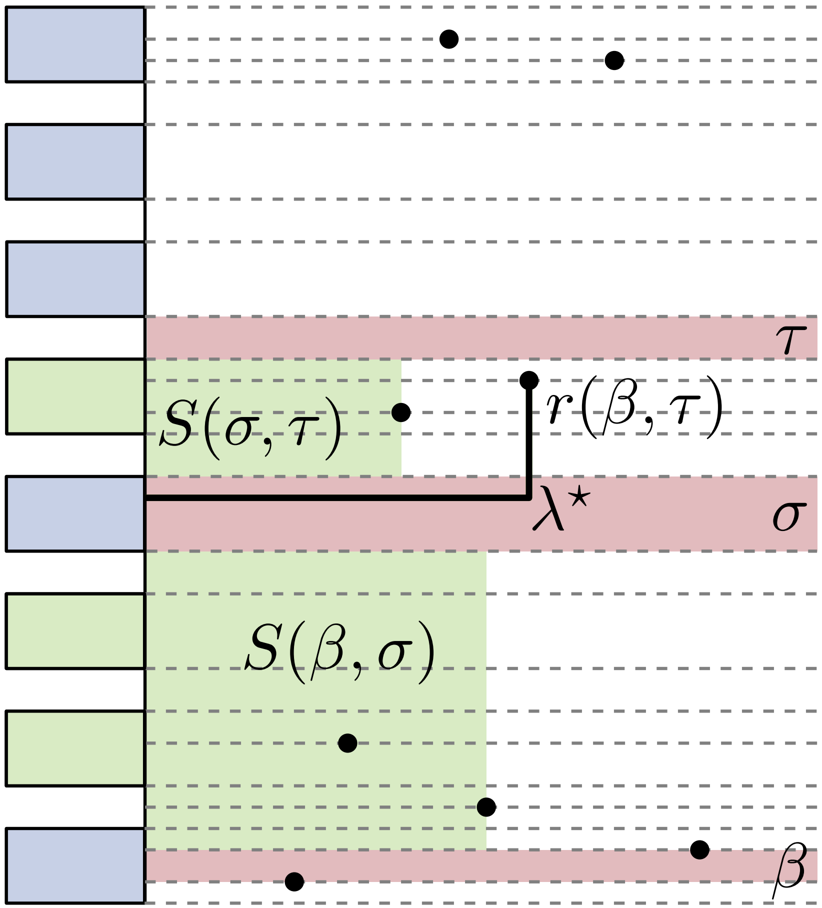
- alle Streifen zwischen β und τ
- alle k vollständigen Label
- k linkeste Punkte
- davon rechtester Punkt $r(\beta,\tau)$

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 ? Zeit und ? Platz eine zulässige po-Randbeschriftung mit minimalen Kosten berechnet werden.

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

Vorberechnungen:

- Sortiere Punkte nach x- und y-Koordinaten: $O(n \log n)$
- Bestimme $\lambda^\star(p,\sigma)$ und $\text{bad}(\lambda^\star(p,\sigma)$ $\forall$ Punkte und Streifen.
  Annahme: Jeweils $O(n)$ Zeitaufwand.

$\longrightarrow$ $O(n^3)$ Zeit und $O(n^2)$ Speicher

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:

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

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

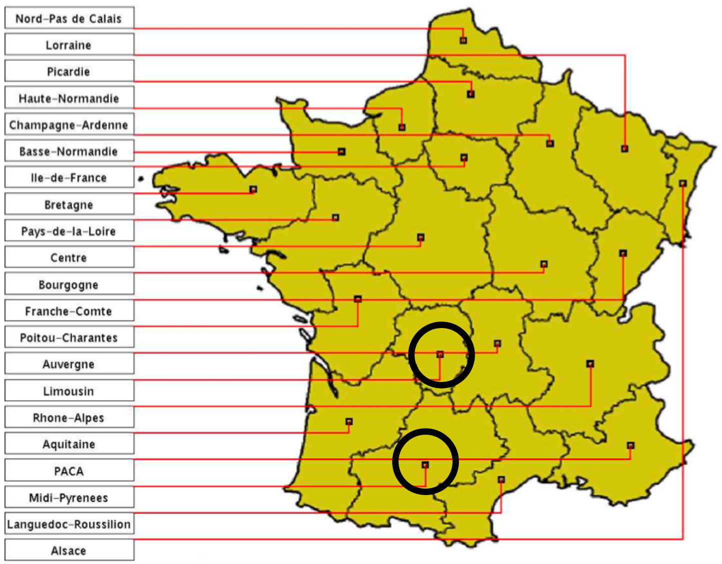
In einem Rekursionsschritt:

- Bestimme $r(\beta,\tau)$
- Bestimme zul. Streifen für $\sigma(\beta,\tau)$: Simultanes Durchlaufen von Labels und Punkten
- Erstelle x- und y-Sortierungen für Teilinstanzen

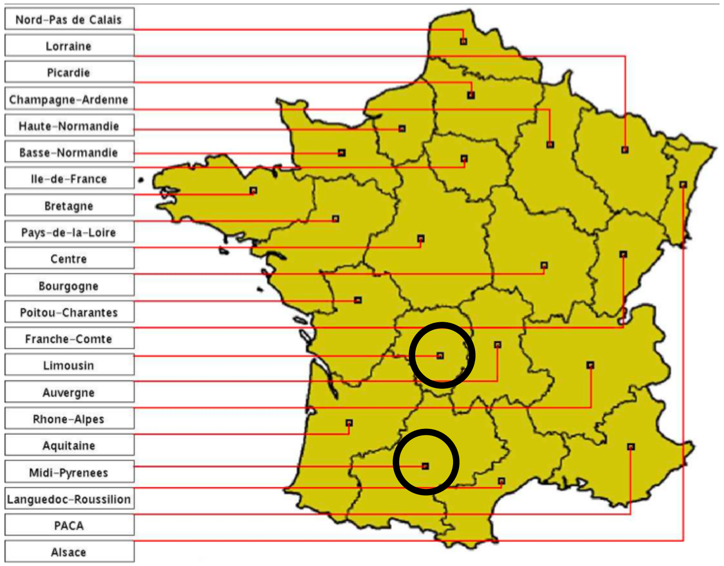
$\longrightarrow$ $O(n)$ Zeitaufwand pro Tabelleneintrag.

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.



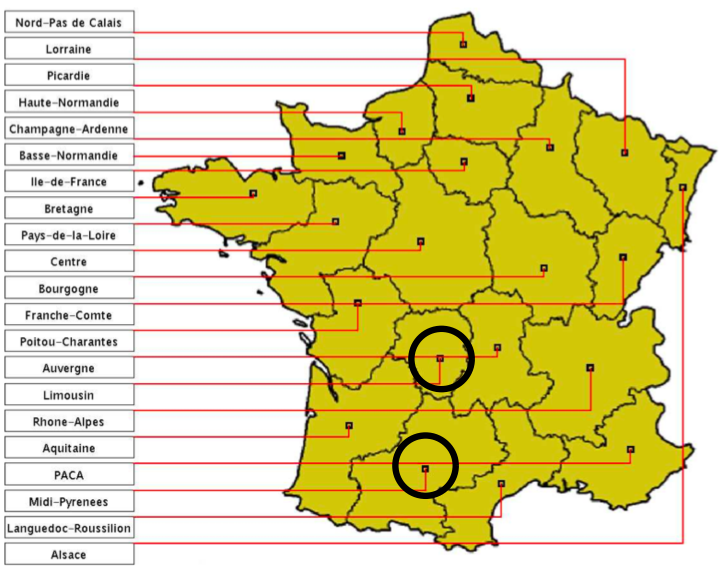
Längenminimierung



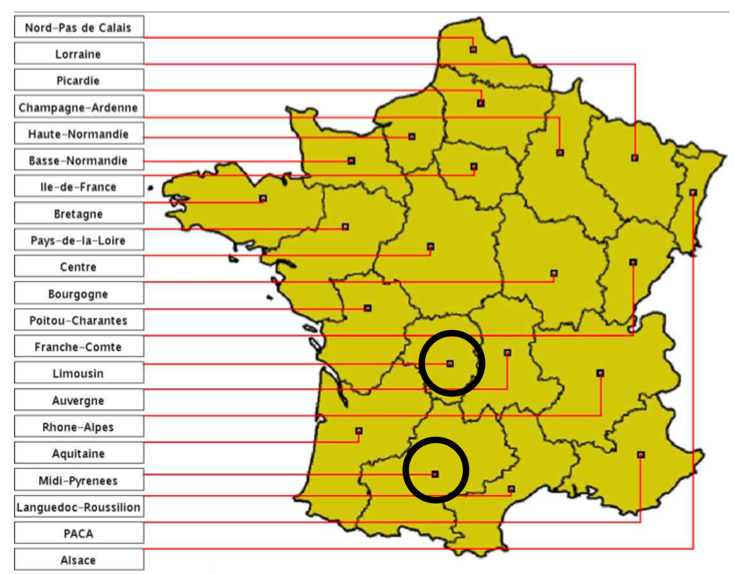
Länge & Leader-Punkt-Abstände

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.



Längenminimierung



Länge & Leader-Punkt-Abstände