Übung Algorithmische Kartografie

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

Benjamin Niedermann

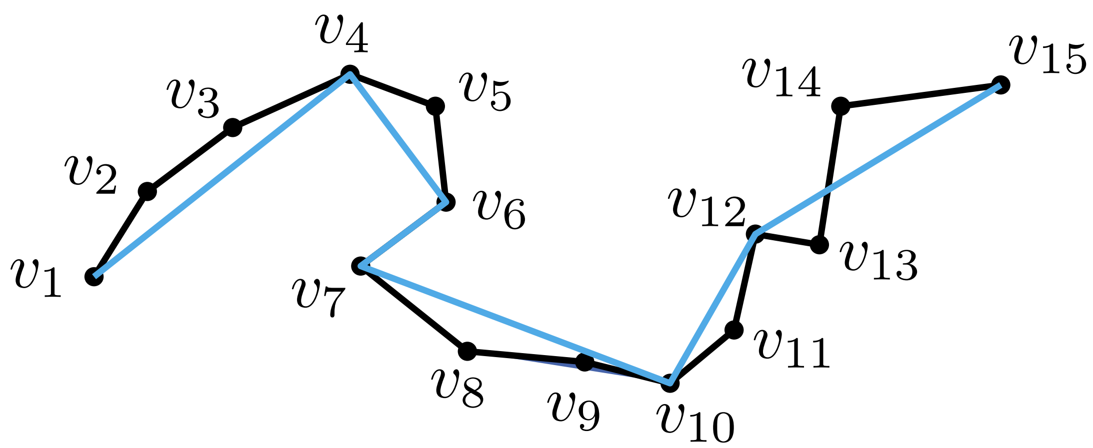
18.07.2013

Wiederholung

Themen der Vorlesung:

- Linienvereinfachung (VL 01+02)
- Linienschematisierung (VL 03)
- Punktbeschriftung (VL 04+05)
- Randbeschriftung (VL 05+06)
- Beschriftung in dynamischen Karten (VL 07–09)
- Proportional Symbol Maps (VL 09)
- Flächenkartogramme (VL 10+11)
- Flächenaggregation (VL 12)
- Algorithmen in der Praxis (VL 13)

Linienvereinfachung

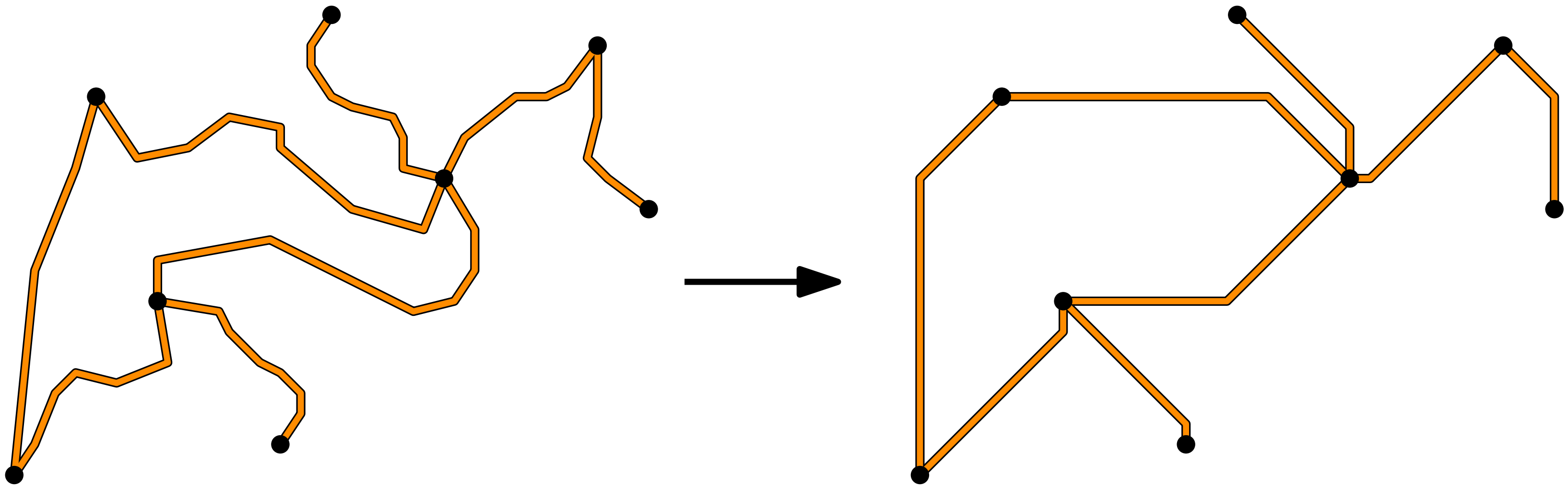


Geg: Pfad P

Ges: Approximation/Ausdünnung Q von P

- einfache lokale Verfahren
- Douglas-Peucker Algorithmus
- Beschleunigung von Hershberger/Snoeyink
- Algorithmus von Visvalingam/Whyatt
- Formulierung als Optimierungsproblem auf Graphen
- konsistente Vereinfachung ganzer Unterteilungen (Tangentensegmente, Punktzuweisung, konsistente Shortcuts)

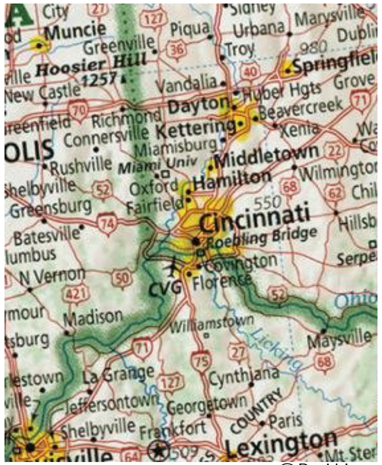
Linienschematisierung



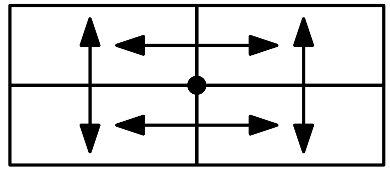
Geg: Straßenkarte M als Menge von Pfaden
Ges: topologisch äquivalente Schematisierung von M mit beschränkten Kantenrichtungen

- Verfahren von Cabello et al.
- mehrstufiges Vorgehen via Rektifizierung, kanonische Form, Bestimmung Totalordnung der Pfade, Einfügen nach dieser Ordnung

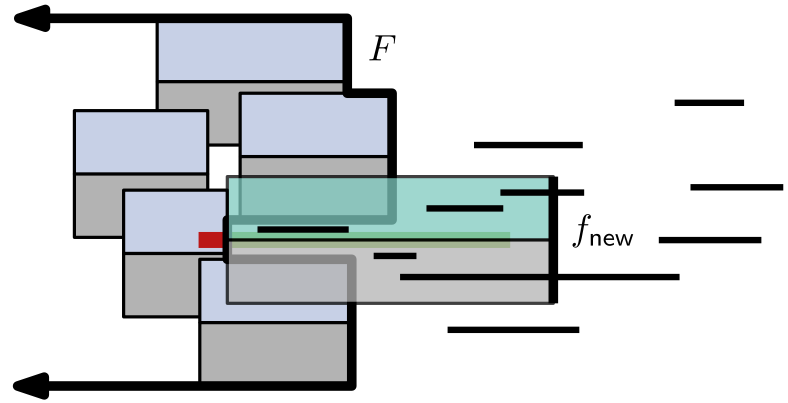
Punktbeschriftung (Slider Modell)



©David Imus



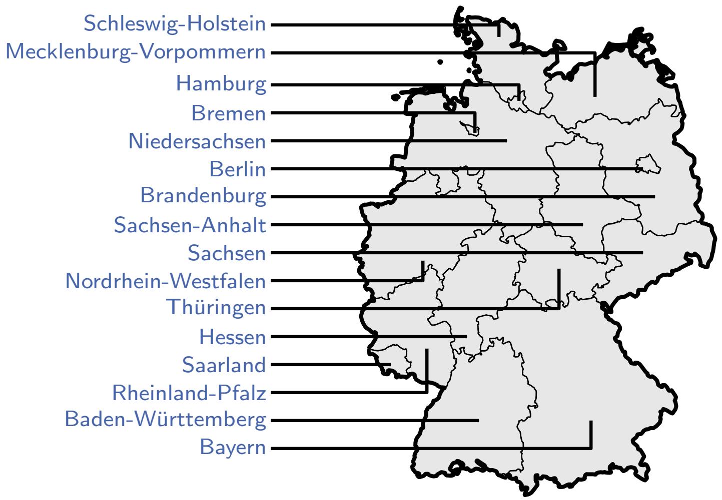
4S

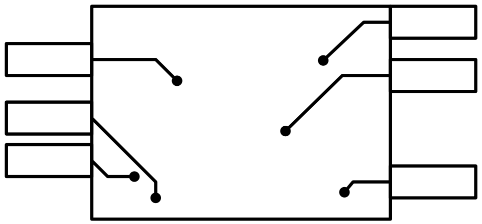


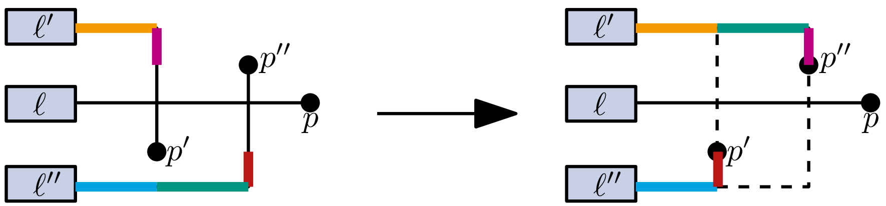
Geg: Punktmenge P und Labelmenge L
Ges: konfliktfreie Auswahl und Platzierung einer mögl. großen Teilmenge von L

- 4-Slider Modell
- NP-Schwere via Planares 3-Sat
- $1/2$-Approximationsalgorithmus
- $O(n \log n)$ Laufzeit durch (geometrische) Datenstrukturen: red-black Trees, Heaps, Priority Search Trees

Randbeschriftungen







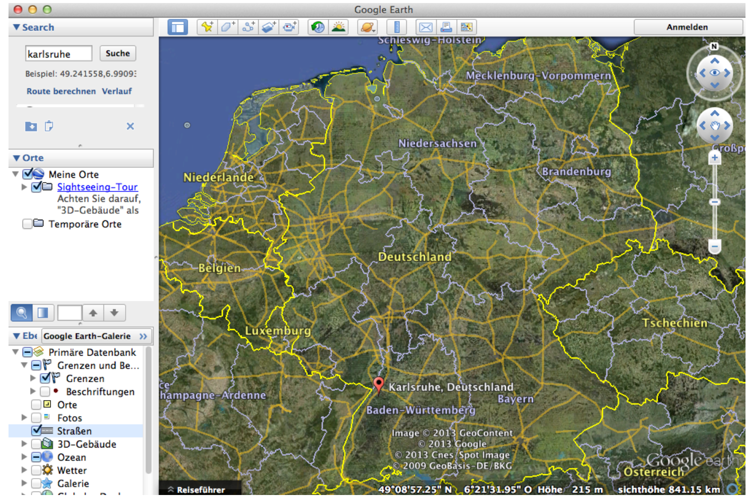
Geg: Punktmenge P und Labelmenge L

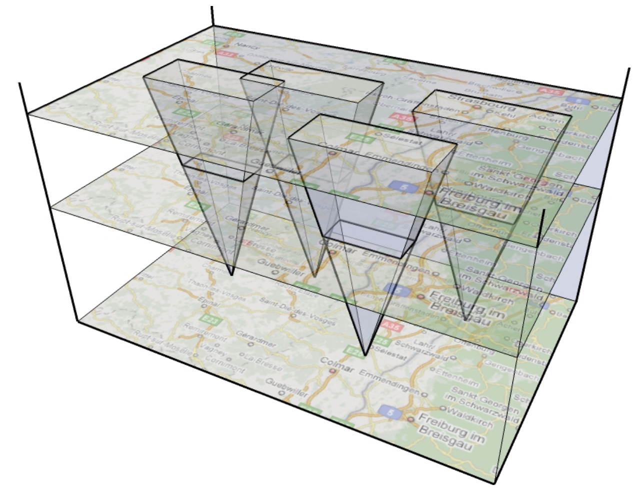
Ges: Platzierung von L am Rand der Zeichnung und Verbinden durch optimale kreuzungsfreie Leader

- **einseitiger Fall, po-Leader, Längenminimierung**
- $O(n^2)$ Laufzeit mit Neuverdrahtung
- $O(n \log n)$ Laufzeit mit Sweep-Verfahren
- **vierseitiger Fall, od- & pd-Leader, Längenminimierung**
- Lösung über gewichtetes bipartites Matching und Kreuzungsauflösung

Beschriftung in dynamischen Karten

Karlsruhe Institute of Technology

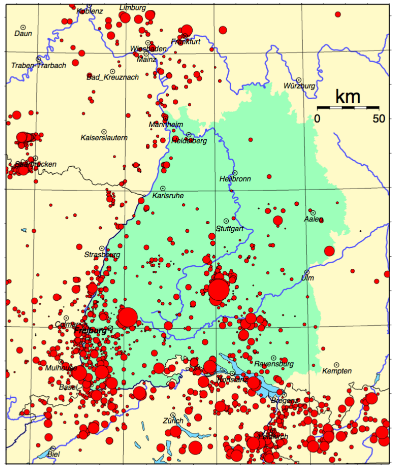


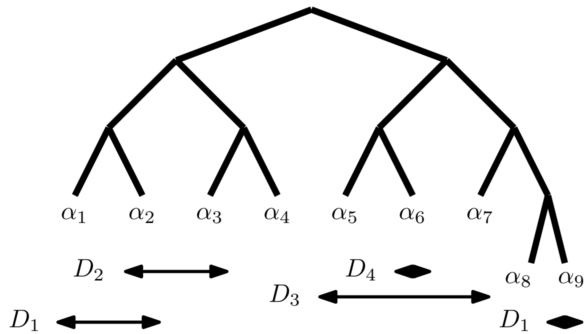


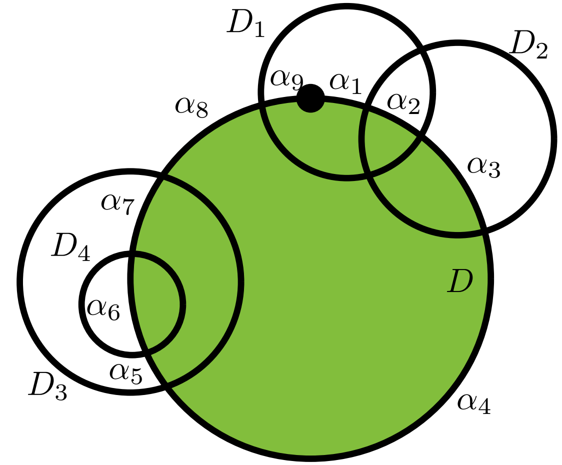
Geg: Punktmenge P und Labelmenge L
Ges: Konsistente Auswahl von $L' \subseteq L$ maximiert über alle möglichen Ansichten

- **dynamisches Zoomen**
- NP-Schwere (planares 3-Sat)
- zwei Approximationsalgorithmen (Greedy & ebenenbasiert)
- **dynamisches Rotieren**
- 1/4-Approximation und EPTAS mit line stabbing
- **Trajektorienbasiertes Beschriften**
- Betrachtung als Intervalle mit Konflikten, beschränkte Labelzahl

Proportional Symbol Maps





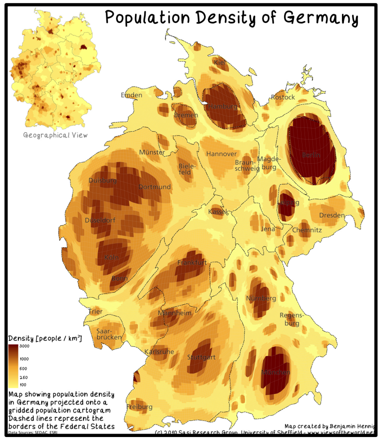


Geg: gewichtete Punktmenge P

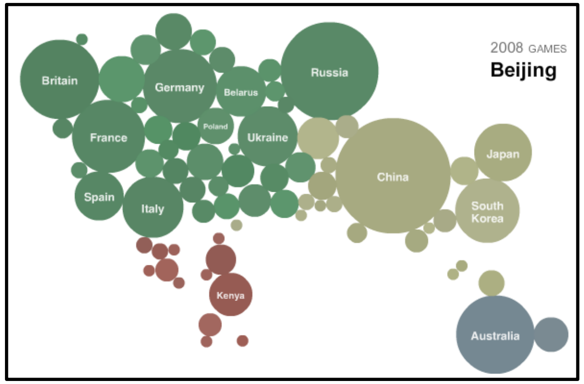
Ges: Repräsentation der Gewichte über Fläche von (überlappenden) Kreisen; maximiere minimal sichtbaren Rand

- Greedy Algorithmus
- effiziente Implementierung durch Nutzung von modifizierten Segment Trees

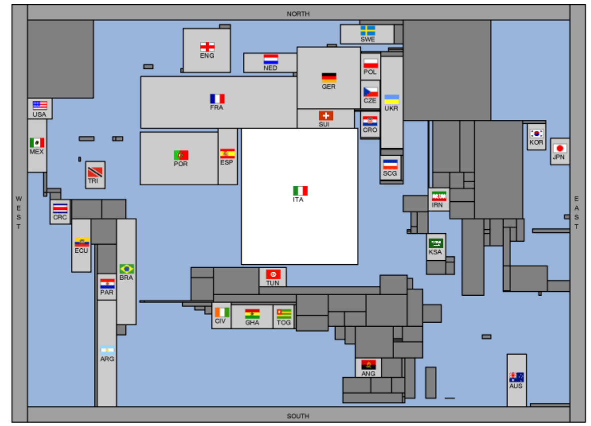
Flächenkartogramme



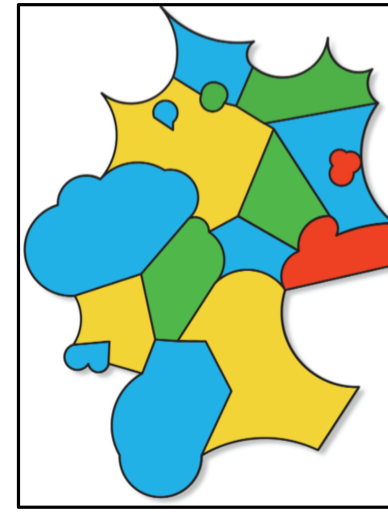
©Benjamin Hennig



©New York Times



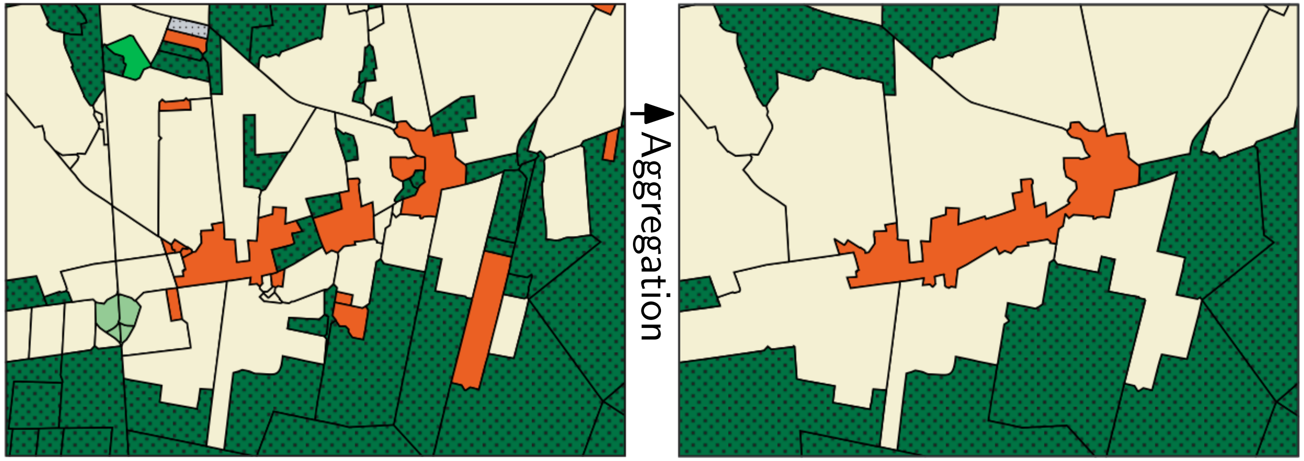
©Bettina Speckmann



Geg: politische Karte M mit gewichteten Regionen
Ges: topologisch äquivalente Karte, mit Flächen proportional zu Gewichten (*Flächenkartogramm*)

- physikalisches Diffusionsmodell
- Kreiskartogramme (heuristisch)
- Rechteckskartogramme (teils exakt, teils heuristisch)
- Kreisbogenkartogramme (heuristisch)
- optimale rektilineare 8-Eck-Kartogramme basierend auf T-Formen (mit Schnyder und kanonische Ordnung)

Flächenaggregation



Geg: Flächennutzungskarte M in Maßstab A

Ges: aggregierte Karte M' für Maßstab $B < A$

- NP-Schwere (planares Vertex Cover)
- minimiere Typänderungen und maximiere Kompaktheit
- drei Modelle via gemischt ganzzahlige Programmierung

Prüfungsablauf

Mündliche Prüfung (20 Minuten)

- Termine Einzelprüfungen: 7.8., 11.9., 16.10.
- Anmeldung im Sekretariat
- 5 LP für Stoff aus VL und Übung
- ggf. in Computergrafik mit 3 LP ohne Übung prüfbar

Inhaltlicher Schwerpunkt

- Verständnis der behandelten Probleme, Modelle, Algorithmen und Beweisideen
- Bewertung/Einschätzung, Auswahl von Alternativen

Vorbereitung

- Vorlesungsfolien, Mitschrieb, Literatur
- Sprechstunde (am besten per Mail anmelden)

Ausblick

weitere Vorlesungen/Seminare

- Algorithmen zur Visualisierung von Graphen, WS13/14
- Algorithmische Geometrie, SS14
- Seminar Algorithmentechnik WS 13/14

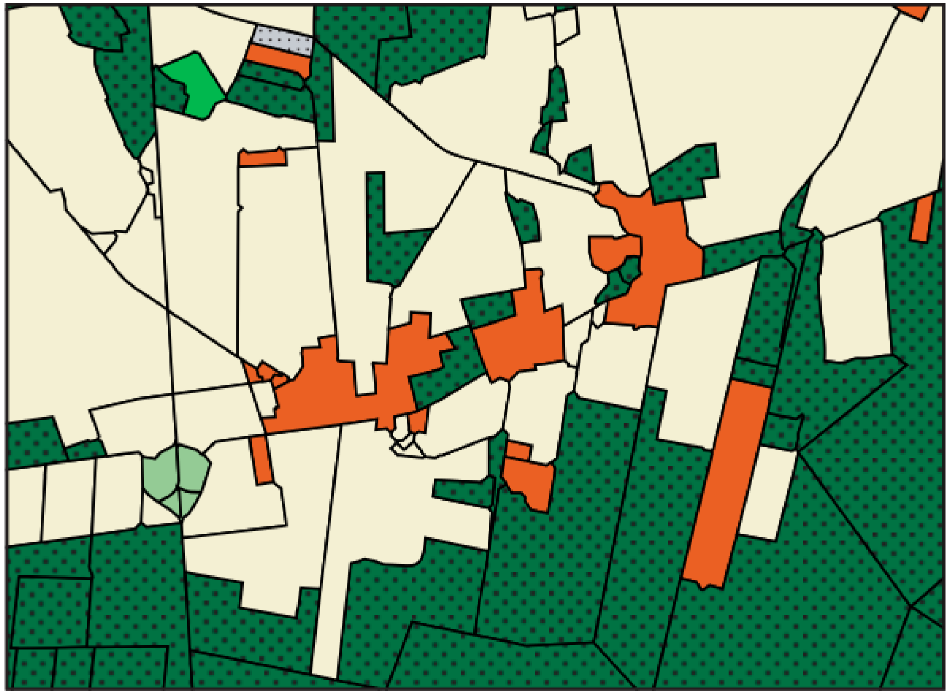
Masterarbeiten/HiWi-Jobs

- Algorithmische Kartografie (z.B. Labeling, Kartogramme, Schematisierung)
- Graphenvisualisierung
- geometrische Algorithmen
- sowohl theoretische als auch praktische Themen

$\rightarrow$ bei Interesse einfach uns ansprechen,
auch unabhängig von Ausschreibungen

Flächenaggregation

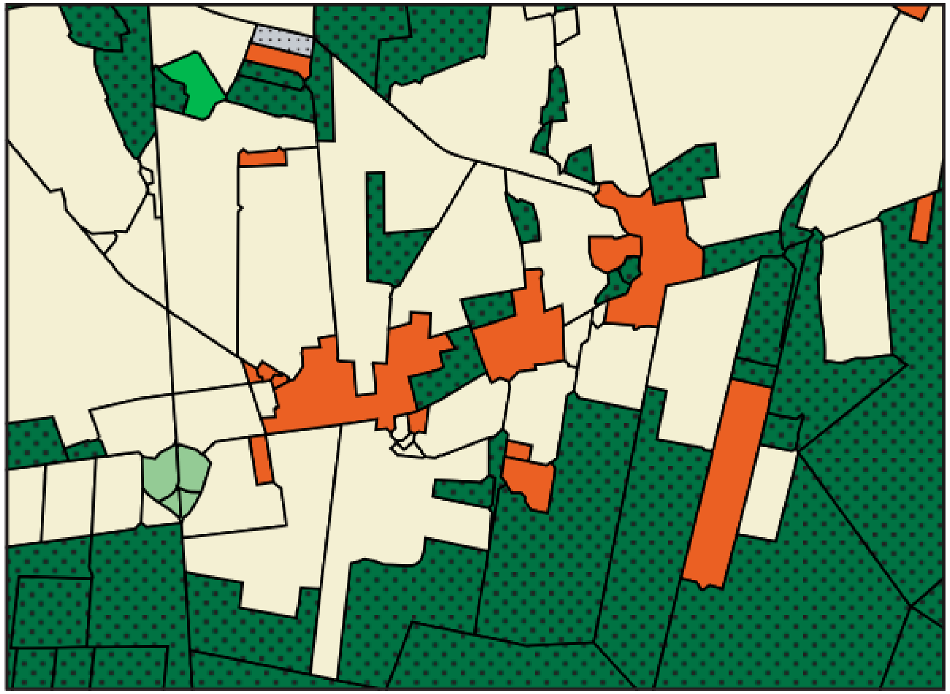
Flächenaggregation



Flächennutzung Maßstab 1:50.000

- Grassland
- Forest
- Farmland
- Settlement
- Industry

Flächenaggregation



Flächennutzung Maßstab 1:50.000

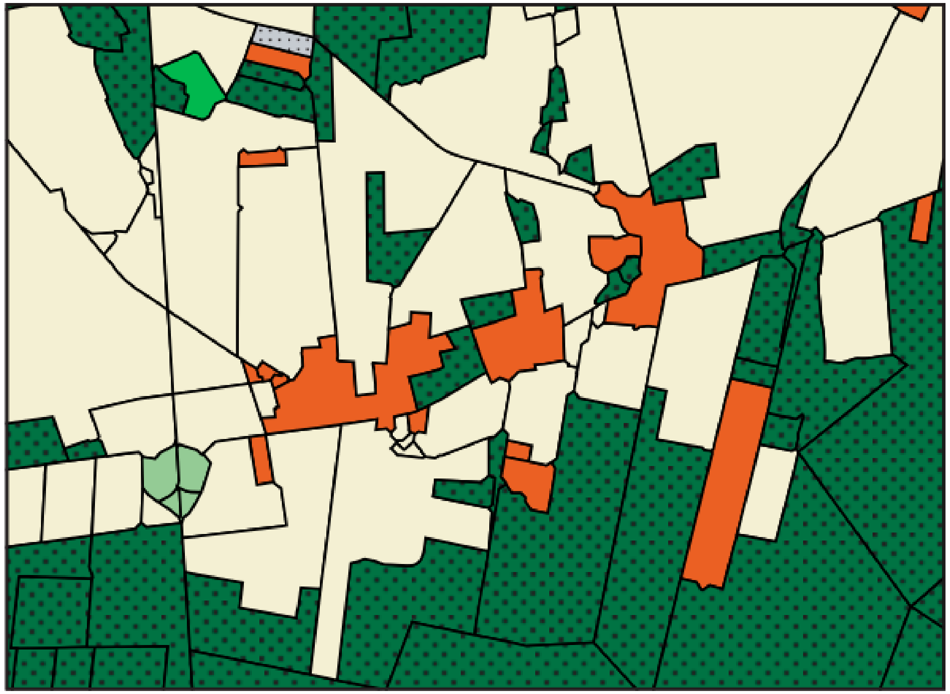
- Grassland
- Forest
- Farmland
- Settlement
- Industry

direktes
Skalieren

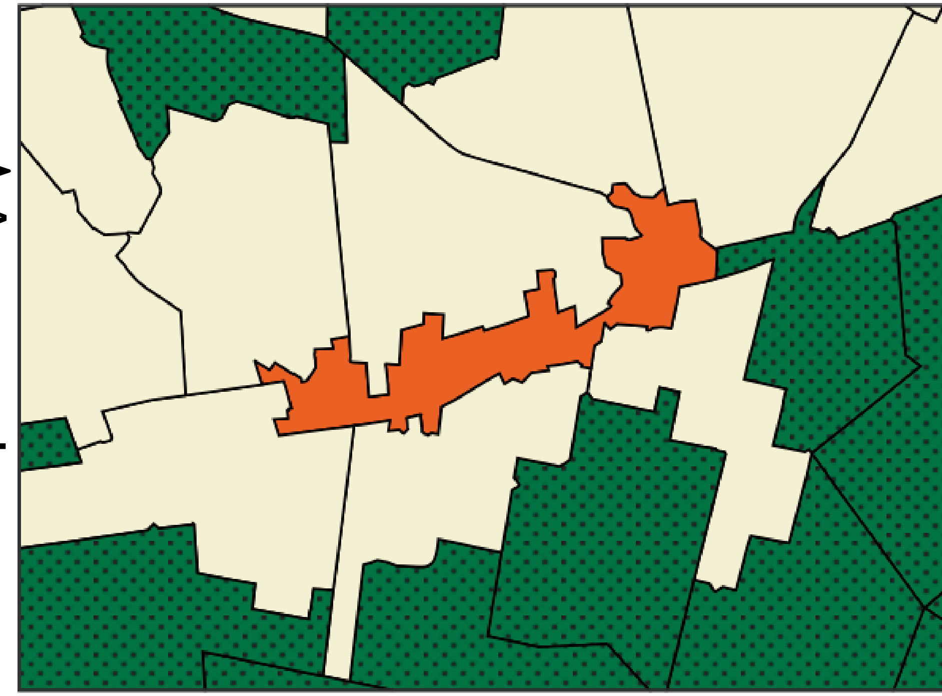
?

Flächennutzung Maßstab 1:250.000

Flächenaggregation



Aggregation



Flächennutzung Maßstab 1:50.000

Grassland

Forest

Farmland

Settlement

Industry

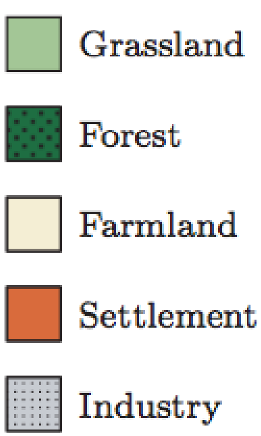
direktes Skalieren

?

Flächennutzung Maßstab 1:250.000

Flächenaggregation

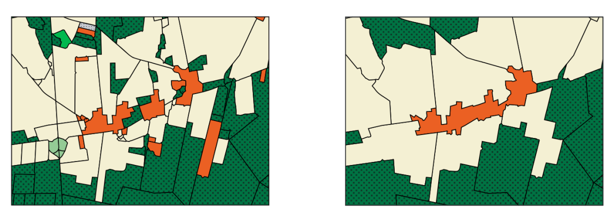
Flächennutzung Maßstab 1:50.000



direktes Skalieren

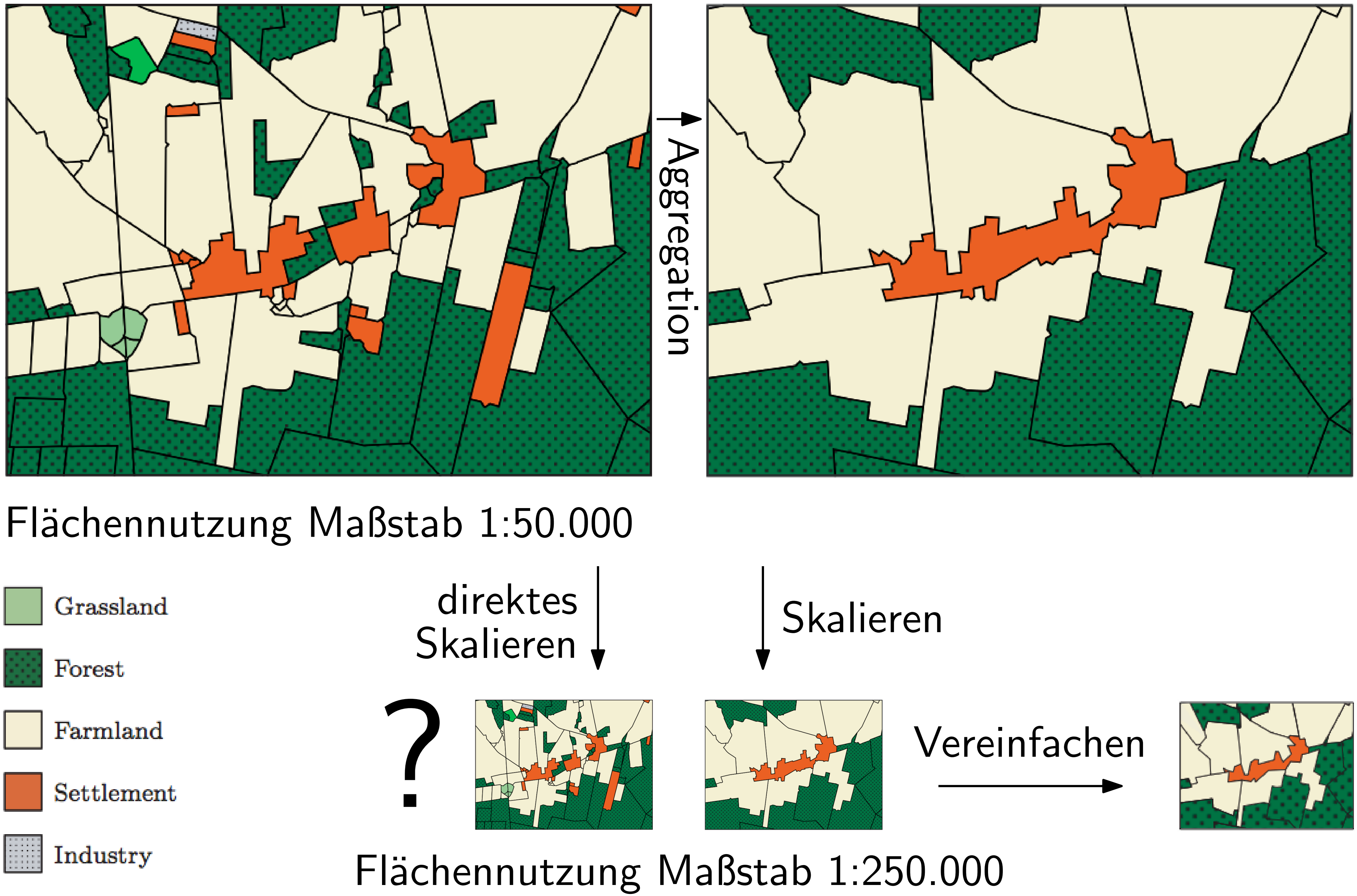
Skalieren

?

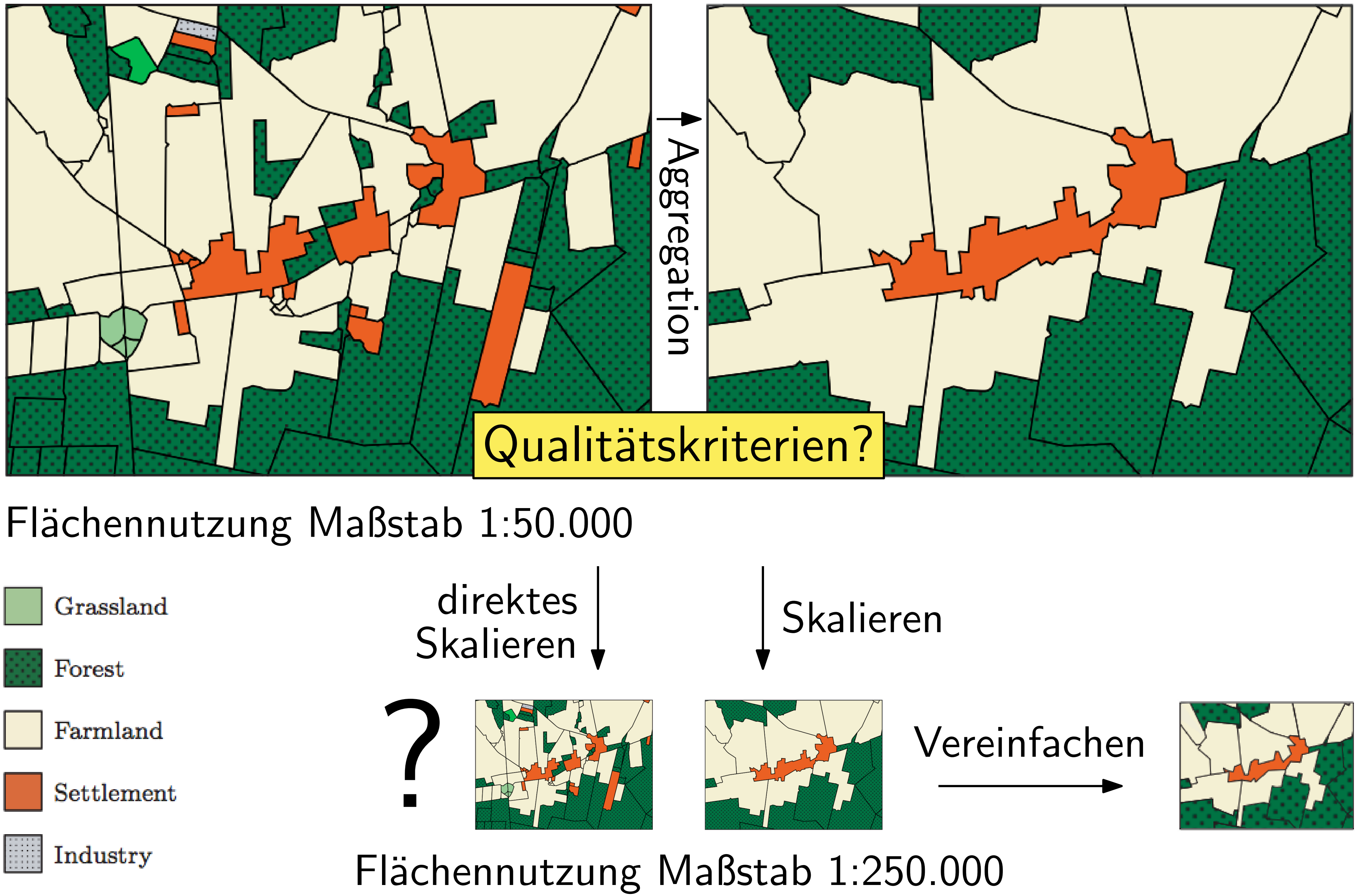


Flächennutzung Maßstab 1:250.000

Flächenaggregation



Flächenaggregation



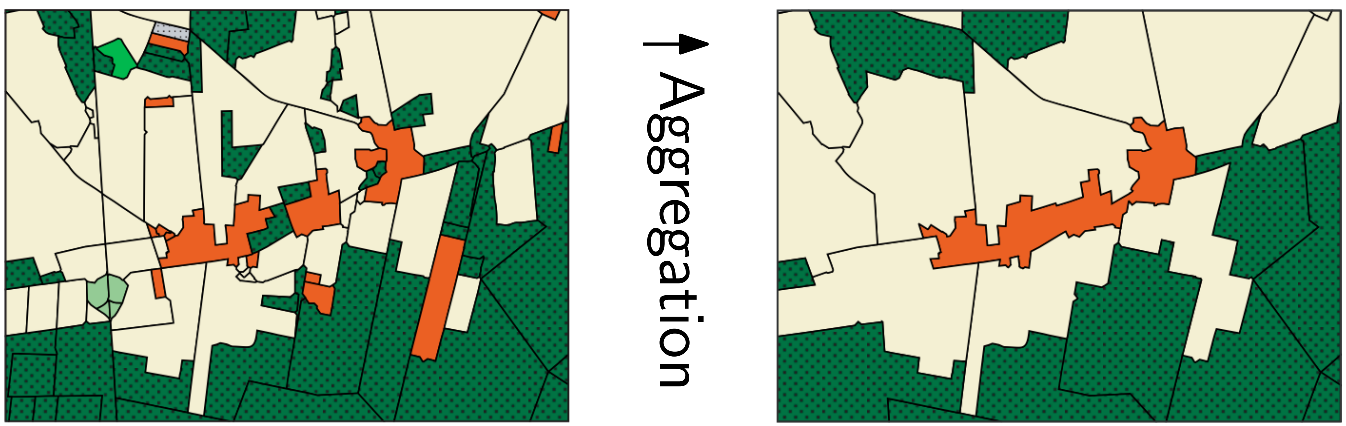
Aufgabenstellung

Daten: topographische Datenbank mit Flächennutzungsdaten

- Unterteilung der Ebene in Polygone bestimmter Nutzungskategorien
- z.B.: Siedlung, Gewerbe, Ackerland, Wald, ...

Ziel: generiere *sinnvolle* Karte in Maßstab 1:Z

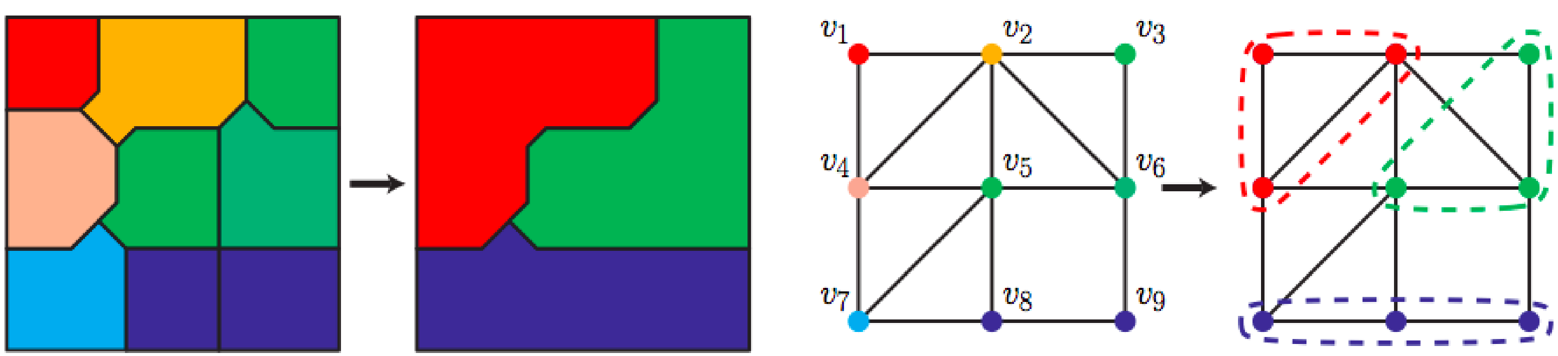
- dargestellte Flächen überschreiten Mindestgröße
- dazu Aggregation benachbarter Flächen nötig
- möglichst kleine Kategorienänderungen
- möglichst kompakte Flächen



Formale Problemstellung AREAAGGREGATION

Geg:
- planarer Graph $G = (V, E)$ mit Knotengewichten $w \colon V \to \mathbb{R}^+$ und Knotenfärbung $\gamma \colon V \to \Gamma$
- Mindestgewicht jeder Farbe $\theta \colon \Gamma \to \mathbb{R}^+$
- semantischer Abstand $d \colon \Gamma \times \Gamma \to \mathbb{R}_0^+$
- (Nicht-)Kompaktheit $c \colon 2^V \times \Gamma \to \mathbb{R}_0^+$
- Parameter $s \in [0, 1]$

Ges: Knotenfärbung $\gamma' \colon V \to \Gamma$ und Partitionierung $P = \{V_1, \ldots, V_p\}$ von V mit
- $\gamma'(v) = \gamma'_i$ für alle $v \in V_i$
- $\sum_{v \in V_i} w(v) \geq \theta(\gamma'_i)$
- induzierter Graph $G[V_i]$ ist zusammenhängend
- $\exists$ Knoten $v \in V_i$, der seine Farbe behält, d.h. $\gamma(v) = \gamma'(v)$



Formale Problemstellung AreaAggregation

Geg:
- planarer Graph $G = (V, E)$ mit Knotengewichten $w \colon V \to \mathbb{R}^+$ und Knotenfärbung $\gamma \colon V \to \Gamma$
- Mindestgewicht jeder Farbe $\theta \colon \Gamma \to \mathbb{R}^+$
- semantischer Abstand $d \colon \Gamma \times \Gamma \to \mathbb{R}_0^+$
- (Nicht-)Kompaktheit $c \colon 2^V \times \Gamma \to \mathbb{R}_0^+$
- Parameter $s \in [0, 1]$

Ges: Knotenfärbung $\gamma' \colon V \to \Gamma$ und Partitionierung $P = \{V_1, \ldots, V_p\}$ von V mit
- $\gamma'(v) = \gamma'_i$ für alle $v \in V_i$
- $\sum_{v \in V_i} w(v) \geq \theta(\gamma'_i)$
- induzierter Graph $G[V_i]$ ist zusammenhängend
- $\exists$ Knoten $v \in V_i$, der seine Farbe behält, d.h. $\gamma(v) = \gamma'(v)$

minimiere dabei
- Kosten $f = s \cdot f_{\text{col}} + (1 - s) \cdot f_{\text{comp}}$
- $f_{\text{col}} = \sum_{v \in V} w(v) \cdot d(\gamma(v), \gamma'(v))$
- $f_{\text{comp}} = \sum_{V_i \in P} c(V_i, \gamma'_i)$

Typische Heuristik

Algorithmus RegionGrowing

$S \leftarrow$ Menge der Regionen kleiner als Mindestfläche
while $S \neq \emptyset$ **do**
  $a \leftarrow$ kleinste Region in S
  verschmelze a mit passendstem Nachbarn
  update S

Typische Heuristik

Algorithmus RegionGrowing

$S \leftarrow$ Menge der Regionen kleiner als Mindestfläche
while $S \neq \emptyset$ **do**
  $a \leftarrow$ kleinste Region in S
  verschmelze a mit passendstem Nachbarn
  update S

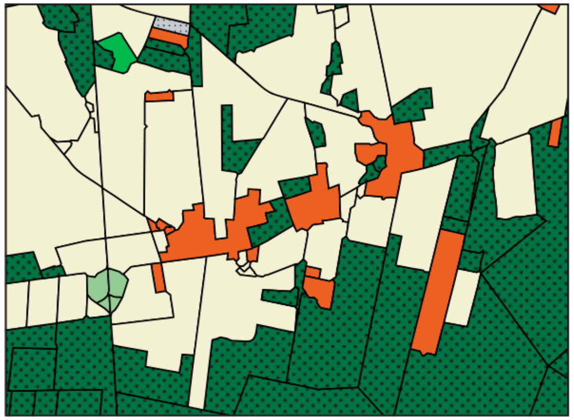
Vor- und Nachteile?

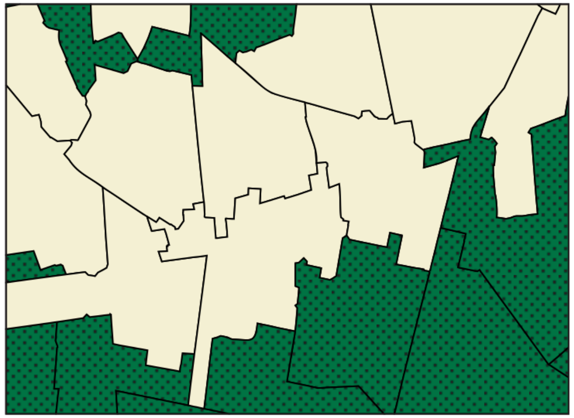
Typische Heuristik

Algorithmus RegionGrowing

$S \leftarrow$ Menge der Regionen kleiner als Mindestfläche
while $S \neq \emptyset$ **do**
- $a \leftarrow$ kleinste Region in S
- verschmelze a mit passendstem Nachbarn
- update S

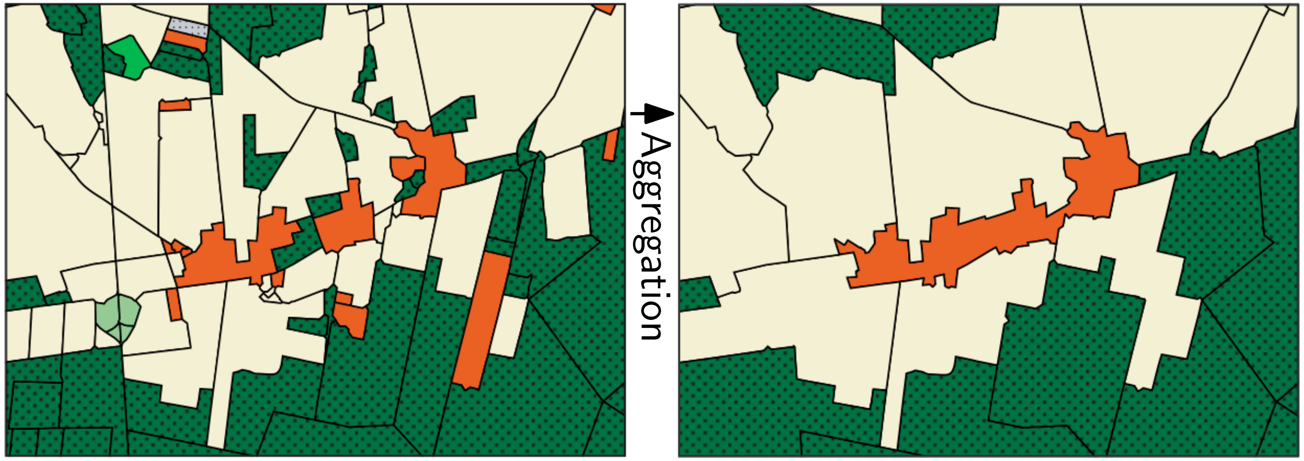
Vor- und Nachteile?





- einfach zu implementieren
- Mindestflächen werden eingehalten
- nur lokale greedy Entscheidungen
- wichtige Features können verschwinden
- keine Qualitätsgarantie für Änderung der Kategorien

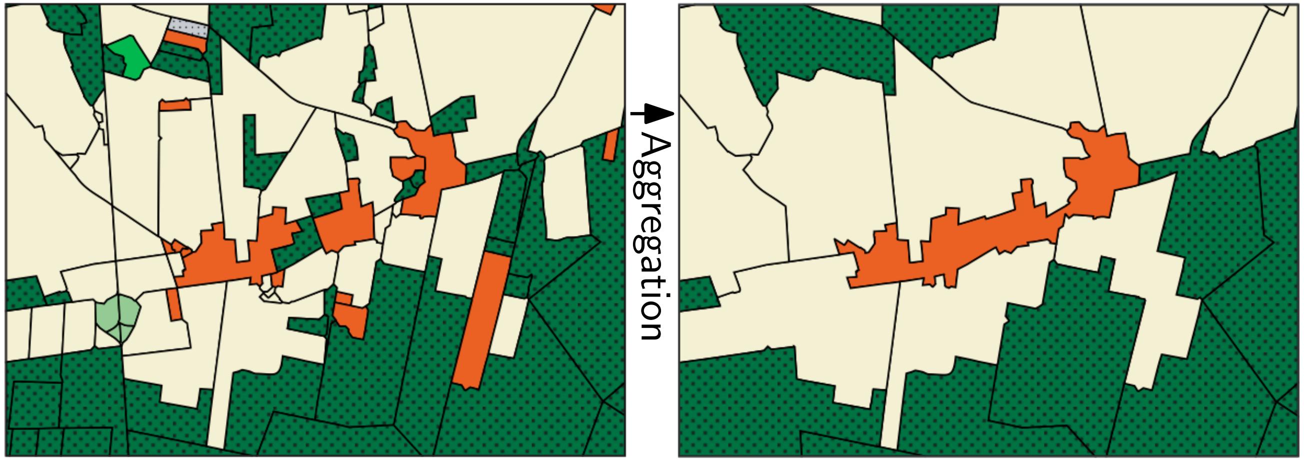
Darüber hinaus:



Geg: Flächennutzungskarte M in Maßstab A
Ges: aggregierte Karte M' für Maßstab $B < A$

- NP-Schwere (planares Vertex Cover)
- minimiere Typänderungen und maximiere Kompaktheit
- drei Modelle via gemischt ganzzahlige Programmierung

Darüber hinaus:



Geg: Flächennutzungskarte M in Maßstab A
Ges: aggregierte Karte M' für Maßstab $B < A$

- NP-Schwere (planares Vertex Cover)
- minimiere Typänderungen und maximiere Kompaktheit
- drei Modelle via gemischt ganzzahlige Programmierung

Wie könnte man noch vorgehen?

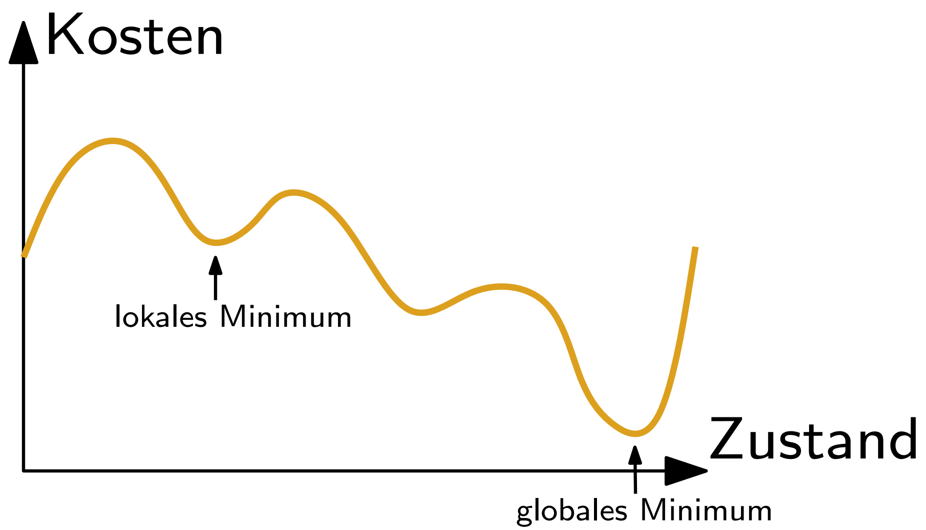
Simulated Annealing

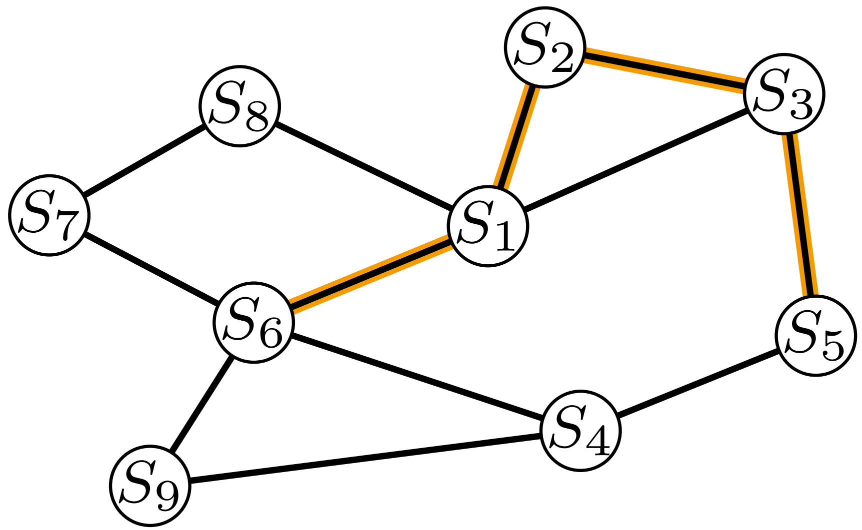
Eingabe: Startzustand S, Starttemperatur T und Zieltemperatur T_0.

solange T größer als T_0 **tue**

- $S' \leftarrow$ Wähle zufälligen benachbarten Zustand von S.
- $\Delta \leftarrow cost(S') - cost(S)$
- **wenn** $\Delta \leq 0$ **dann** — Abstieg
  - $S \leftarrow S'$
- **wenn** $\Delta > 0$ **dann** — Aufstieg
  - $S \leftarrow S'$ mit Wahrsch. $e^{-\frac{\Delta}{T}}$
- $T \leftarrow r \cdot T$ mit $0 < r < 1$ — Abkühlen

return S





Simulated Annealing

Wie auf Flächenaggregation anwenden?

Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

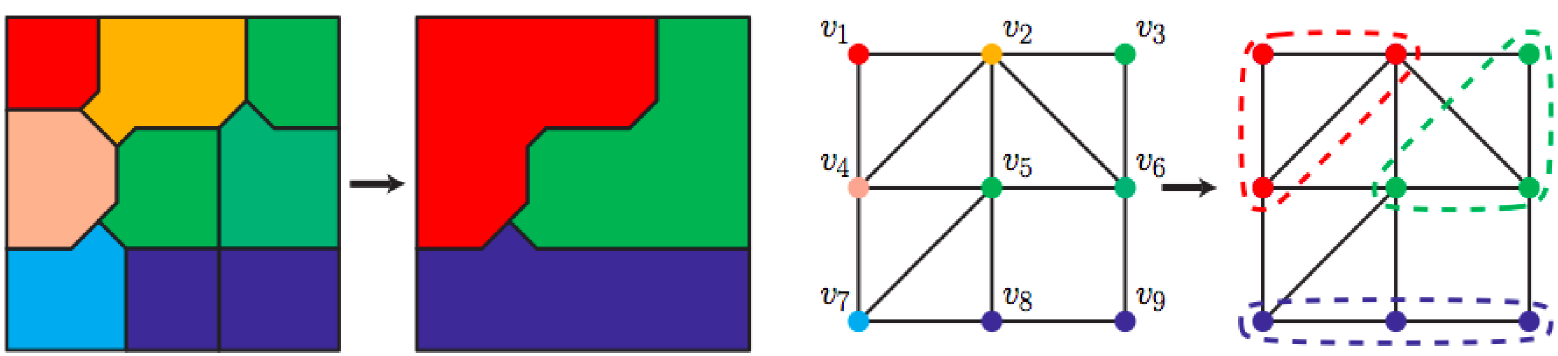
Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Zu 0.: **Knotenfärbung** $\gamma' : V \to \Gamma$ und **Partitionierung** $P = \{V_1, \ldots, V_p\}$ von V mit

- $\gamma'(v) = \gamma'_i$ für alle $v \in V_i$
- $\sum_{v \in V_i} w(v) \geq \theta(\gamma'_i)$
- induzierter Graph $G[V_i]$ ist zusammenhängend
- $\exists$ Knoten $v \in V_i$, der seine Farbe behält, d.h. $\gamma(v) = \gamma'(v)$



Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Zu 1.: Wende Algorithmus `RegionGrowing` an, um Startzustand S zu erhalten.

Algorithmus RegionGrowing

$S \leftarrow$ Menge der Regionen kleiner als Mindestfläche
while $S \neq \emptyset$ **do**
  $a \leftarrow$ kleinste Region in S
  verschmelze a mit passendstem Nachbarn
  update S

Simulated Annealing

Wie auf Flächenaggregation anwenden?

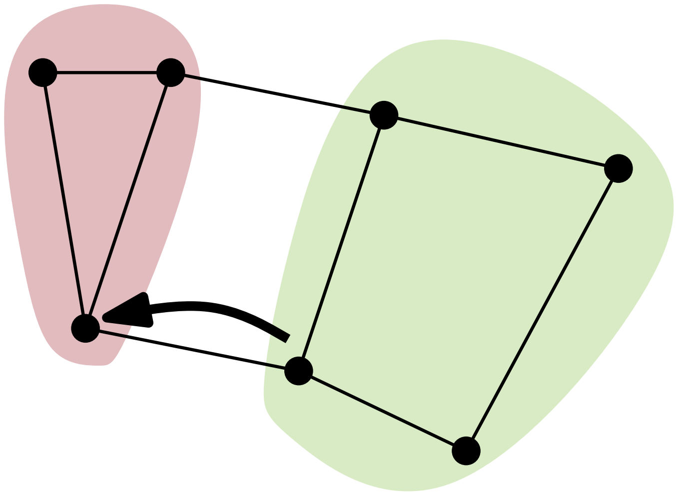
0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Zu 2.: Nachbarschaft eines Zustands $=$ Zustände die durch einen *Knotenaustausch entstehen*

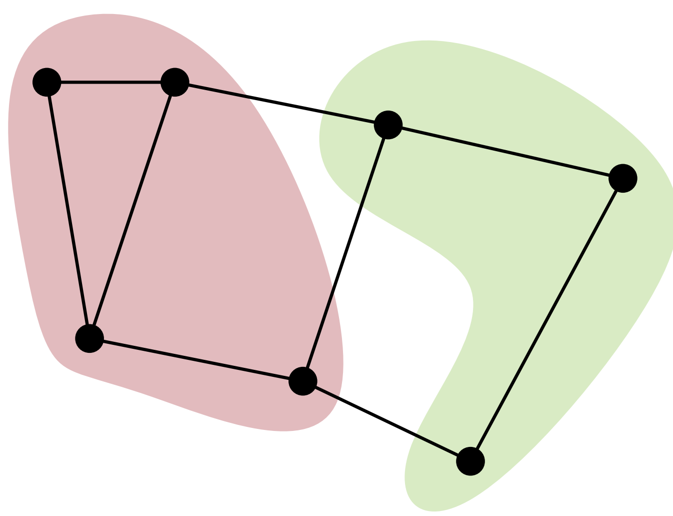


Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Zu 2.: Nachbarschaft eines Zustands $=$ Zustände die durch einen *Knotenaustausch entstehen*



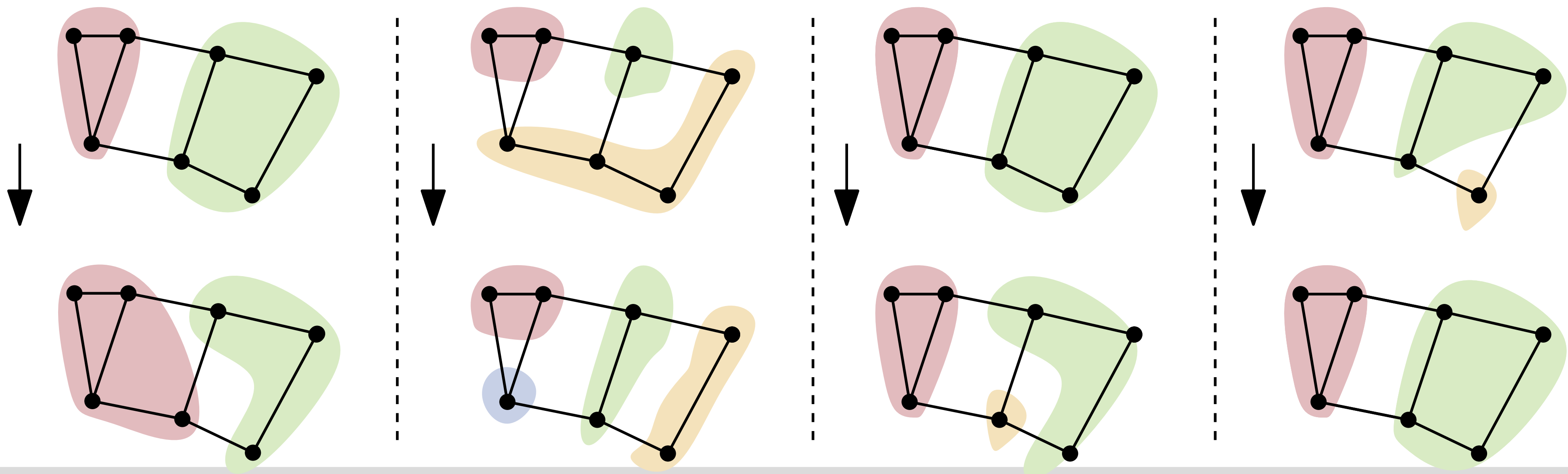
Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Zu 2.: Nachbarschaft eines Zustands = Zustände die durch einen *Knotenaustausch entstehen*

Welche Zustandsübergänge genau zulassen?



Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Zu 2.: Nachbarschaft eines Zustands $=$ Zustände die durch einen *Knotenaustausch entstehen*

Was wenn für jeden Nachbarzustand gilt:

$$\text{Es gibt Farbe } \gamma_i', \text{ sodass } \sum_{v \in V_i} w(v) < \theta(\gamma_i')$$

Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Zu 2.: Nachbarschaft eines Zustands $=$ Zustände die durch einen *Knotenaustausch entstehen*

Was wenn für jeden Nachbarzustand gilt:

$$\text{Es gibt Farbe } \gamma_i', \text{ sodass } \sum_{v \in V_i} w(v) < \theta(\gamma_i')$$

Lsg.:

1. Hebe Bedingung auf.
2. Solche eigentlich unzulässigen Zustände erhalten Extrakosten.
3. Nach Verfahren wende `RegionGrowing` an, um Bed. zu erfüllen.

Simulated Annealing

Wie auf Flächenaggregation anwenden?

0. Was ist ein Zustand?
1. Wie Startzustand festlegen?
2. Wie Nachbarschaft von Zuständen definieren?
3. Wie T, T_0 und r festlegen?

Zu 3.: Parameter werden üblicherweise experimentell festgelegt.