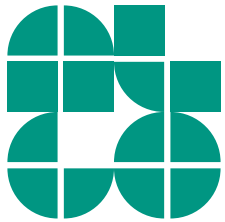
Vorlesung Algorithmische Kartografie

Boundary Labeling

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

Benjamin Niedermann

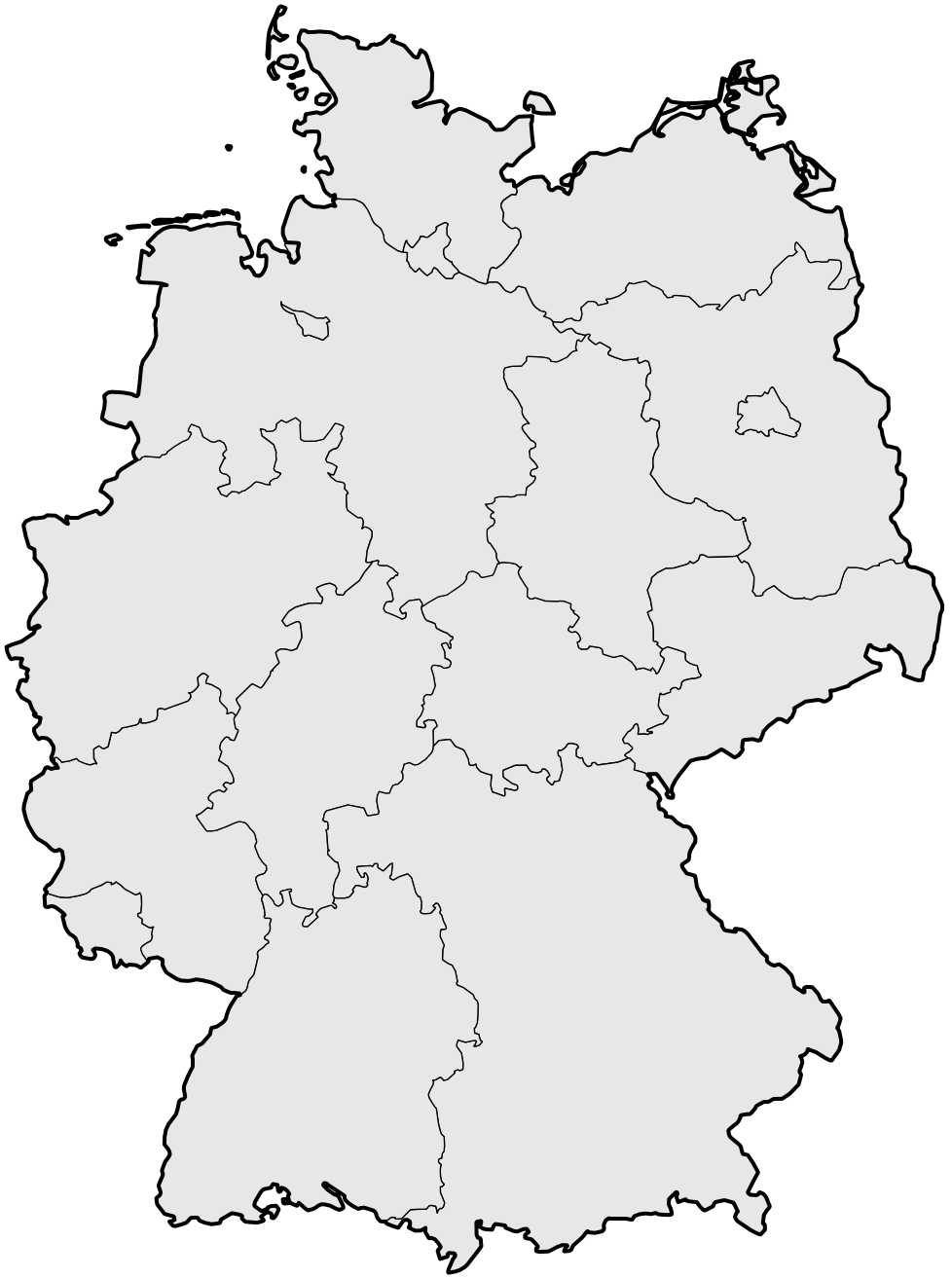
21.05.2013



Organisatorisches

Übung diesen Donnerstag von 9:45 Uhr bis 11:15 Uhr.

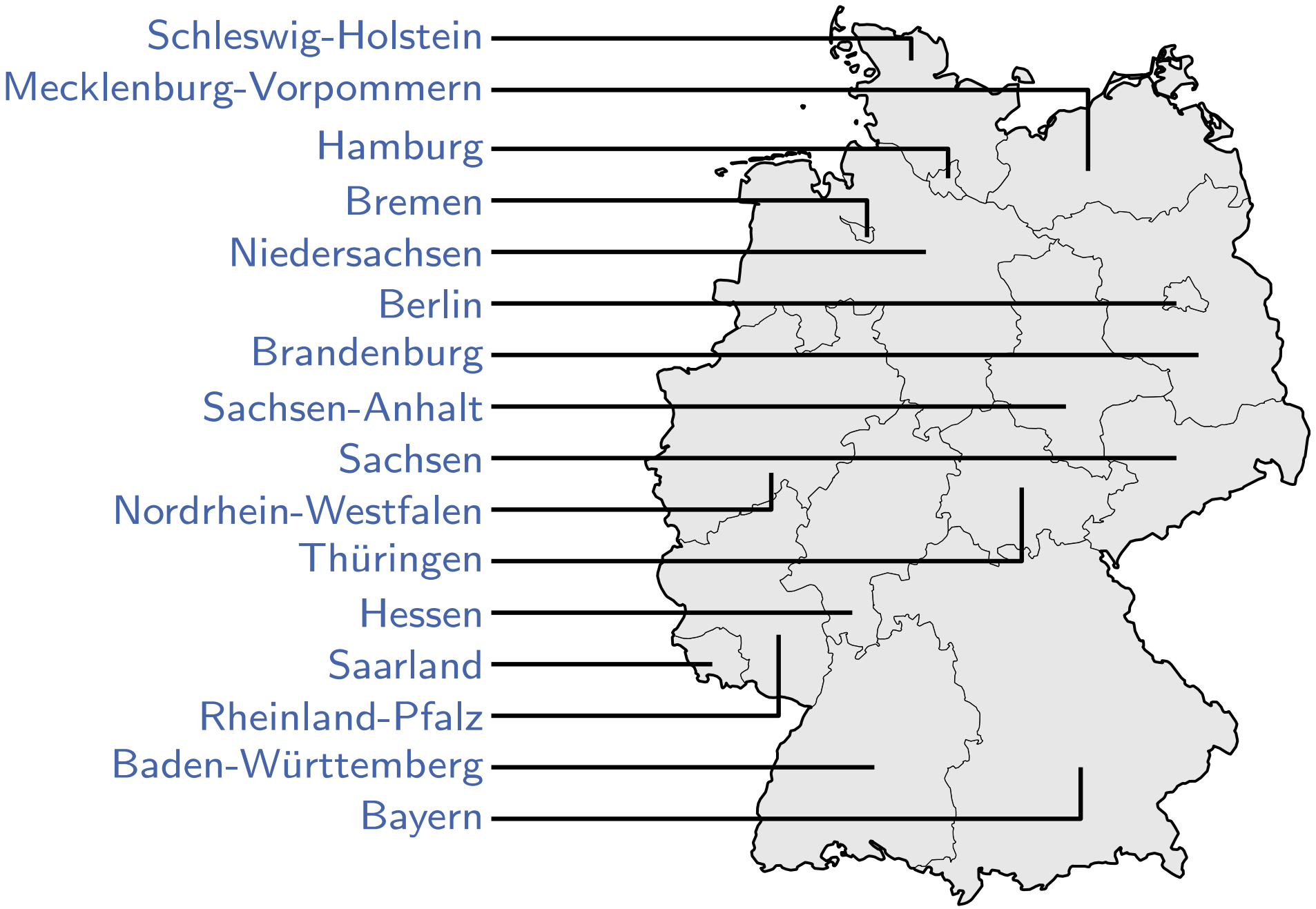
Motivation



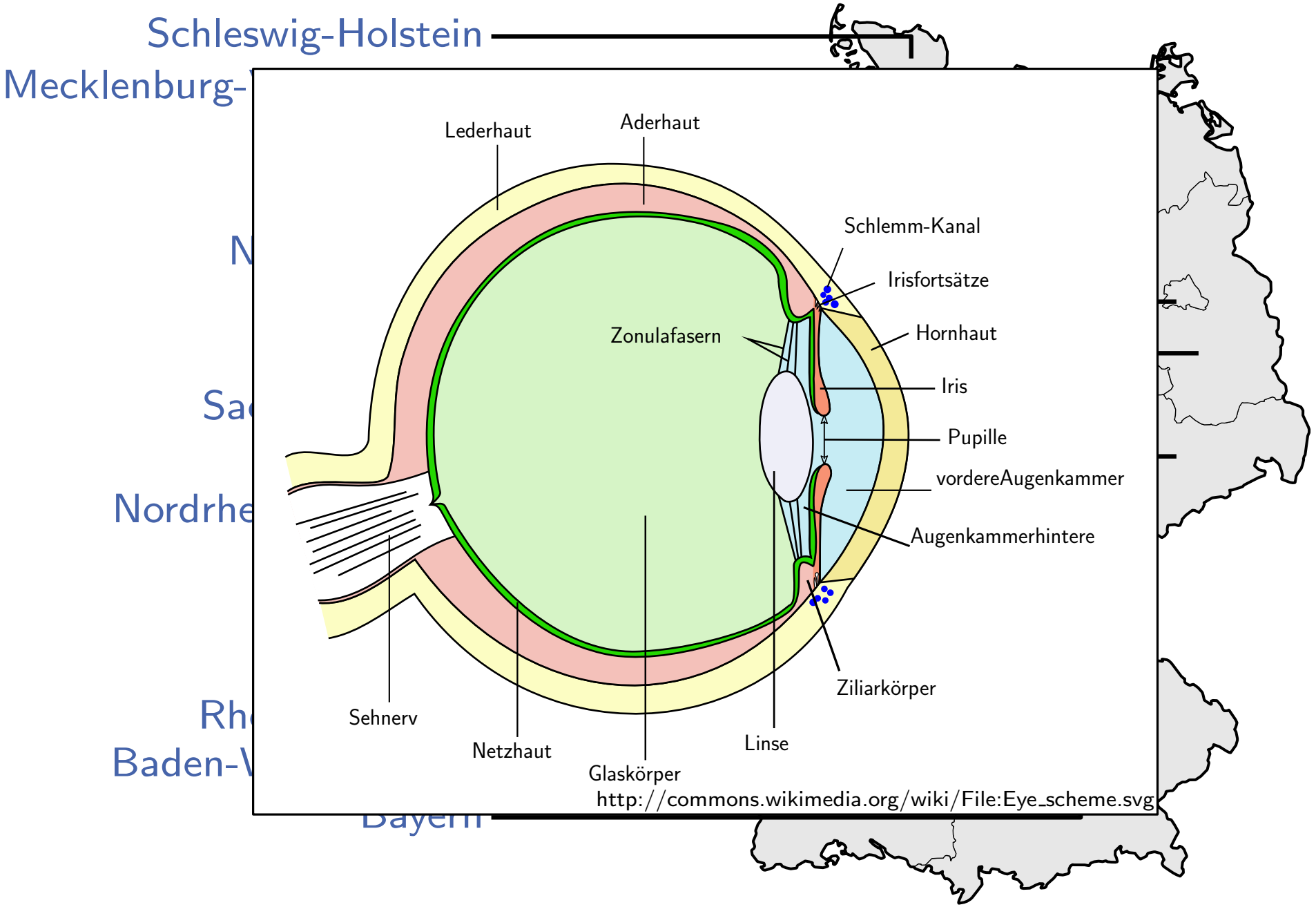
Motivation


Schleswig-Holstein
Mecklenburg-Vorpommern
Hamburg
Bremen
Niedersachsen
Berlin
Brandenburg
Sachsen-Anhalt
Nordrhein-Westfalen
Sachsen
Thüringen
Hessen
Rheinland-Pfalz
Saarland
Bayern
Baden-Württemberg

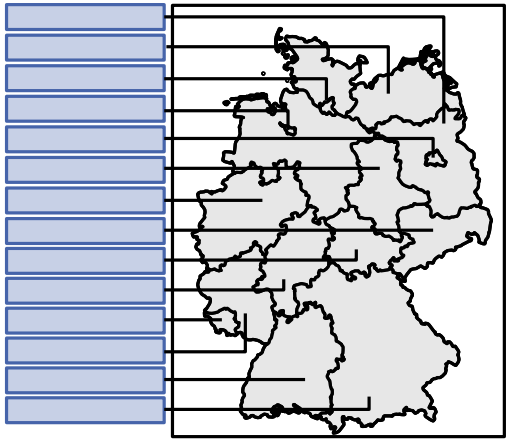
Motivation


Schleswig-Holstein
Mecklenburg-Vorpommern
Hamburg
Bremen
Niedersachsen
Berlin
Brandenburg
Sachsen-Anhalt
Sachsen
Nordrhein-Westfalen
Thüringen
Hessen
Saarland
Rheinland-Pfalz
Baden-Württemberg
Bayern

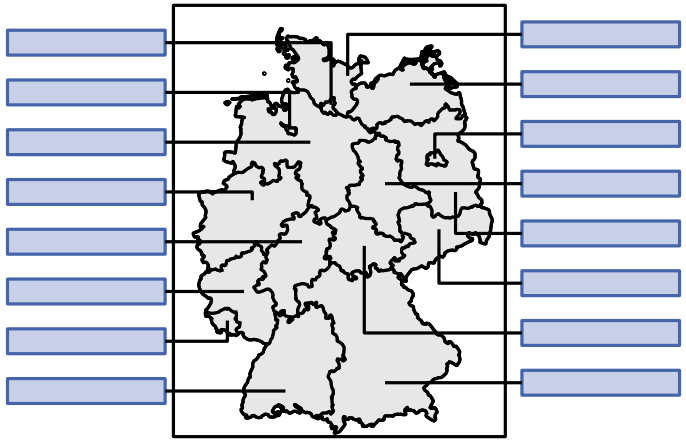
Motivation


Schleswig-Holstein
Mecklenburg-
Nordrhe
Baden-
Bayern
Lederhaut
Aderhaut
Schlemm-Kanal
Irisfortsätze
Zonulafasern
Hornhaut
Iris
Pupille
vordereAugenkammer
Augenkammerhintere
Ziliarkörper
Sehnerv
Netzhaut
Glaskörper
Linse
http://commons.wikimedia.org/wiki/File:Eye_scheme.svg

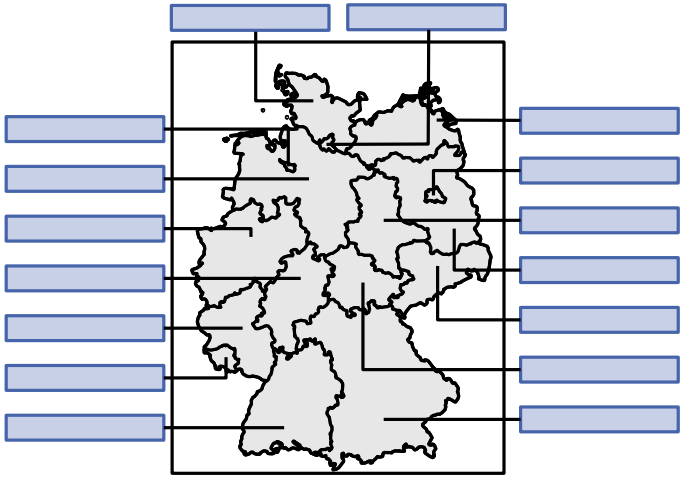
Arten von Boundary-Labeling



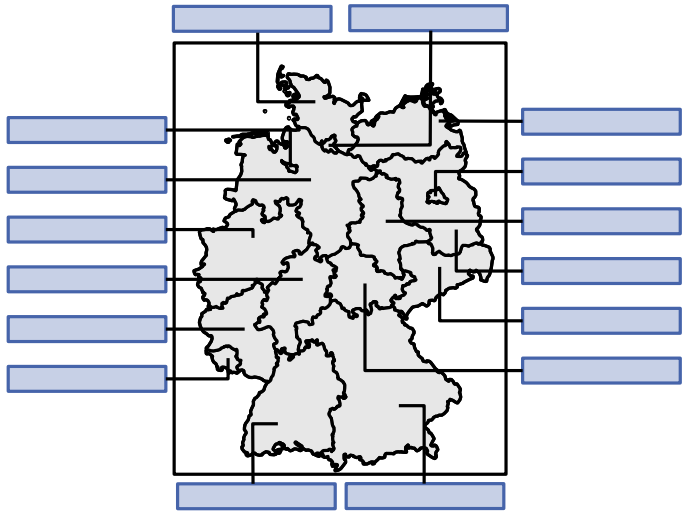
Einseitiger Fall



Zweiseitiger Fall

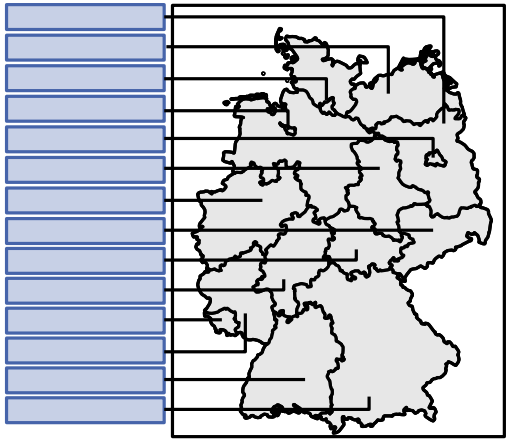


Dreiseitiger Fall

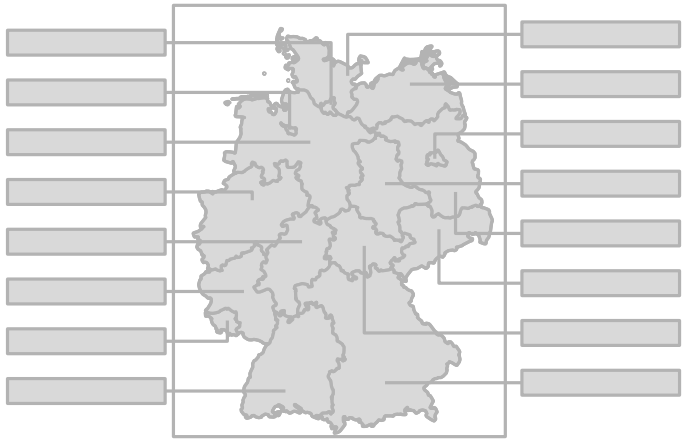


Vierseitiger Fall

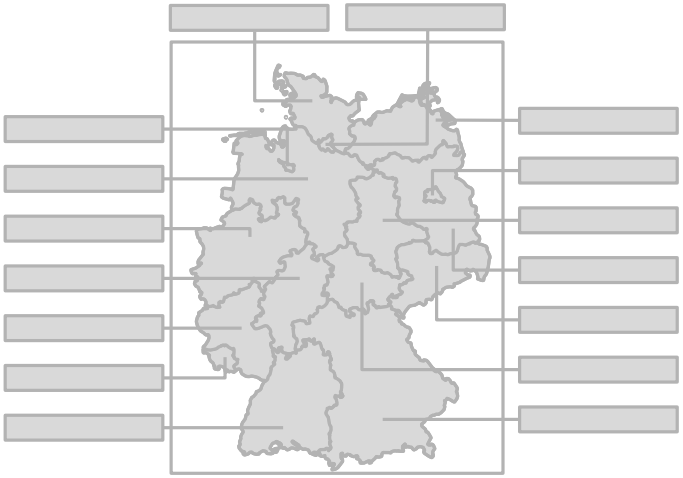
Arten von Boundary-Labeling



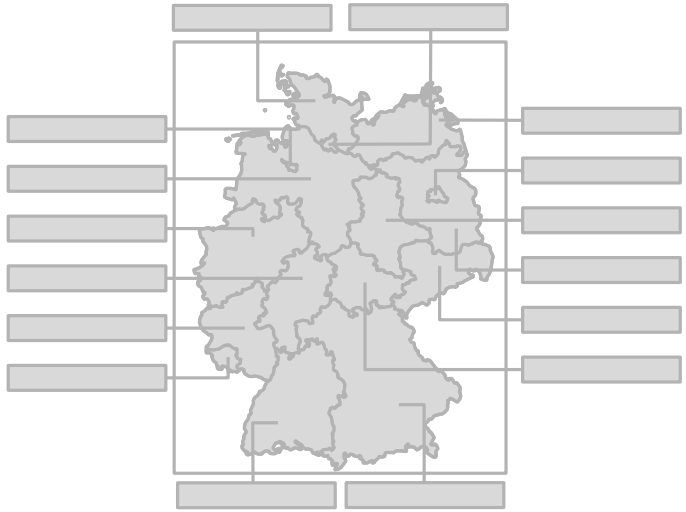
Einseitiger Fall



Zweiseitiger Fall



Dreiseitiger Fall

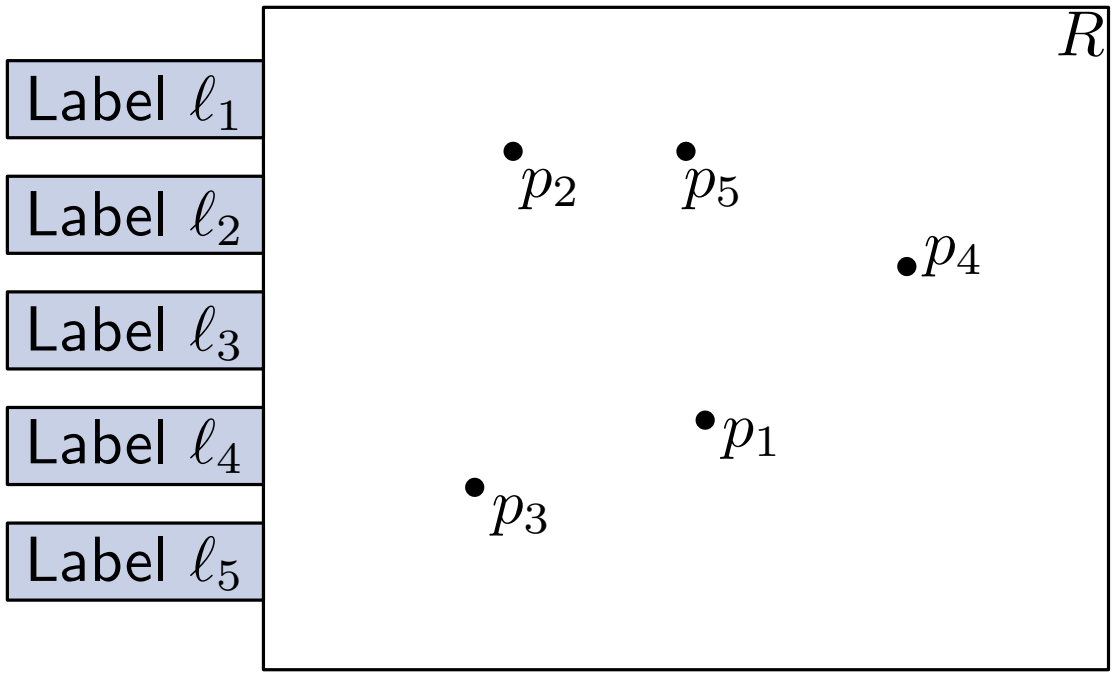


Vierseitiger Fall

Problemstellung

Gegeben:

- Punkte $P = \{p_1, \ldots, p_n\}$ in einem Rechteck R.
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \ldots, \ell_n\}$ mit rechter Seite auf linker Seite von R. Rechtecke aus L heißen *Label*.



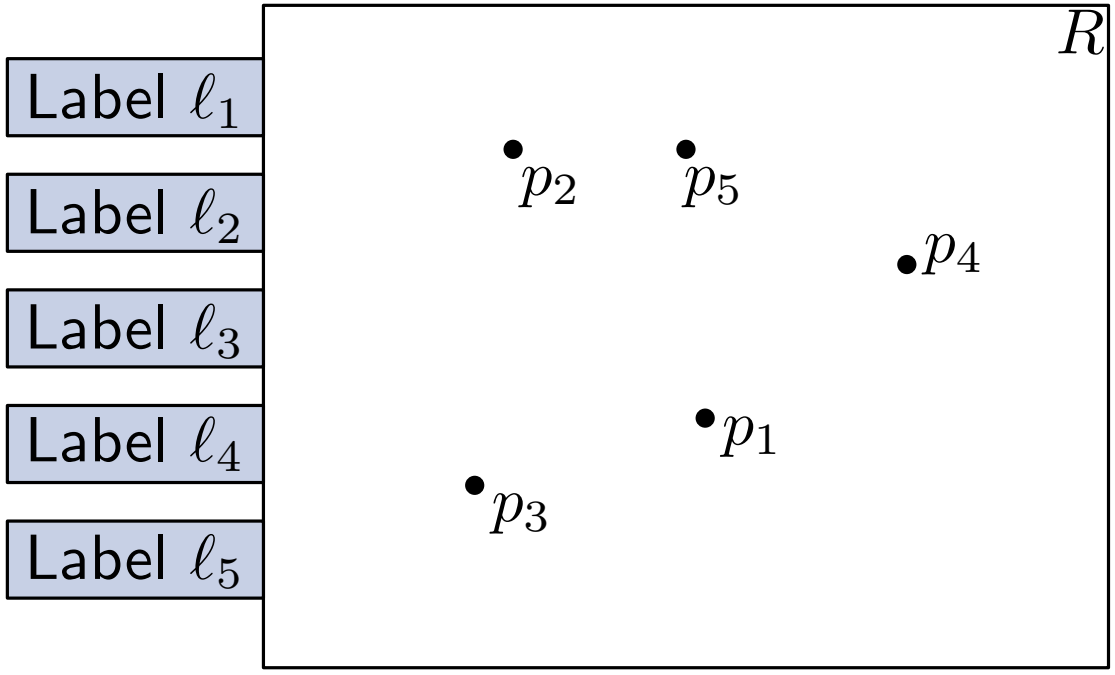
Problemstellung

Gegeben:

- Punkte $P = \{p_1, \ldots, p_n\}$ in einem Rechteck R.
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \ldots, \ell_n\}$ mit rechter Seite auf linker Seite von R. Rechtecke aus L heißen *Label*.

Annahme:

- Jedes Label ist groß genug, sodass die Beschreibung eines jeden Punkts hinein passt.
- Es sind genauso viele Label wie Punkte vorhanden.



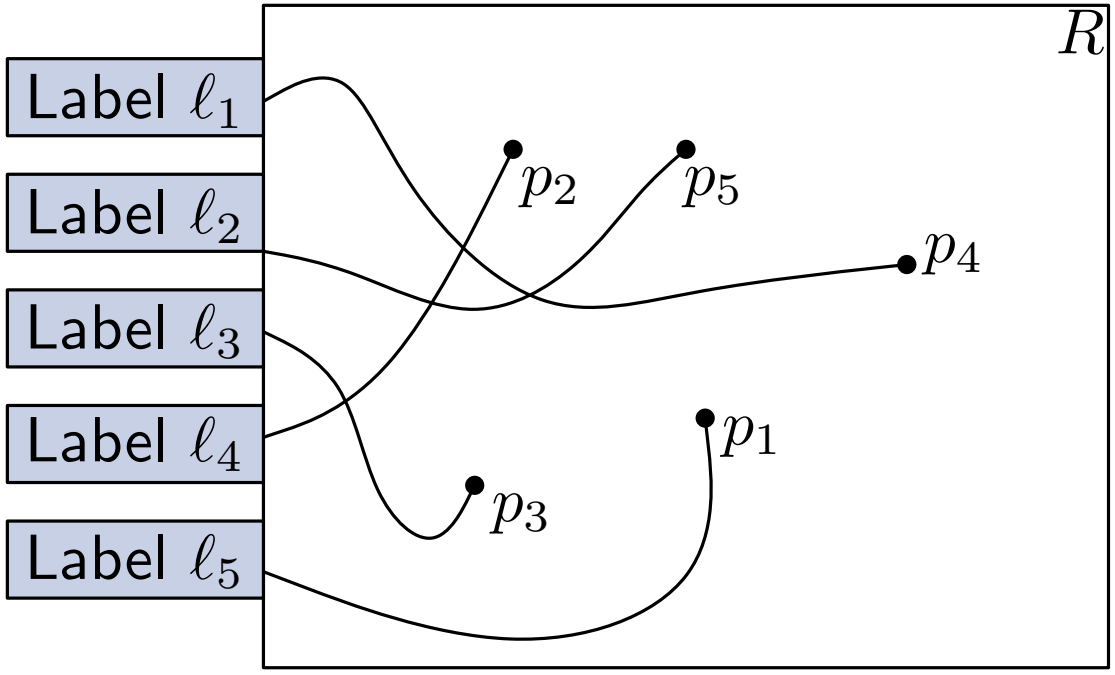
Problemstellung

Gegeben:

- Punkte $P = \{p_1, \ldots, p_n\}$ in einem Rechteck R.
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \ldots, \ell_n\}$ mit rechter Seite auf linker Seite von R. Rechtecke aus L heißen *Label*.

Definition: Eine Menge $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$ an Kurven in R heißt *Zuordnung* von L und P, wenn

- jede Kurve einen Punkt $p_i \in P$ mit einem Label $\ell_j \in L$ verbindet und
- jeder Punkt und jedes Label genau von einer Kurve angebunden wird.

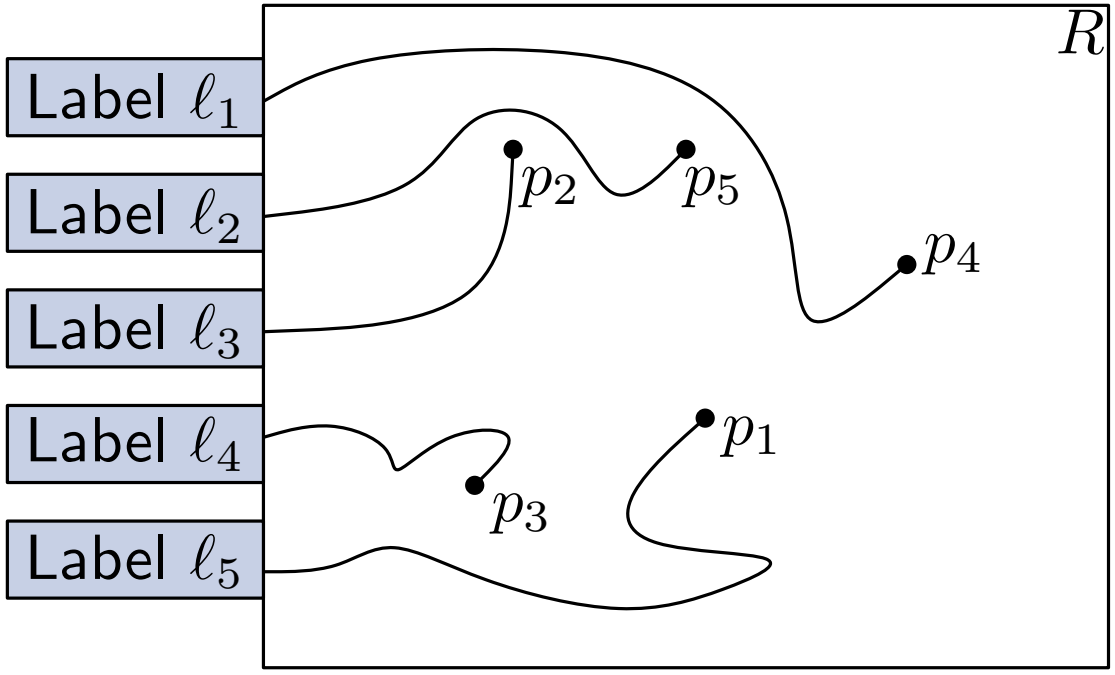


Problemstellung

Gegeben:

- Punkte $P = \{p_1, \ldots, p_n\}$ in einem Rechteck R.
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \ldots, \ell_n\}$ mit rechter Seite auf linker Seite von R. Rechtecke aus L heißen *Label*.

Gesucht: *Kreuzungsfreie* Zuordnung $\mathcal{L}$ von L und P, d.h. die Kurven in $\mathcal{L}$ schneiden sich nicht gegenseitig.

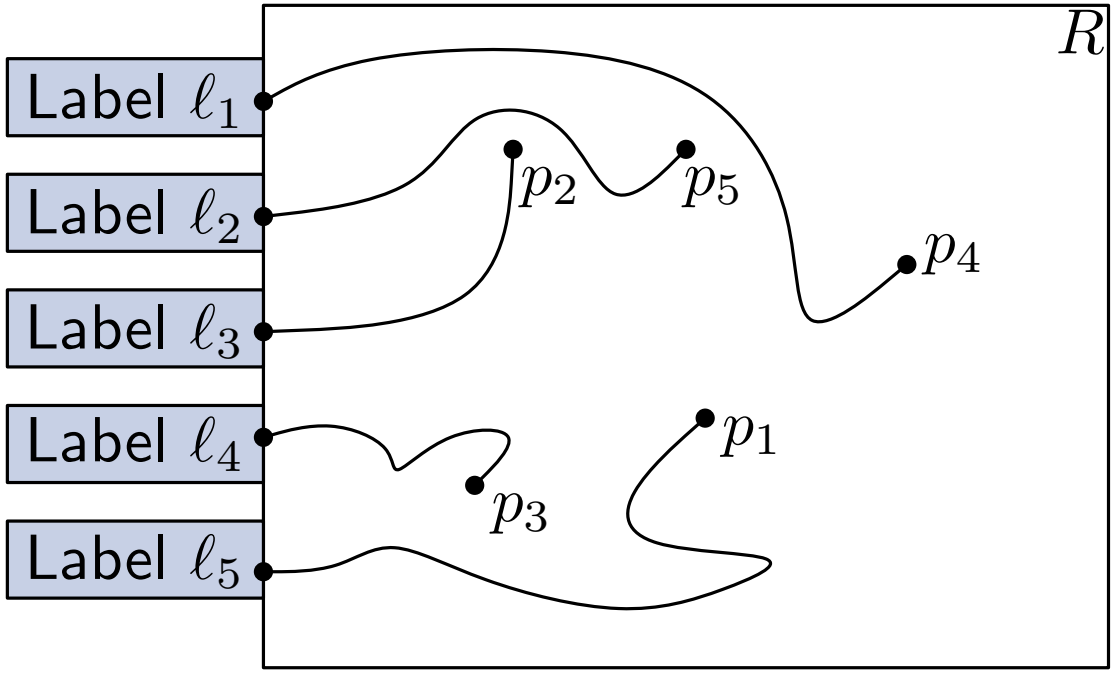


Problemstellung

Gegeben:

- Punkte $P = \{p_1, \ldots, p_n\}$ in einem Rechteck R.
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \ldots, \ell_n\}$ mit rechter Seite auf linker Seite von R. Rechtecke aus L heißen *Label*.

Gesucht: *Kreuzungsfreie* Zuordnung $\mathcal{L}$ von L und P, d.h. die Kurven in $\mathcal{L}$ schneiden sich nicht gegenseitig.



Begriffe:

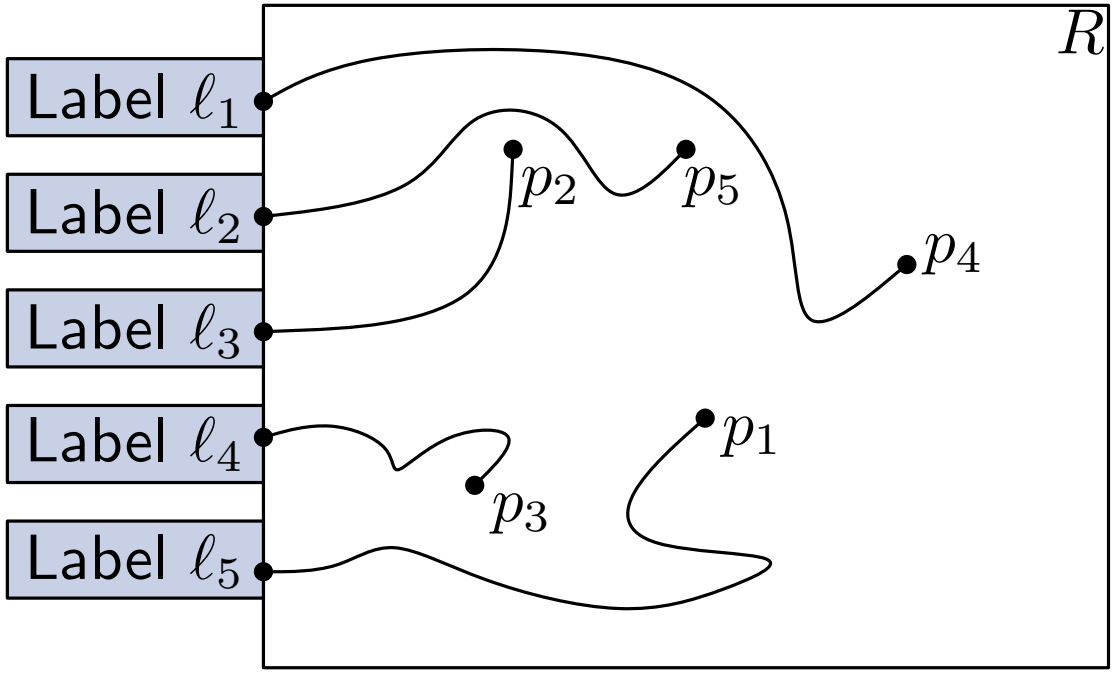
- Kurve λ_i heißt *Leader*.
- Verbindungspunkt von Leader und Label heißt *Port*.

Problemstellung

Gegeben:

- Punkte $P = \{p_1, \ldots, p_n\}$ in einem Rechteck R.
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \ldots, \ell_n\}$ mit rechter Seite auf linker Seite von R. Rechtecke aus L heißen *Label*.

Gesucht: *Kreuzungsfreie* Zuordnung $\mathcal{L}$ von L und P, d.h. die Kurven in $\mathcal{L}$ schneiden sich nicht gegenseitig.



Begriffe:

- Kurve λ_i heißt *Leader*.
- Verbindungspunkt von Leader und Label heißt *Port*.

Zwei Varianten:

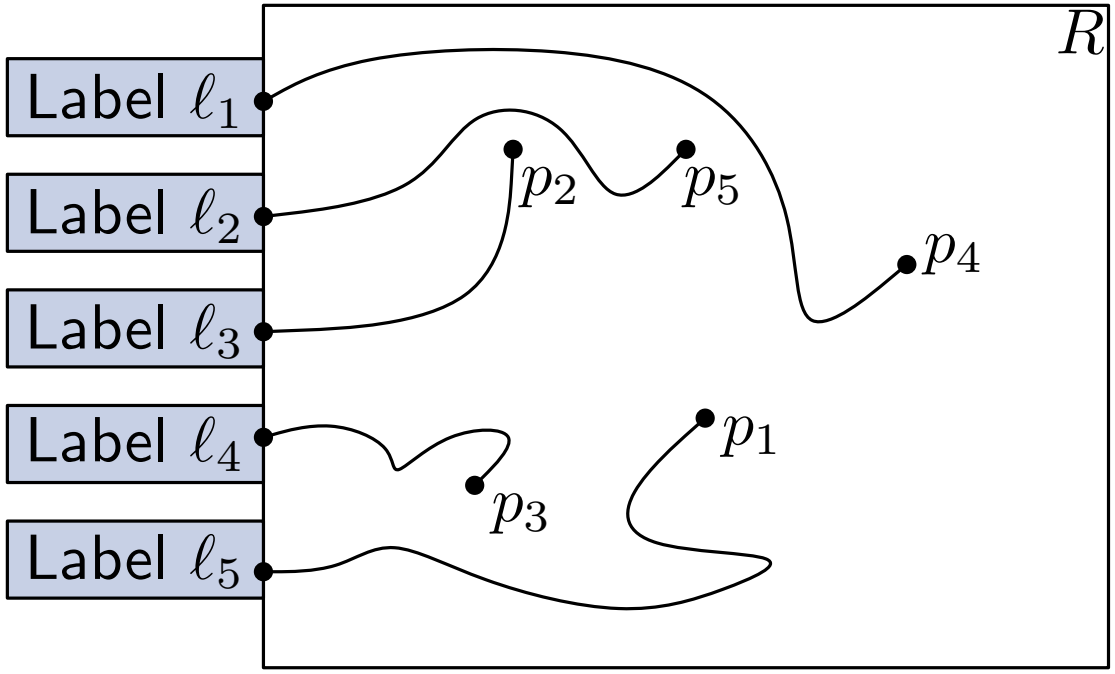
1. Ports fest vorgegeben.
2. Port auf rechter Seite von Label frei wählbar.

Problemstellung

Gegeben:

- Punkte $P = \{p_1, \ldots, p_n\}$ in einem Rechteck R.
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \ldots, \ell_n\}$ mit rechter Seite auf linker Seite von R. Rechtecke aus L heißen *Label*.

Gesucht: *Kreuzungsfreie* Zuordnung $\mathcal{L}$ von L und P, d.h. die Kurven in $\mathcal{L}$ schneiden sich nicht gegenseitig.



Begriffe:

- Kurve λ_i heißt *Leader*.
- Verbindungspunkt von Leader und Label heißt *Port*.

Zwei Varianten:

1. Ports fest vorgegeben.
2. Port auf rechter Seite von Label frei wählbar.

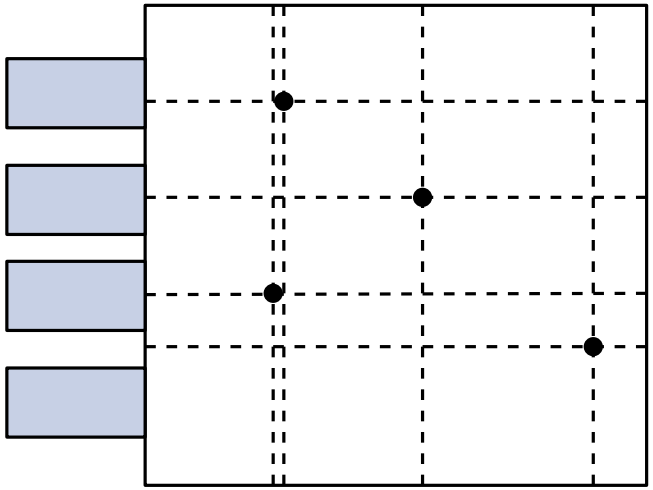
Problemstellung

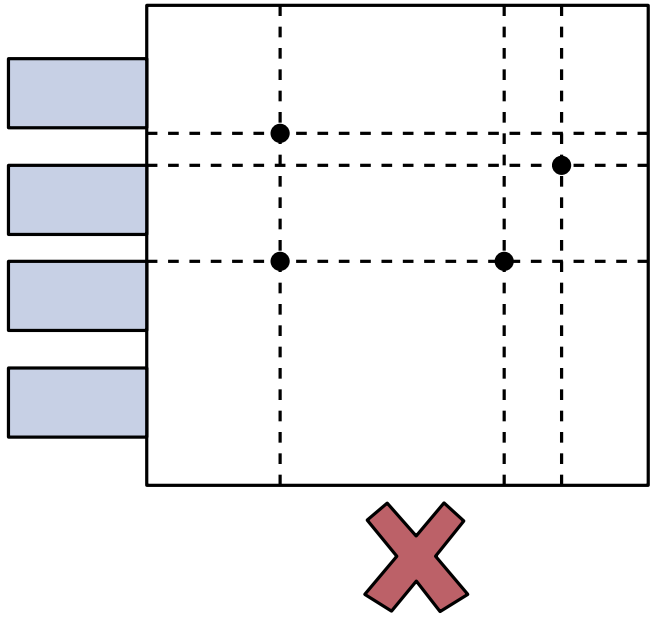
Gegeben:

- Punkte $P = \{p_1, \ldots, p_n\}$ in einem Rechteck R
- $L = \{l_1, \ldots, l_n\}$ mit

Annahme: Allgemeine Lage der Punkte in P und der Label in L.

- Keine zwei Punkte in P liegen auf einer gemeinsamen vertikalen oder horizontalen Geraden.
- Kein Punkt in P liegt auf einer horizontalen Geraden, welche die obere oder untere Seite eines Labels in L verlängert.

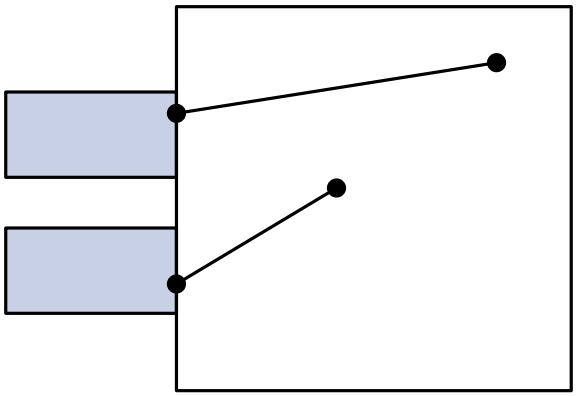




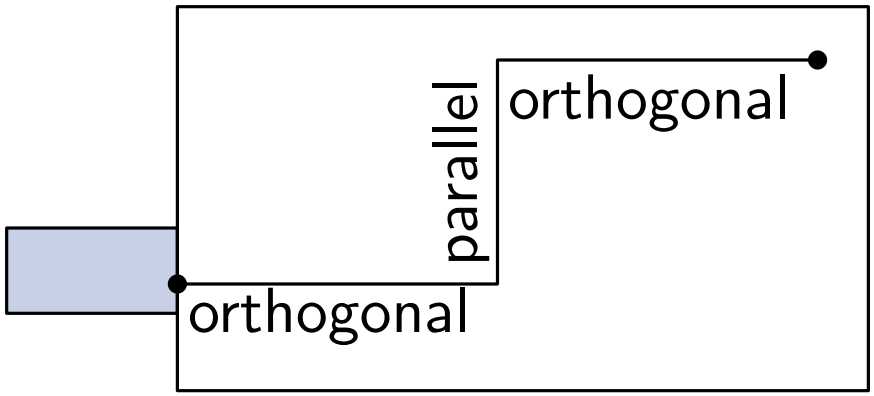
Typen von Leadern

Üblich: Leader werden aus Sequenz von Strecken zusammengesetzt.

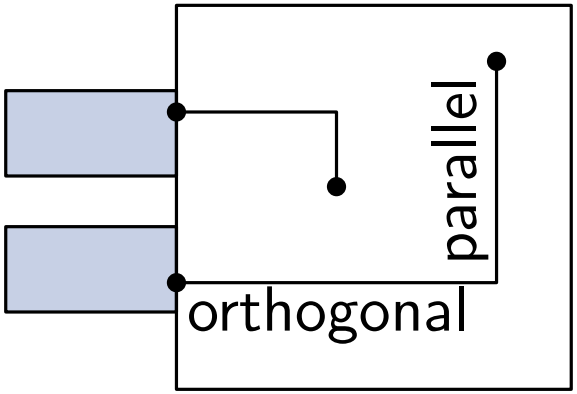
Typische Beispiele:



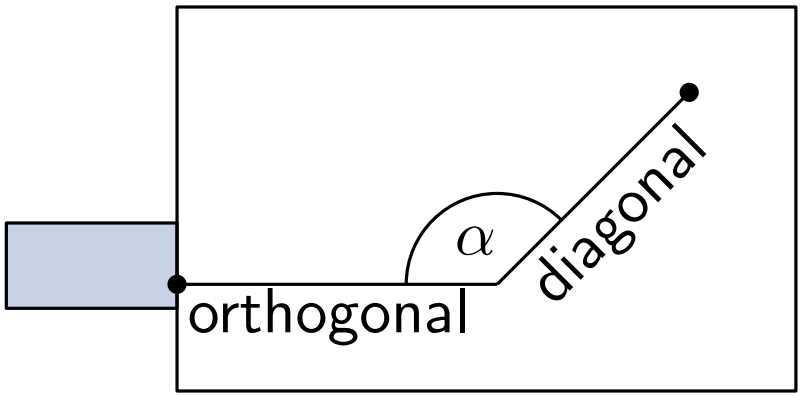
s-Leader



opo-Leader



po-Leader

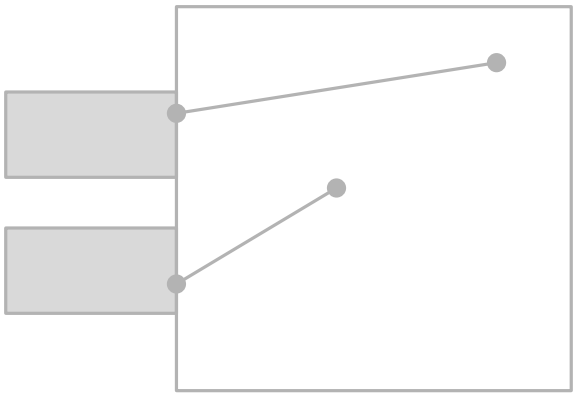


do-Leader

Typen von Leadern

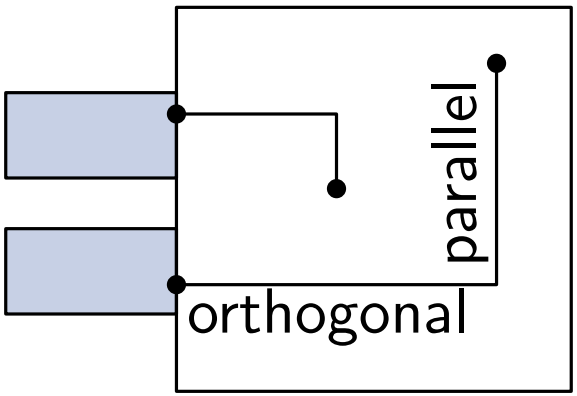
Üblich: Leader werden aus Sequenz von Strecken zusammengesetzt.

Typische Beispiele:



s-Leader

opo-Leader



po-Leader

do-Leader

Zielfunktion

Bewertungsfunktion:

$$\mathrm{bad} : L \times P \to \mathbb{R}$$

Bewertet Zuordnung von Labeln L und Punkten P.

Optimierungsproblem:
Finde für gegebene Instanz (R, P, L) eine kreuzungsfreie Zuordnung $\mathcal{L}$, sodass

$$\sum_{\lambda \in \mathcal{L}} \mathrm{bad}(\ell(\lambda), p(\lambda))$$

unter allen kreuzeungsfreien Zuordnungen minimiert wird.

Notation:

- $\ell(\lambda)$: Label aus L, das von Leader λ angebunden wird.
- $p(\lambda)$: Punkt aus p, der von Leader λ angebunden wird.

Zielfunktion

Bewertungsfunktion:

$$\text{bad} : L \times P \to \mathbb{R}$$

Bewertet Zuordnung von Labeln L und Punkten P.

Optimierungsproblem:
Finde für gegebene Instanz (R, P, L) eine kreuzungsfreie Zuordnung $\mathcal{L}$, sodass

$$\sum_{\lambda \in \mathcal{L}} \text{bad}(\ell(\lambda), p(\lambda))$$

unter allen kreuzeungsfreien Zuordnungen minimiert wird.

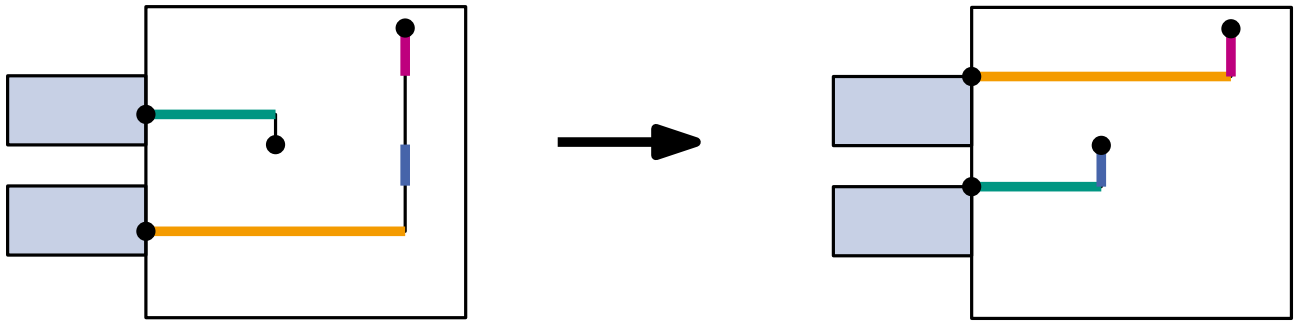
Notation:

- $\ell(\lambda)$: Label aus L, das von Leader λ angebunden wird.
- $p(\lambda)$: Punkt aus p, der von Leader λ angebunden wird.

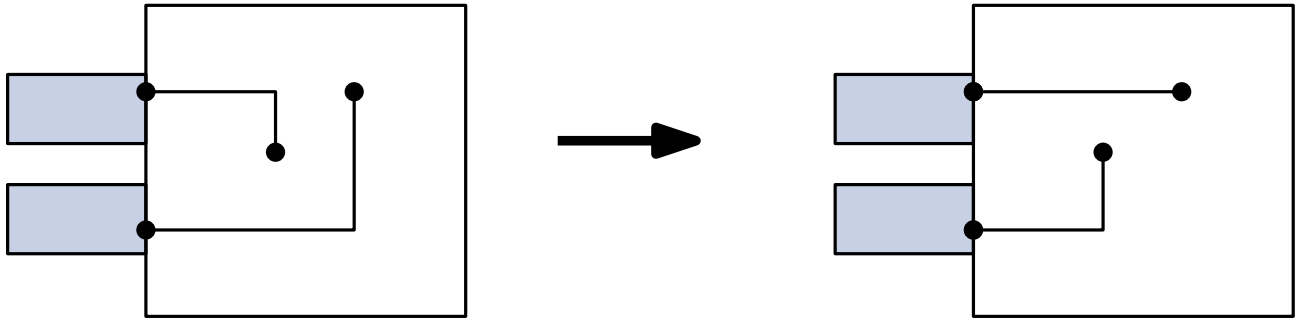
Beispiele für Bewertungsfunktionen?

Beispiele für Bewertungsfunktionen

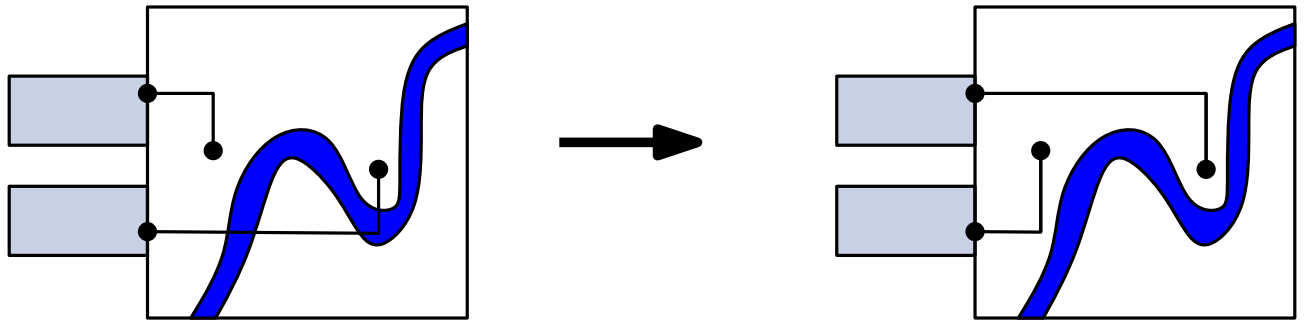
Längenminimierung:



Knickminimierung:



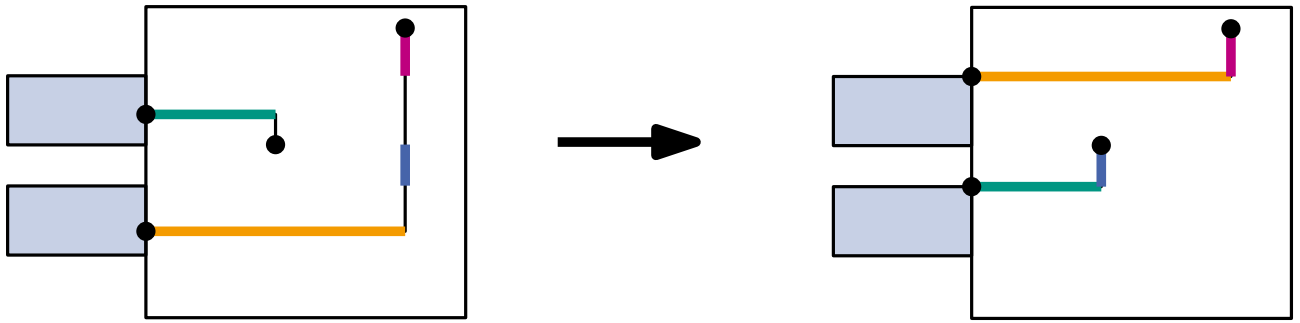
Interferenzminimierung mit Karte



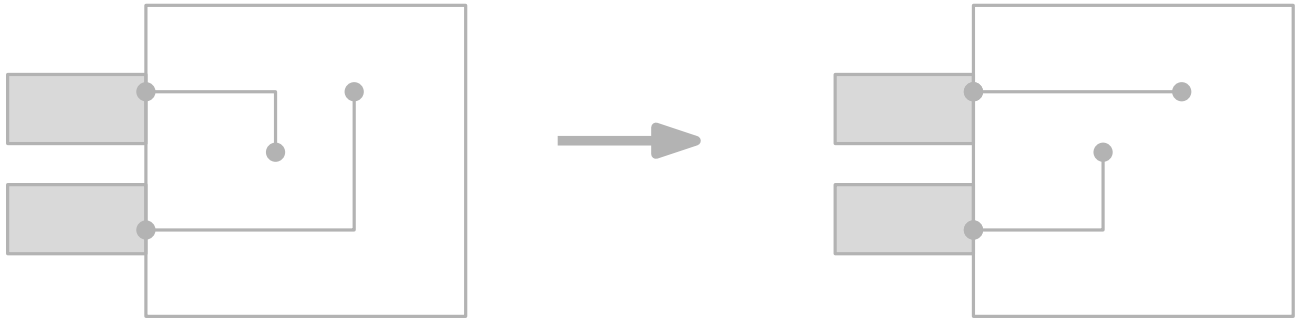
Beispiele für Bewertungsfunktionen

Karlsruhe Institute of Technology

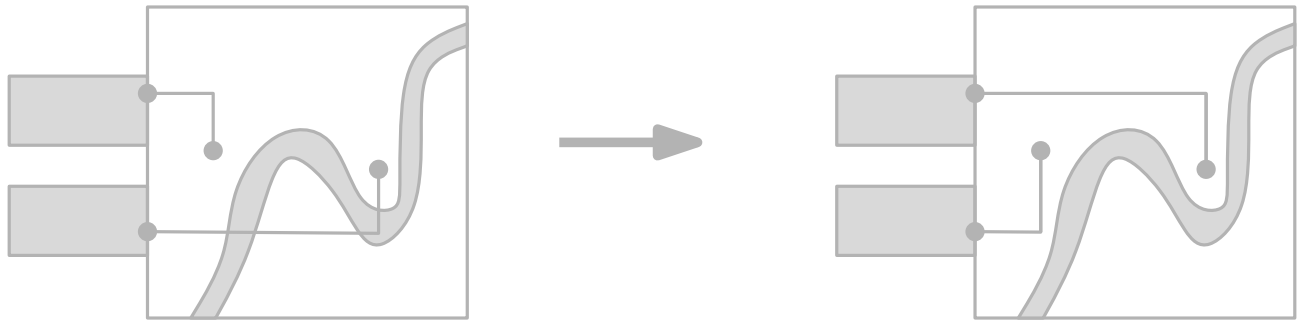
Längenminimierung:



Knickminimierung:

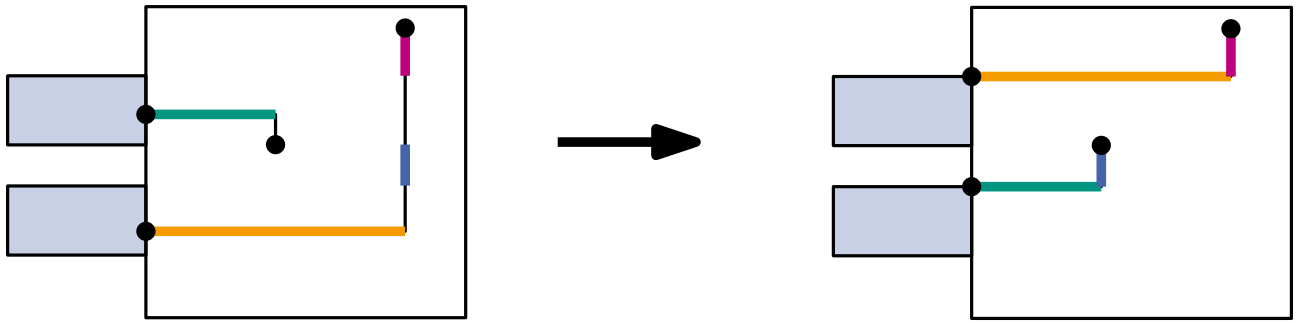


Interferenzminimierung mit Karte



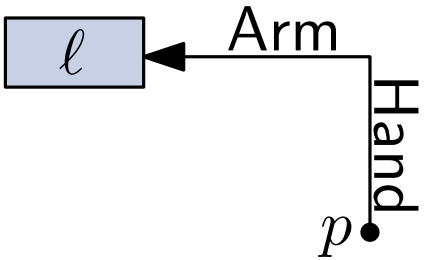
Beispiele für Bewertungsfunktionen

Längenminimierung:



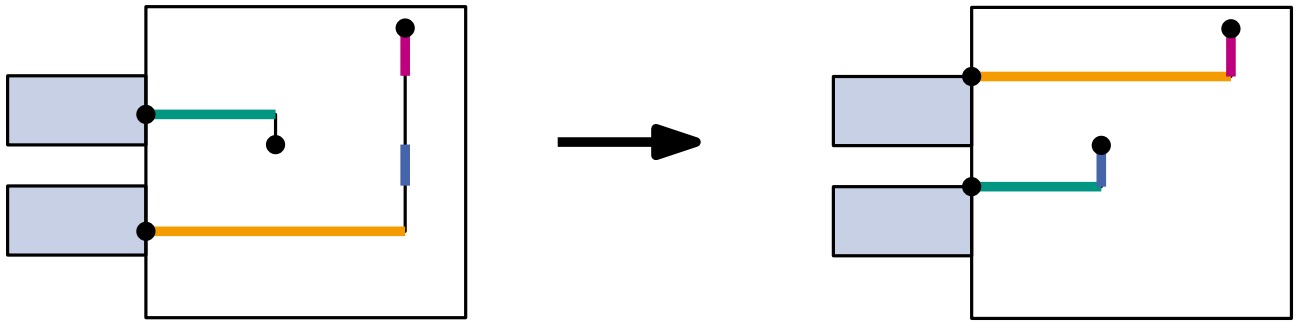
Begriffe:

- Länge einer Zuordnung $\mathcal{L}$: Die Summe der Leader-Längen.
- $\lambda(p, \ell)$ bezeichnet po-Leader der von p nach ℓ führt.



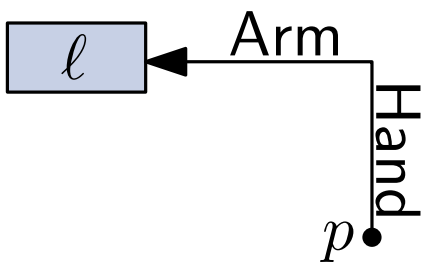
Beispiele für Bewertungsfunktionen

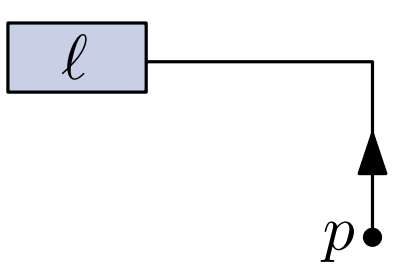
Längenminimierung:



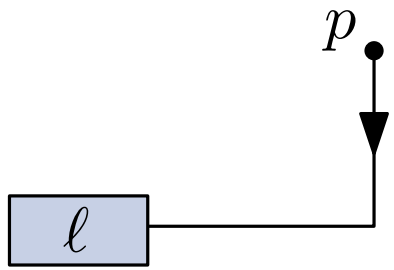
Begriffe:

- Länge einer Zuordnung $\mathcal{L}$: Die Summe der Leader-Längen.
- $\lambda(p, \ell)$ bezeichnet po-Leader der von p nach ℓ führt.





aufsteigender Leader



absteigender Leader

Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in $\mathcal{L}^*$ schneiden sich nicht.

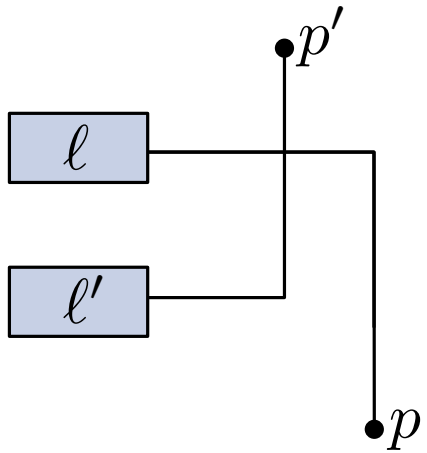
Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in $\mathcal{L}^*$ schneiden sich nicht.

Annahme es gäbe solche Leader:



Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in $\mathcal{L}^*$ schneiden sich nicht.

Annahme es gäbe solche Leader:

Kürzere Gesamtlänge der Leader nach Neuverdrahtung

↯ Minimalität von $\mathcal{L}^*$

Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in $\mathcal{L}^*$ schneiden sich nicht.

B) Kein direkter Leader in $\mathcal{L}^*$ kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Neuverdrahtung

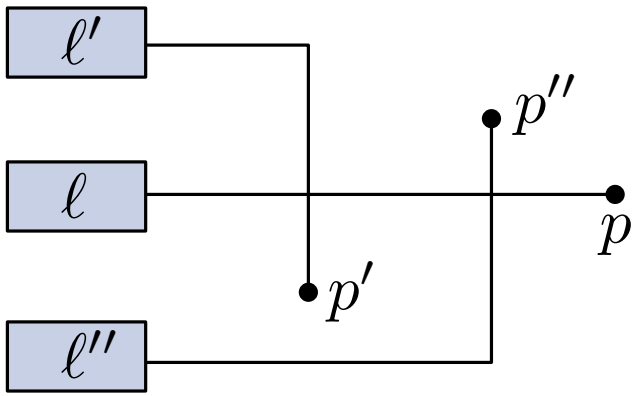
Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in $\mathcal{L}^*$ schneiden sich nicht.

B) Kein direkter Leader in $\mathcal{L}^*$ kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Annahme es gäbe einen solchen Leader:



Einziger Fall, denn wegen 1. können sich $\lambda(p, \ell)$ und $\lambda(p', \ell')$ nicht schneiden.

Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in $\mathcal{L}^*$ schneiden sich nicht.

B) Kein direkter Leader in $\mathcal{L}^*$ kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Annahme es gäbe einen solchen Leader:

Kürzere Gesamtlänge der Leader nach Neuverdrahtung

↯ Minimalität von $\mathcal{L}^*$

Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in $\mathcal{L}^*$ schneiden sich nicht.

B) Kein direkter Leader in $\mathcal{L}^*$ kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Absteigende Kreuzung: Beteiligung eines absteigenden Leader.

Aufsteigende Kreuzung: Beteiligung eines aufsteigenden Leader.

Menge absteigender Kreuzungen und Menge aufsteigender Kreuzungen sind somit disjunkt.

Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in $\mathcal{L}^*$ schneiden sich nicht.

B) Kein direkter Leader in $\mathcal{L}^*$ kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Idee:

Entferne alle aufsteigenden Kreuzungen $\mathcal{L}^*$, sodass

- die Gesamtlänge der Leader unverändert bleibt und
- keine absteigenden Kreuzungen entstehen.

Selbes Vorgehen für absteigende Kreuzungen durch Spiegeln der Instanz.

Kreuzungsfreie Zuordnung $\mathcal{L}$ mit gleicher Länge wie $\mathcal{L}^*$.

Neuverdrahtung

Fortsetzung Beweis:

Algorithmus zur Entferung aller aufsteigenden Kreuzungen:

Sortiere Label von unten nach oben: $\ell_1, \ldots, \ell_n$

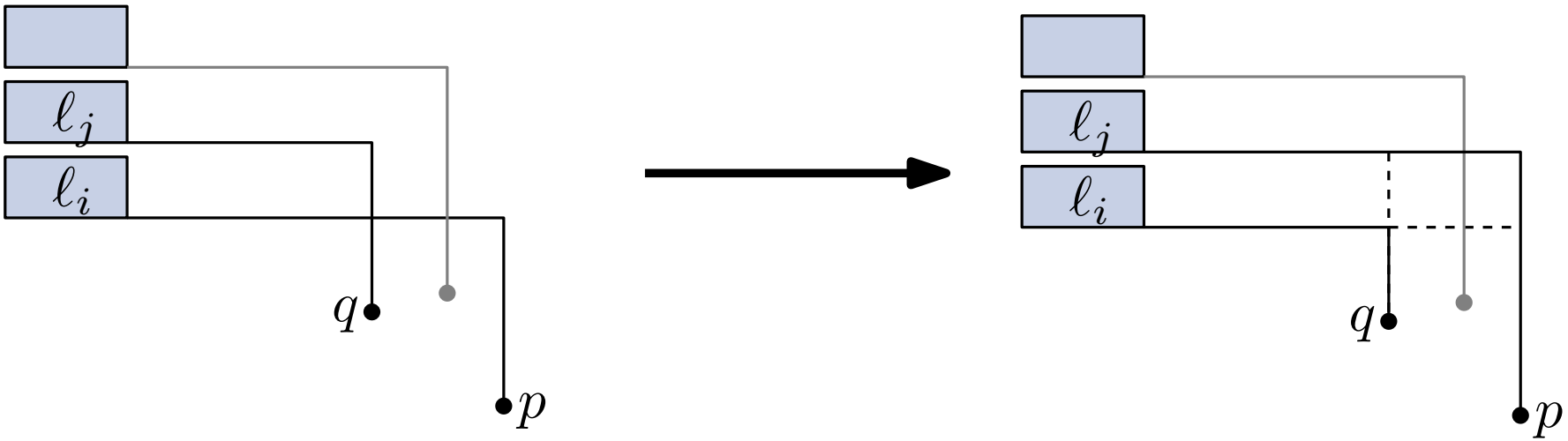
für $i = 1 \ldots n$ **tue**

Sei $\lambda(p, \ell_i)$ Leader in $\mathcal{L}^*$.

wenn $\lambda(p, \ell_i)$ aufsteigender Leader ist **dann**

Bestimme leader $\lambda(q, \ell_j)$ mit linkester Kreuzung mit $\lambda(p, \ell_i)$.

Neuverdrahtung: $\lambda(q, \ell_i)$ und $\lambda(p, \ell_j)$.



Neuverdrahtung

Fortsetzung Beweis:

Algorithmus zur Entferung aller aufsteigenden Kreuzungen:

Sortiere Label von unten nach oben: $\ell_1, \ldots, \ell_n$
für $i = 1 \ldots n$ **tue**
  Sei $\lambda(p, \ell_i)$ Leader in $\mathcal{L}^*$.
  wenn $\lambda(p, \ell_i)$ aufsteigender Leader ist **dann**
    Bestimme leader $\lambda(q, \ell_j)$ mit linkester Kreuzung mit $\lambda(p, \ell_i)$.
    Neuverdrahtung: $\lambda(q, \ell_i)$ und $\lambda(p, \ell_j)$.

Zeige für Schritt i:

1. Gesamtlänge der Leader bleibt erhalten.
2. Aufsteigende Kreuzungen entstehen nur oberhalb von Arm von $\lambda(p, \ell_i)$.
3. Es entstehen keine absteigenden Kreuzungen.
4. Laufzeit beträgt $O(n)$ Zeit.

Beweis durch Induktion über i: siehe Tafel.

Neuverdrahtung

Satz 1: Für jede Zuordnung $\mathcal{L}^*$ mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung $\mathcal{L}$ mit po-Leadern selber Länge. $\mathcal{L}$ kann mithilfe von $\mathcal{L}^*$ in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

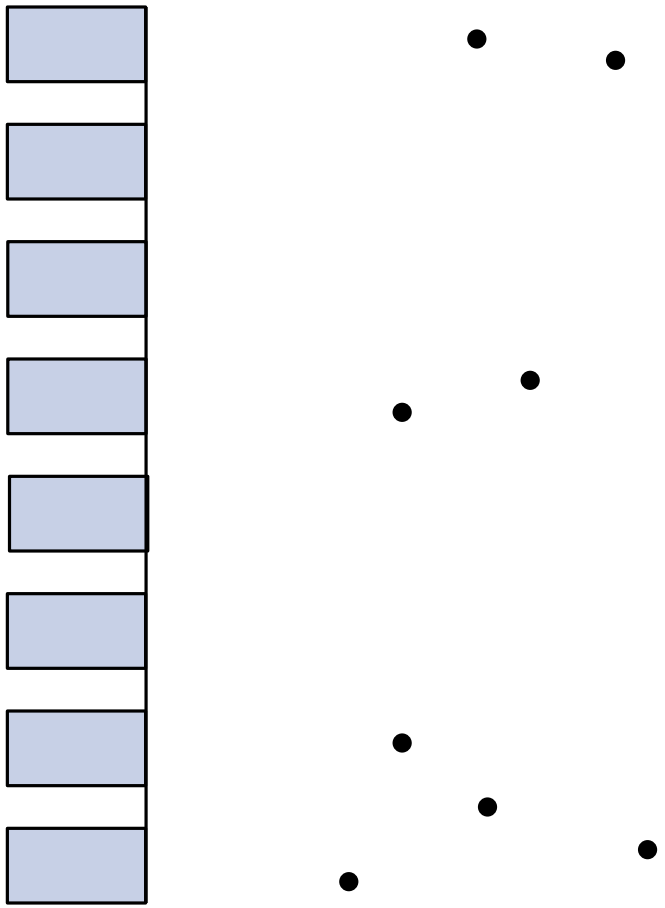
Satz kann auch auf do-Leader angepasst werden.

Sweep-Line-Verfahren

Jetzt: Verfahren für Minimierung der Leader-Längen in $O(n \log n)$ Zeit.

Vorgehen in zwei Phasen:

1. Teile Instanz in unabhängige Teilinstanzen auf.
2. Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

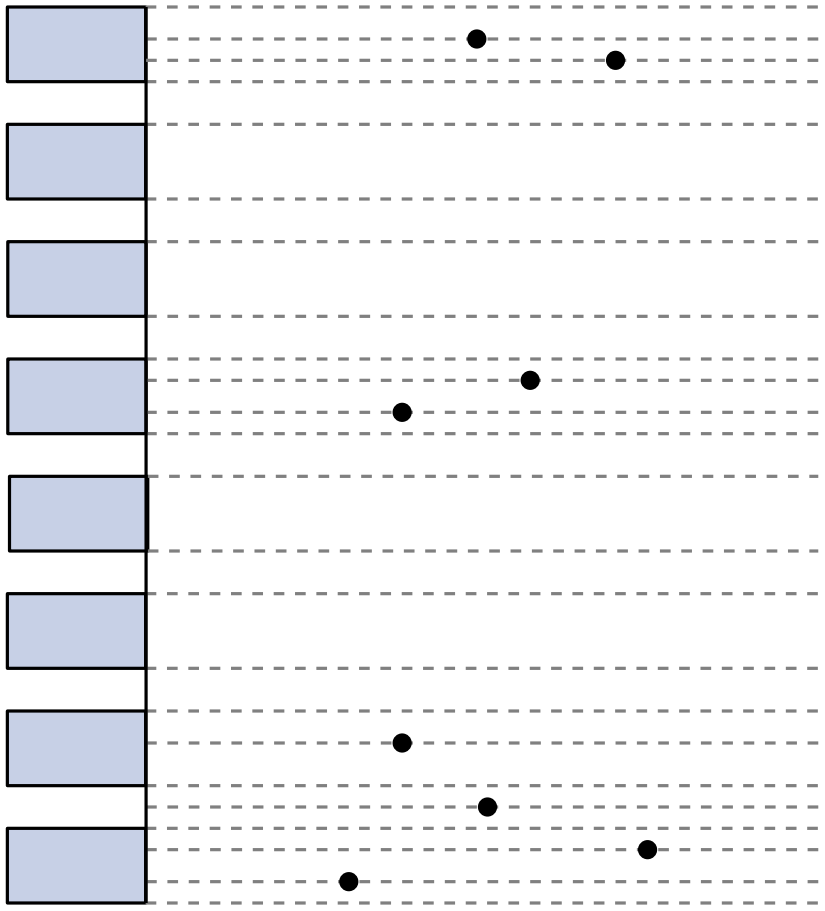


Sweep-Line-Verfahren

Jetzt: Verfahren für Minimierung der Leader-Längen in $O(n \log n)$ Zeit.

Vorgehen in zwei Phasen:

1. Teile Instanz in unabhängige Teilinstanzen auf.
2. Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

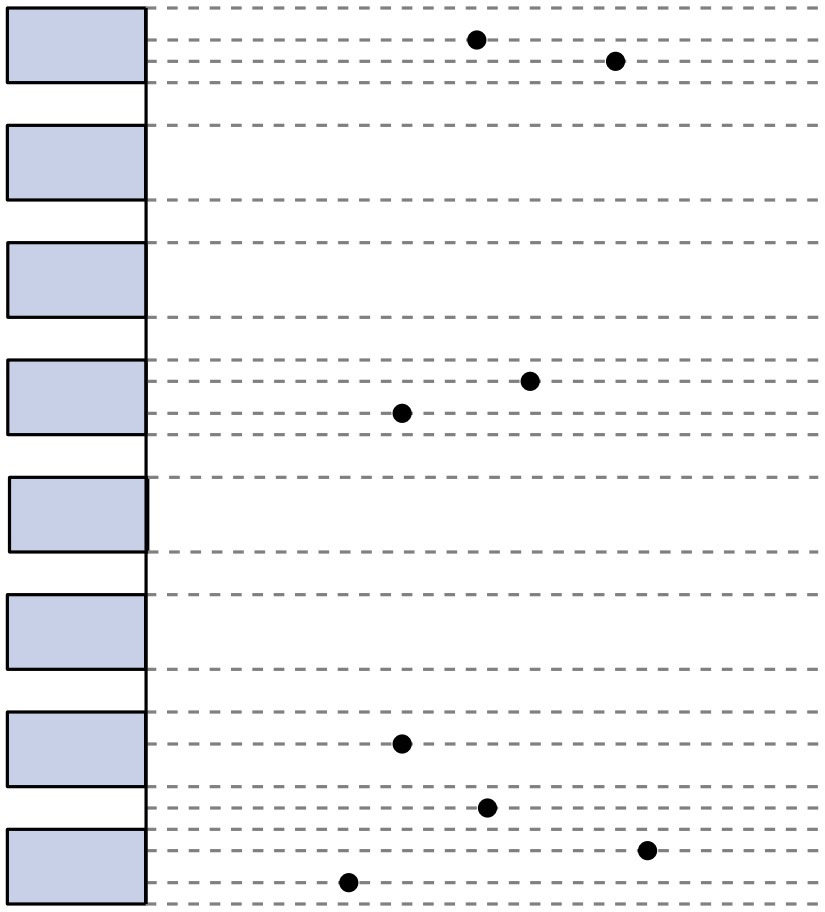


Unterteile hierzu Instanz in Streifen induziert durch Punkte in P und horizontale Seiten der Label.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

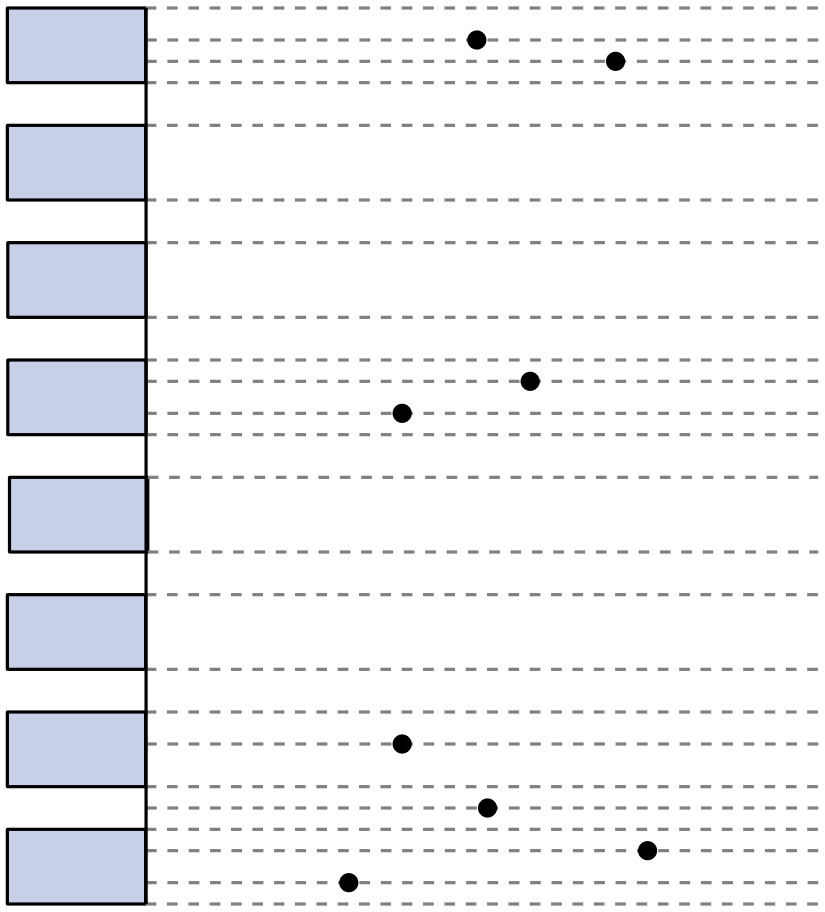
pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > \ell a_\sigma$

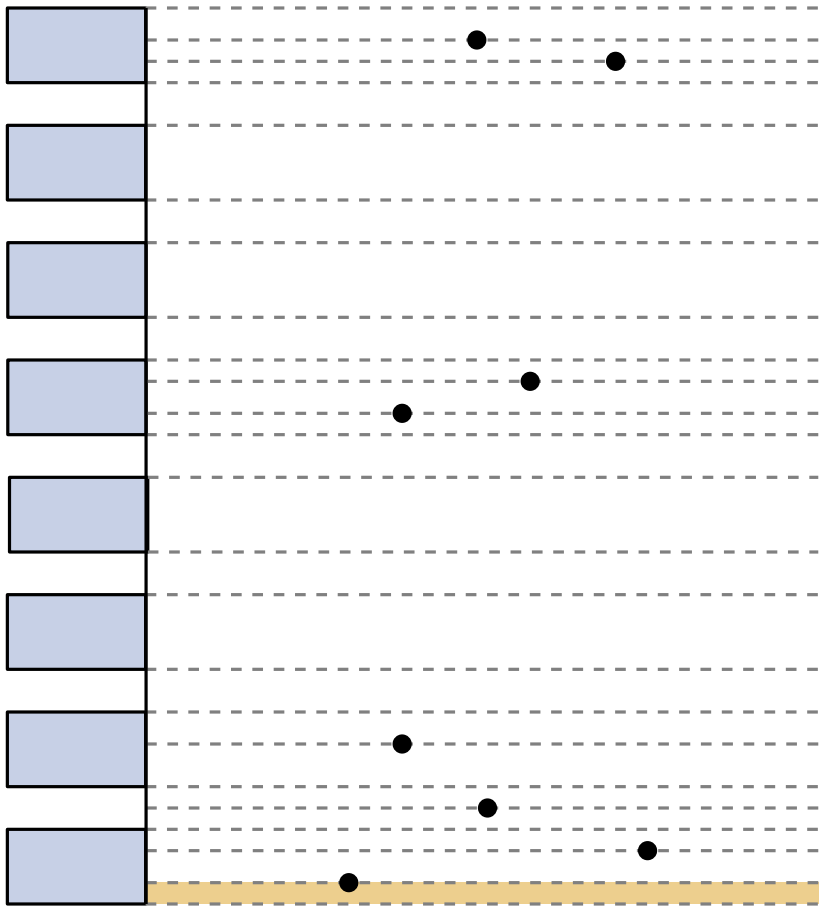
aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > \ell a_\sigma$

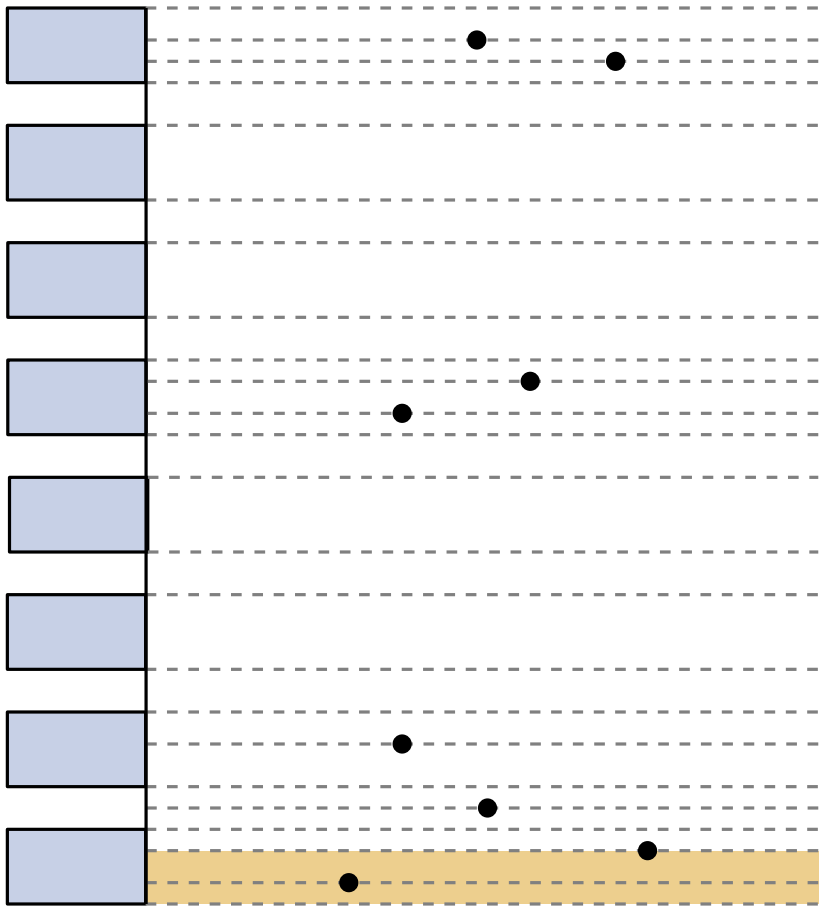
aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > \ell a_\sigma$

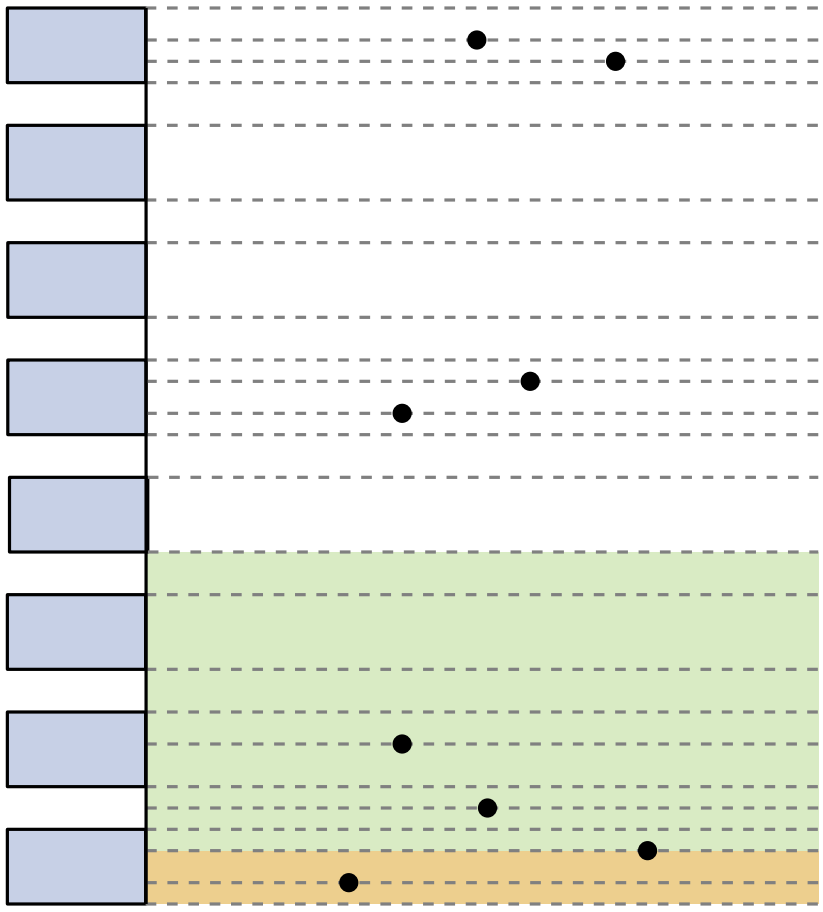
aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > \ell a_\sigma$

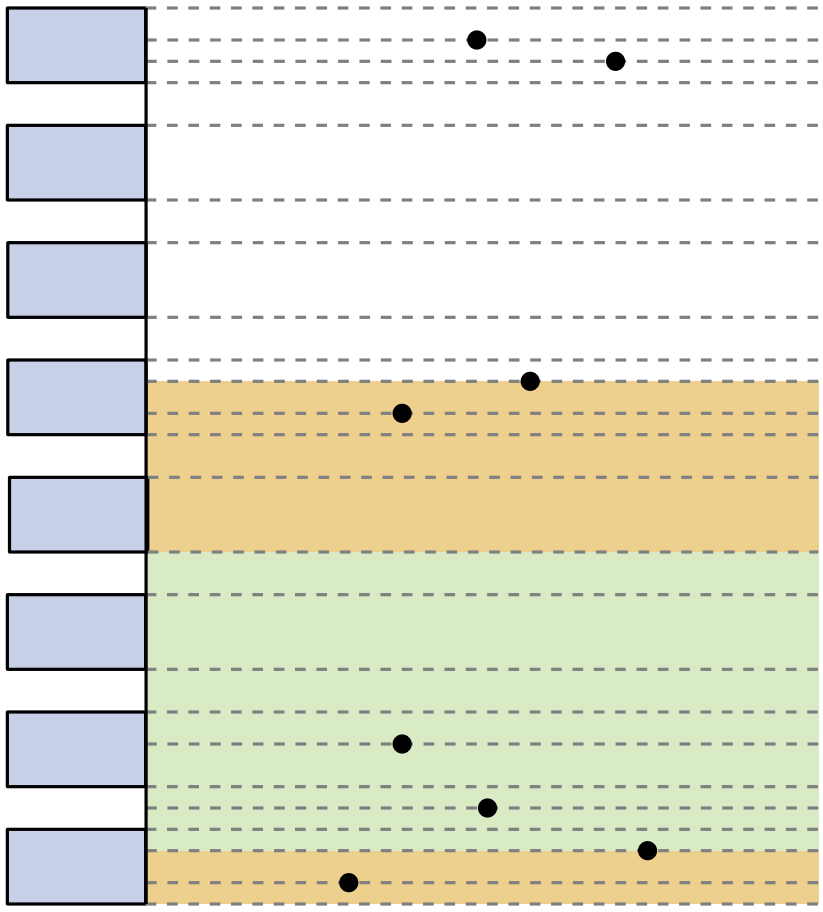
aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > \ell a_\sigma$

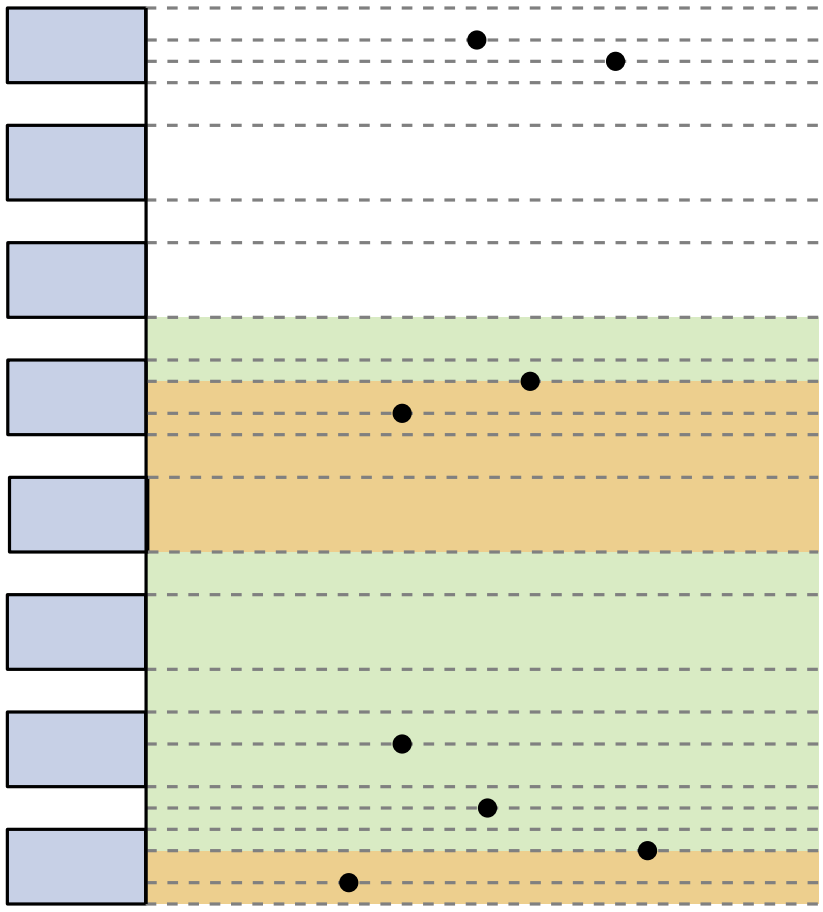
aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > \ell a_\sigma$

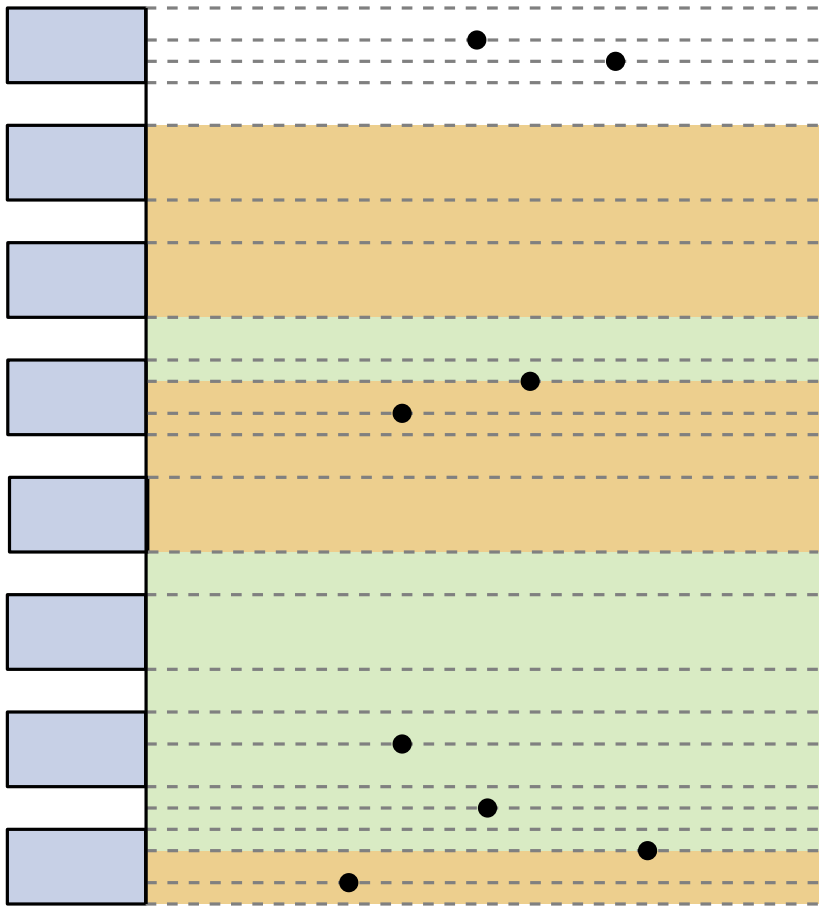
aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > \ell a_\sigma$

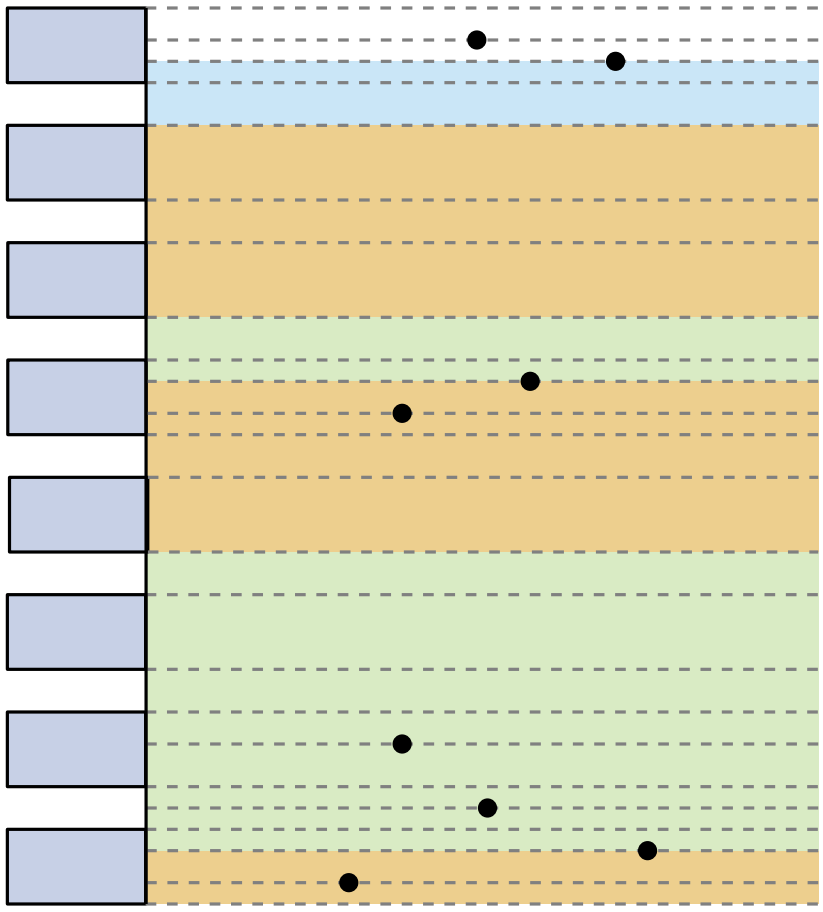
aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > \ell a_\sigma$

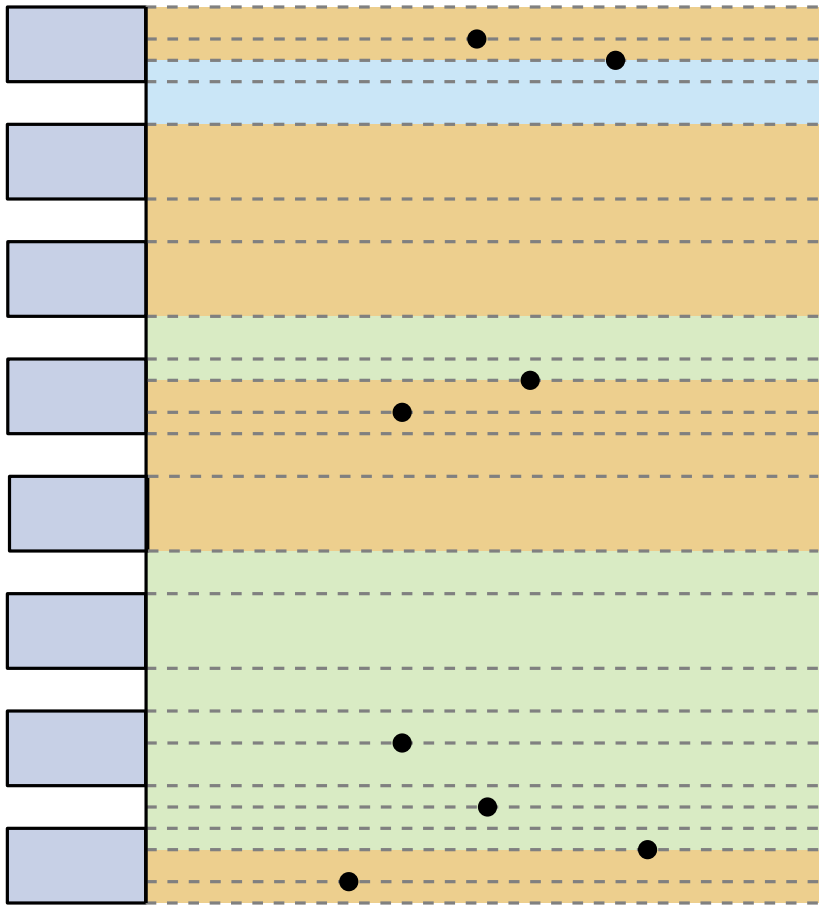
aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ:



pa_σ =#Punkte oberhalb von σ inkl. Punkte auf oberer Kante von σ.

ℓa_σ =#Label oberhalb von σ inkl. Label die σ schneiden.

pb_σ =#Punkte unterhalb von σ inkl. Punkte auf unterer Kante von σ.

ℓb_σ =#Label unterhalb von σ inkl. Label die σ schneiden.

Ein Streifen σ heißt:

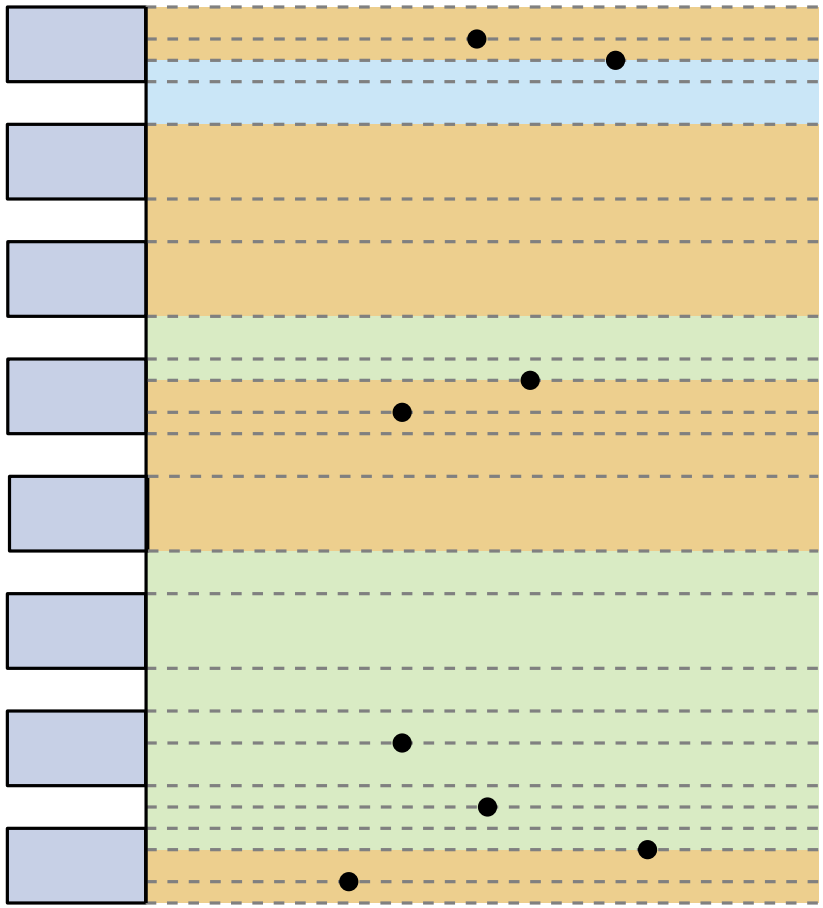
absteigend, falls $pa_\sigma > \ell a_\sigma$

aufsteigend, falls $pb_\sigma > \ell b_\sigma$

neutral, in allen anderen Fällen.

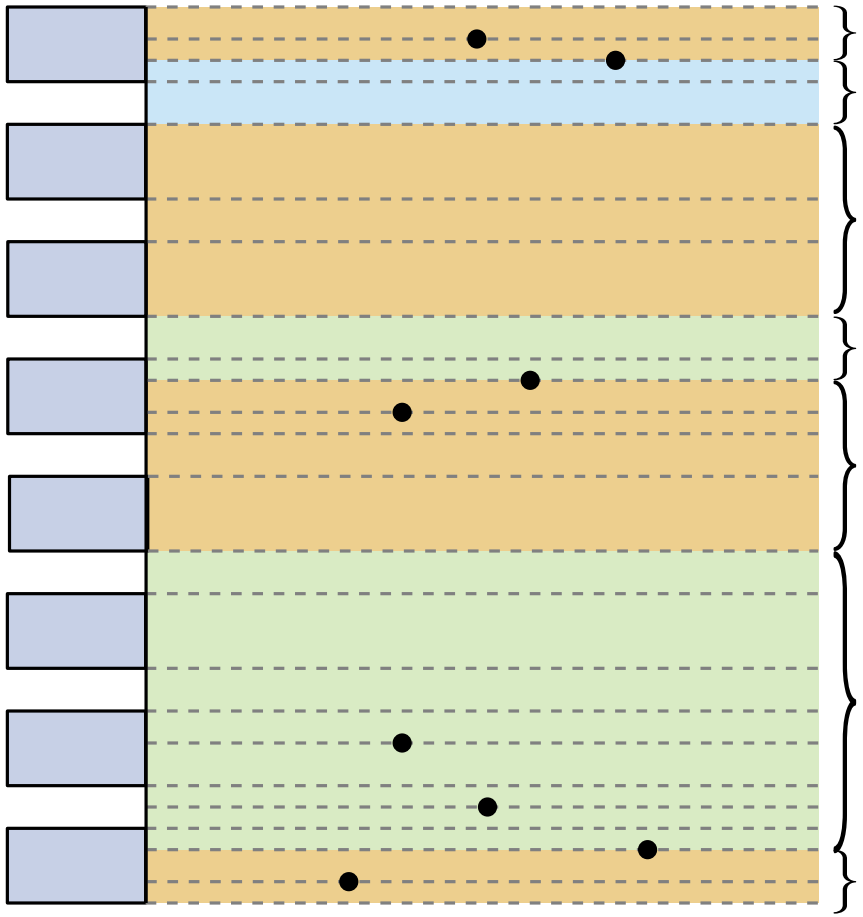
Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.



Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.



absteigende Instanz: inklusive der Grenzen

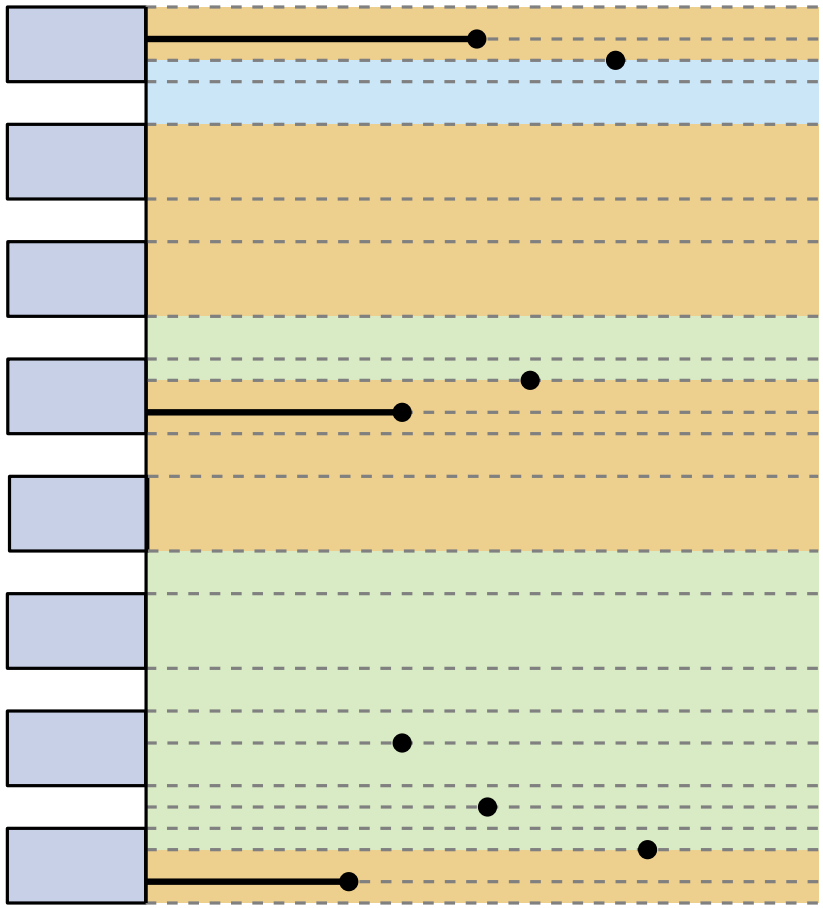
neutrale Instanz: exklusive der Grenzen

aufsteigende Instanz: inklusive der Grenzen

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

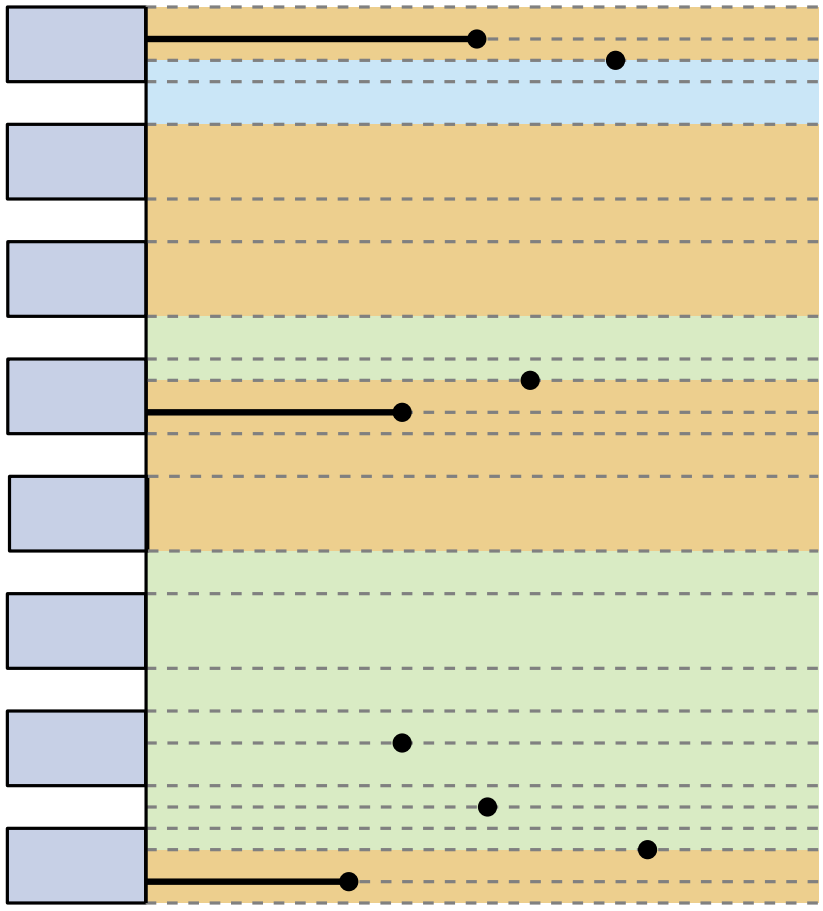


Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

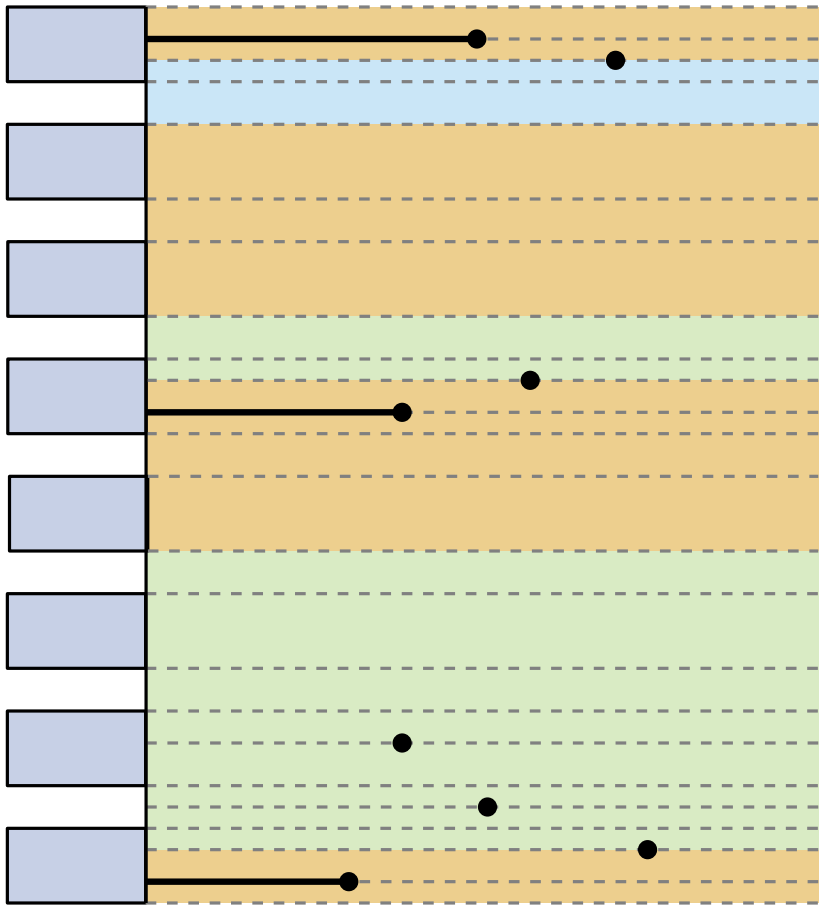
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

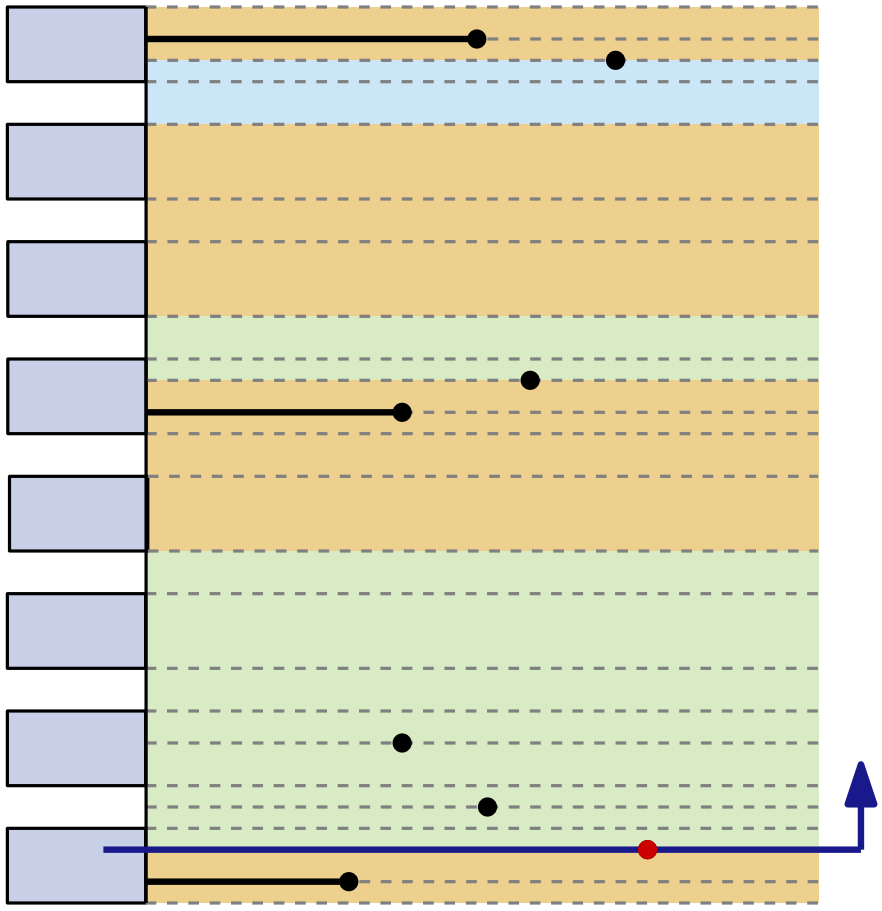
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

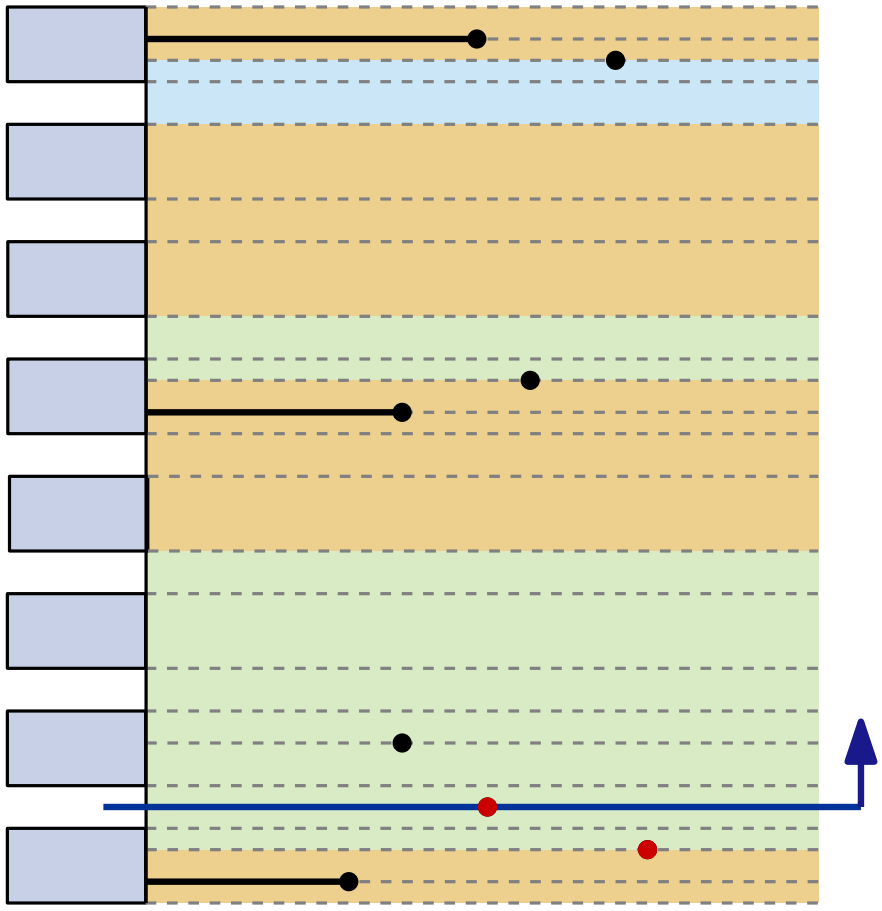
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

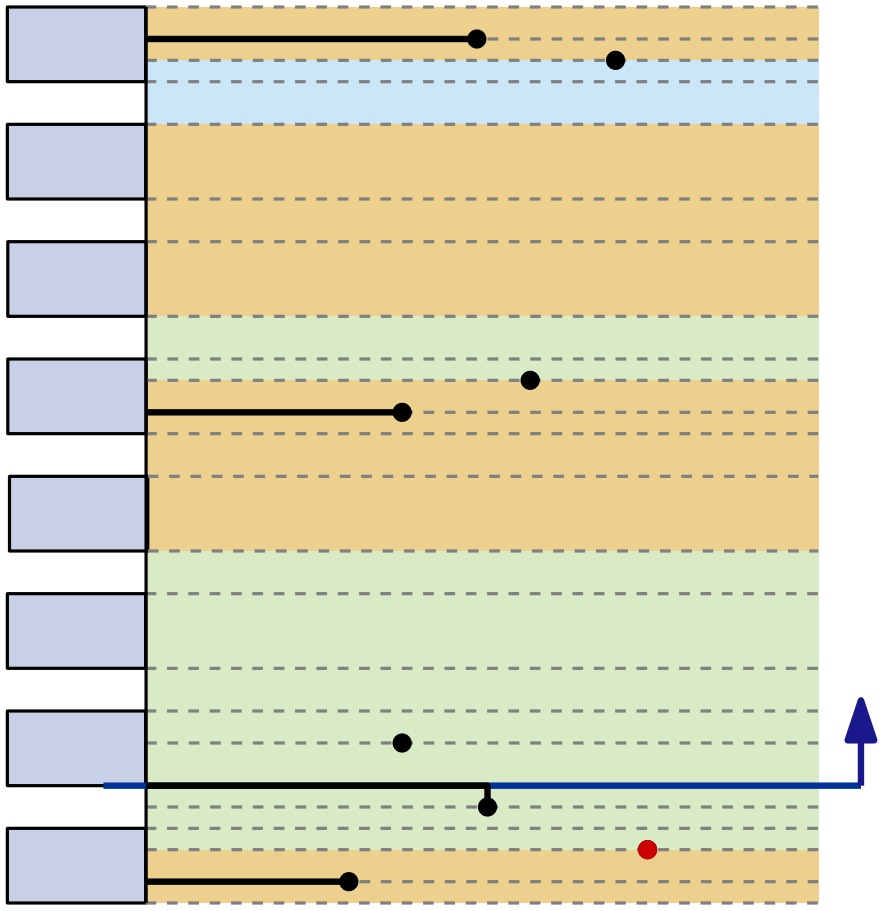
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

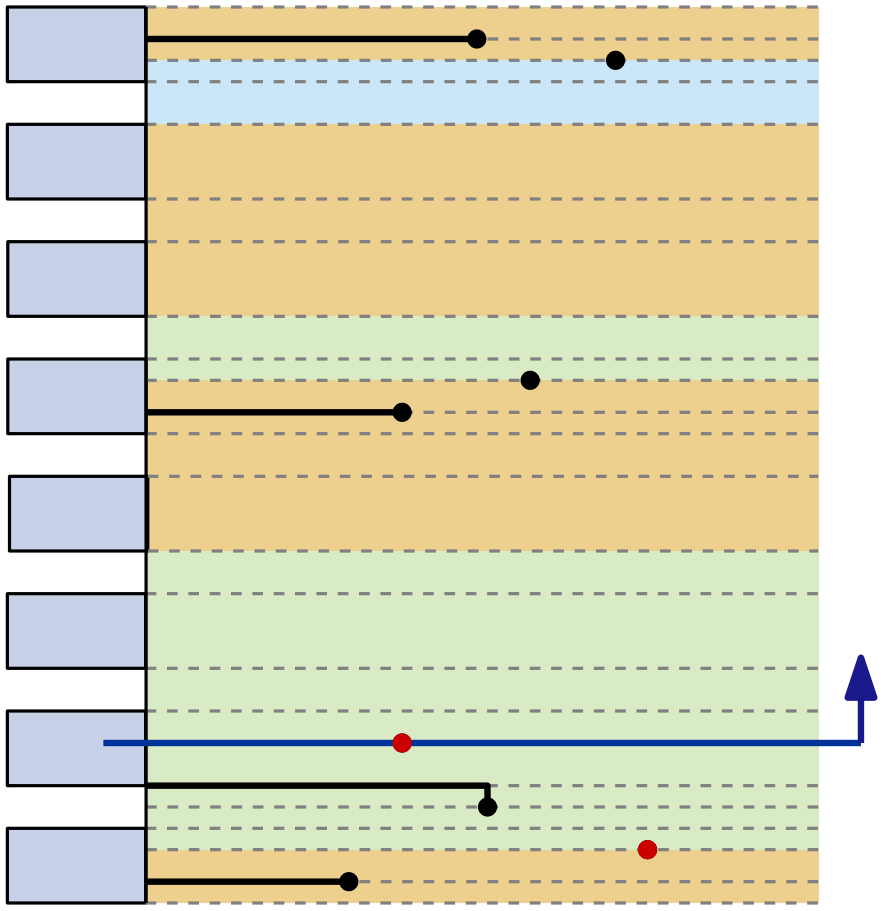
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

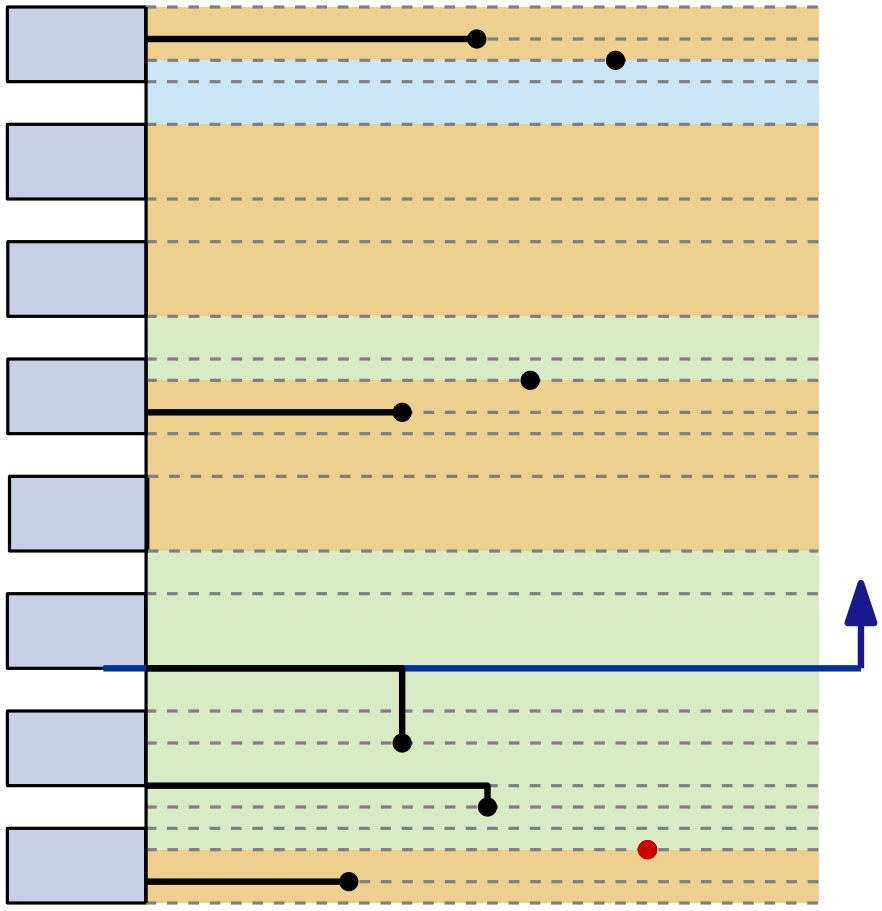
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

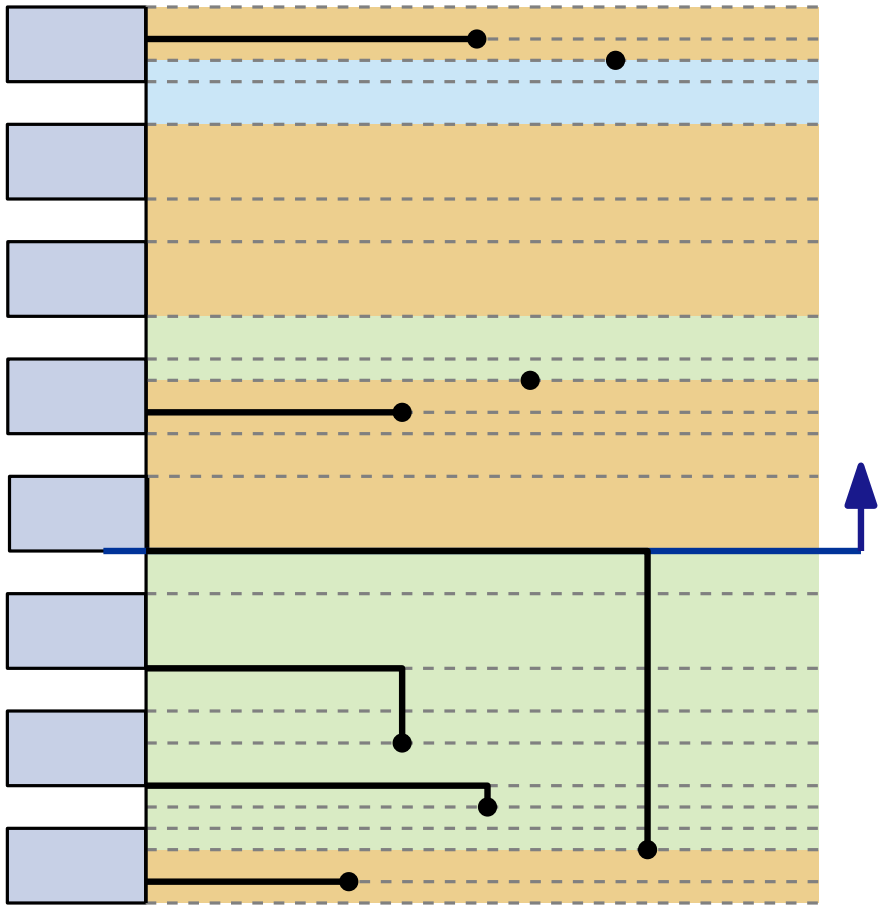
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

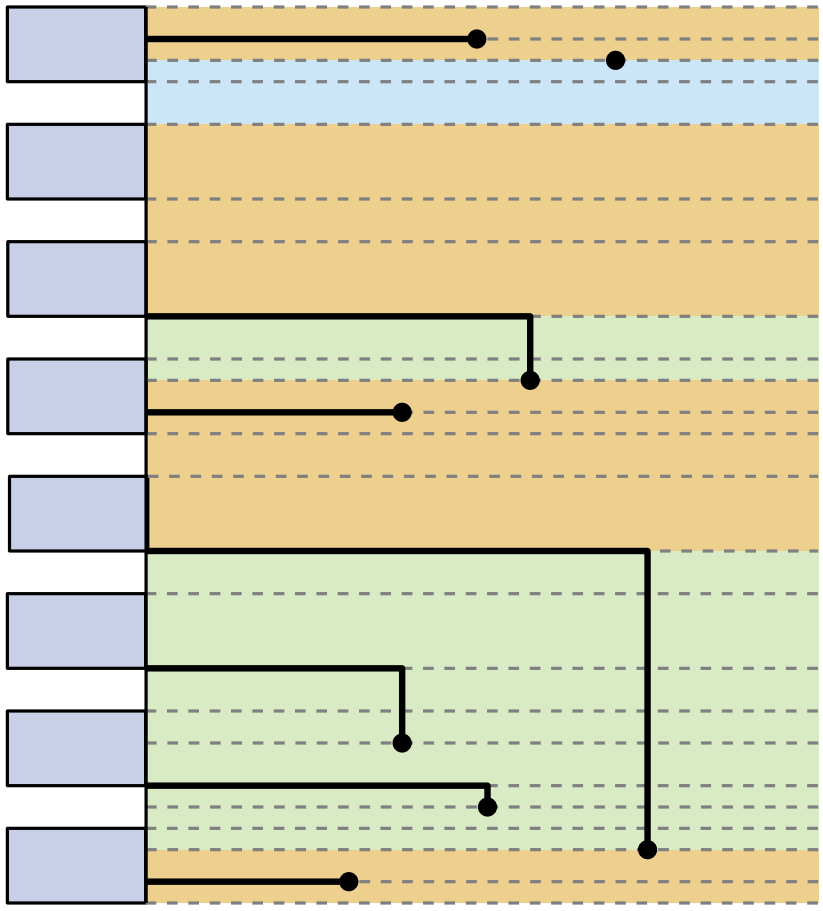
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

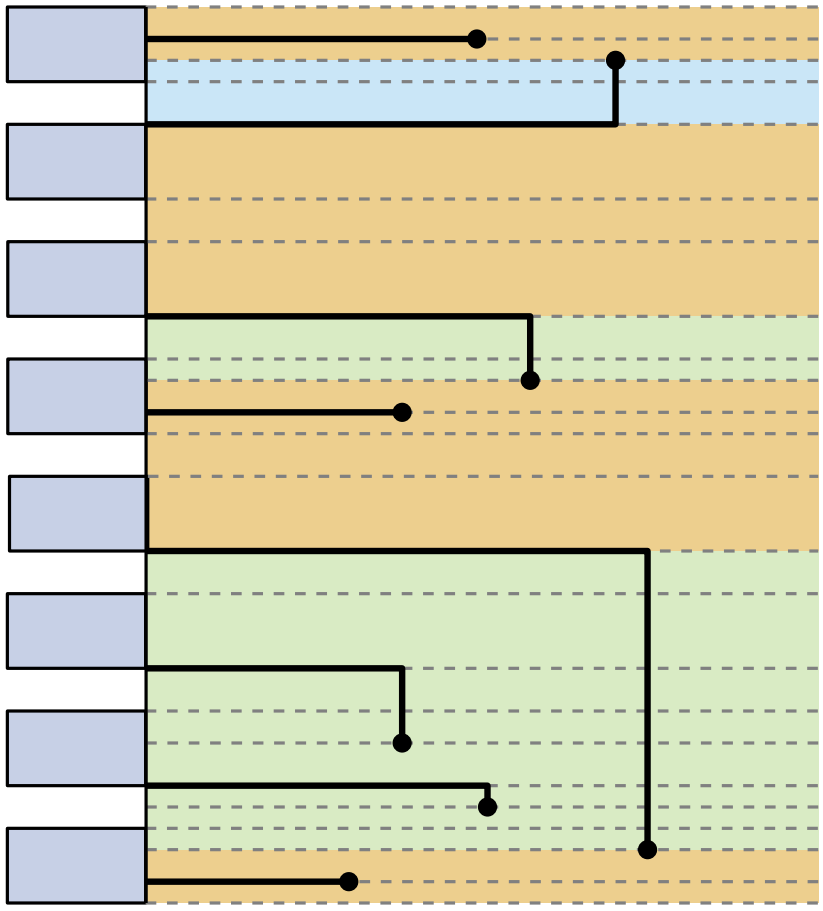
Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p:
Füge p zu einer nach x-Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ:
Entferne Punkt p aus W, der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Analoges Vorgehen für absteigende Instanz.

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > \ell a_\sigma$ → $pa_\sigma - \ell a_\sigma \geq 1$ und $pb_\sigma < \ell b_\sigma$

aufsteigend, falls $pb_\sigma > \ell b_\sigma$ → $pb_\sigma - \ell b_\sigma \geq 1$ und $pa_\sigma < \ell a_\sigma$

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > \ell a_\sigma$ → $pa_\sigma - \ell a_\sigma \geq 1$ und $pb_\sigma < \ell b_\sigma$

aufsteigend, falls $pb_\sigma > \ell b_\sigma$ → $pb_\sigma - \ell b_\sigma \geq 1$ und $pa_\sigma < \ell a_\sigma$

Warum?

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > \ell a_\sigma$ → $pa_\sigma - \ell a_\sigma \geq 1$ und $pb_\sigma < \ell b_\sigma$

aufsteigend, falls $pb_\sigma > \ell b_\sigma$ → $pb_\sigma - \ell b_\sigma \geq 1$ und $pa_\sigma < \ell a_\sigma$

$pa_{\sigma\downarrow} - \ell a_{\sigma\downarrow} \geq 1$

$pa_{\sigma\uparrow} - \ell a_{\sigma\uparrow} < 0$

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > \ell a_\sigma$ $\longrightarrow$ $pa_\sigma - \ell a_\sigma \geq 1$ und $pb_\sigma < \ell b_\sigma$

aufsteigend, falls $pb_\sigma > \ell b_\sigma$ $\longrightarrow$ $pb_\sigma - \ell b_\sigma \geq 1$ und $pa_\sigma < \ell a_\sigma$

$pa_{\sigma\downarrow} - \ell a_{\sigma\downarrow} \geq 1$

$pa_{\sigma\uparrow} - \ell a_{\sigma\uparrow} < 0$

$\longrightarrow$ $|(pa_{\sigma\downarrow} - \ell a_{\sigma\downarrow}) - (pa_{\sigma\uparrow} - \ell a_{\sigma\uparrow})| \geq 2$

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > \ell a_\sigma$ $\longrightarrow$ $pa_\sigma - \ell a_\sigma \geq 1$ und $pb_\sigma < \ell b_\sigma$

aufsteigend, falls $pb_\sigma > \ell b_\sigma$ $\longrightarrow$ $pb_\sigma - \ell b_\sigma \geq 1$ und $pa_\sigma < \ell a_\sigma$

$$pa_{\sigma\downarrow} - \ell a_{\sigma\downarrow} \geq 1$$
$$pa_{\sigma\uparrow} - \ell a_{\sigma\uparrow} < 0$$
$$\longrightarrow \quad |(pa_{\sigma\downarrow} - \ell a_{\sigma\downarrow}) - (pa_{\sigma\uparrow} - \ell a_{\sigma\uparrow})| \geq 2$$

Da Punkte und Label in allgemeiner Lage gilt aber für zwei benachbarte Streifen σ und σ':

$$|(pa_\sigma - \ell a_\sigma) - (pa_{\sigma'} - \ell a_{\sigma'})| \leq 1$$

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > \ell a_\sigma$ $\longrightarrow$ $pa_\sigma - \ell a_\sigma \geq 1$ und $pb_\sigma < \ell b_\sigma$

aufsteigend, falls $pb_\sigma > \ell b_\sigma$ $\longrightarrow$ $pb_\sigma - \ell b_\sigma \geq 1$ und $pa_\sigma < \ell a_\sigma$

$pa_{\sigma\downarrow} - \ell a_{\sigma\downarrow} \geq 1$

$pa_{\sigma\uparrow} - \ell a_{\sigma\uparrow} < 0$

$$\longrightarrow \quad |(pa_{\sigma\downarrow} - \ell a_{\sigma\downarrow}) - (pa_{\sigma\uparrow} - \ell a_{\sigma\uparrow})| \geq 2$$

Da Punkte und Label in allgemeiner Lage gilt aber für zwei benachbarte Streifen σ und σ':

$$|(pa_\sigma - \ell a_\sigma) - (pa_{\sigma'} - \ell a_{\sigma'})| \leq 1$$

$\sigma\downarrow$ und $\sigma\uparrow$ sind nicht benachbart.

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

⟶ Wegen Lemma 1: In jeder optimalen Lösung werden die Punkte einer aufsteigenden/absteigenden Instanz S von Leadern angebunden, die vollständig in S liegen.

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

⟶ Wegen Lemma 1: In jeder optimalen Lösung werden die Punkte einer aufsteigenden/absteigenden Instanz S von Leadern angebunden, die vollständig in S liegen.

2. Sei σ unterster Streifen einer aufsteigenden Instanz S. In jeder optimalen Lösung werden ausschließlich Label oberhalb von σ zur Beschriftung der Punkte in S verwendet (siehe Tafel). Analog: Absteigende Instanz.

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma\downarrow$ und aufsteigenden Streifen $\sigma\uparrow$ befindet sich immer ein neutraler Streifen.

⟶ Wegen Lemma 1: In jeder optimalen Lösung werden die Punkte einer aufsteigenden/absteigenden Instanz S von Leadern angebunden, die vollständig in S liegen.

2. Sei σ unterster Streifen einer aufsteigenden Instanz S. In jeder optimalen Lösung werden ausschließlich Label oberhalb von σ zur Beschriftung der Punkte in S verwendet (siehe Tafel). Analog: Absteigende Instanz.

3. Der Algorithmus berechnet eine solche Zuordnung und diese ist optimal und kreuzungsfrei (siehe Tafel).

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Lemma 3: Eine kreuzungsfreie Zuordnung minimaler Länge kann nicht schneller als in $\Omega(n \log n)$ Zeit berechnet werden (s. Übung).

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Lemma 3: Eine kreuzungsfreie Zuordnung minimaler Länge kann nicht schneller als in $\Omega(n \log n)$ Zeit berechnet werden (s. Übung).

Satz 2: Die Berechnung einer kreuzungsfreien Zuordnung minimaler Länge einer Instanz (P, L, R) mit po-Leadern benötigt Laufzeit $\Theta(n \log n)$ Zeit und $\Theta(n)$ Speicher.