

Vorlesung Algorithmische Kartografie

Flächenaggregation

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

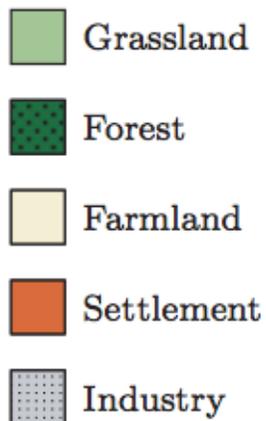
Martin Nöllenburg
09.07.2013



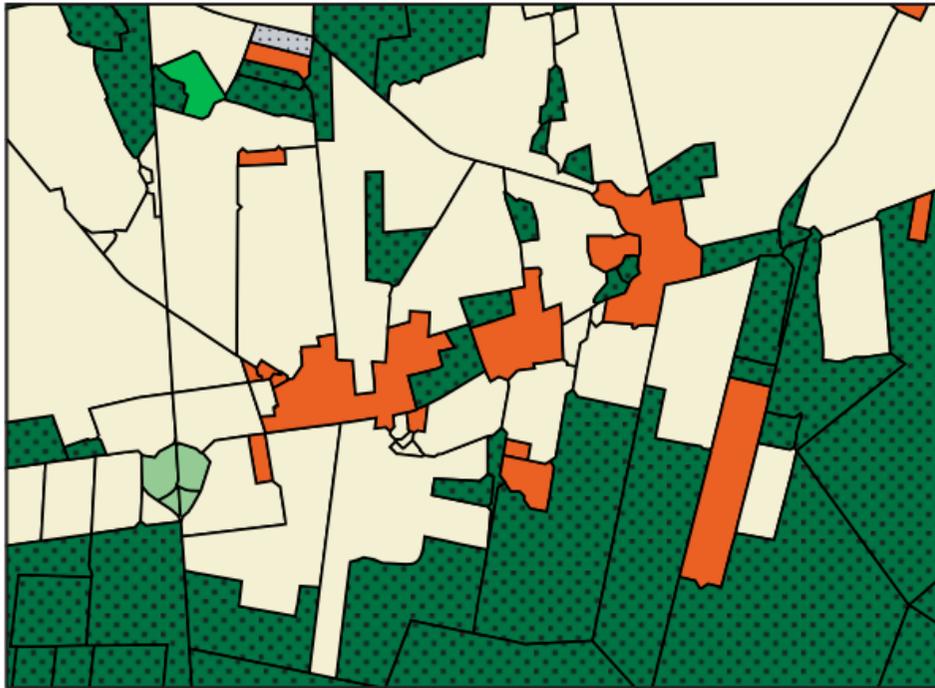
Flächenaggregation



Flächennutzung Maßstab 1:50.000



Flächenaggregation



Flächennutzung Maßstab 1:50.000

 Grassland

 Forest

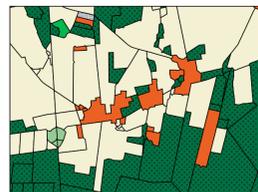
 Farmland

 Settlement

 Industry

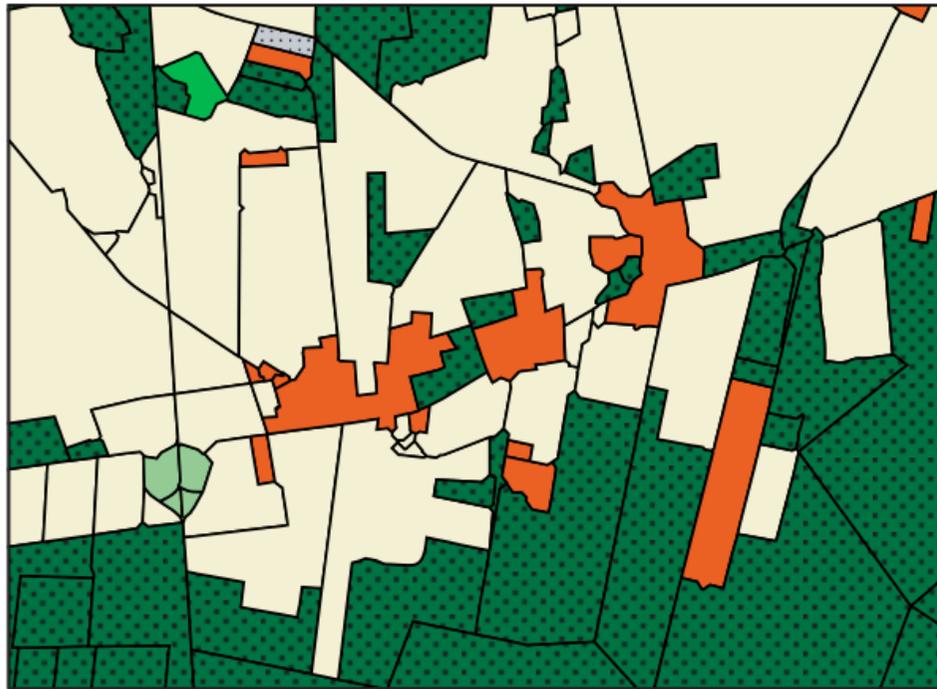
direktes
Skalieren ↓

?

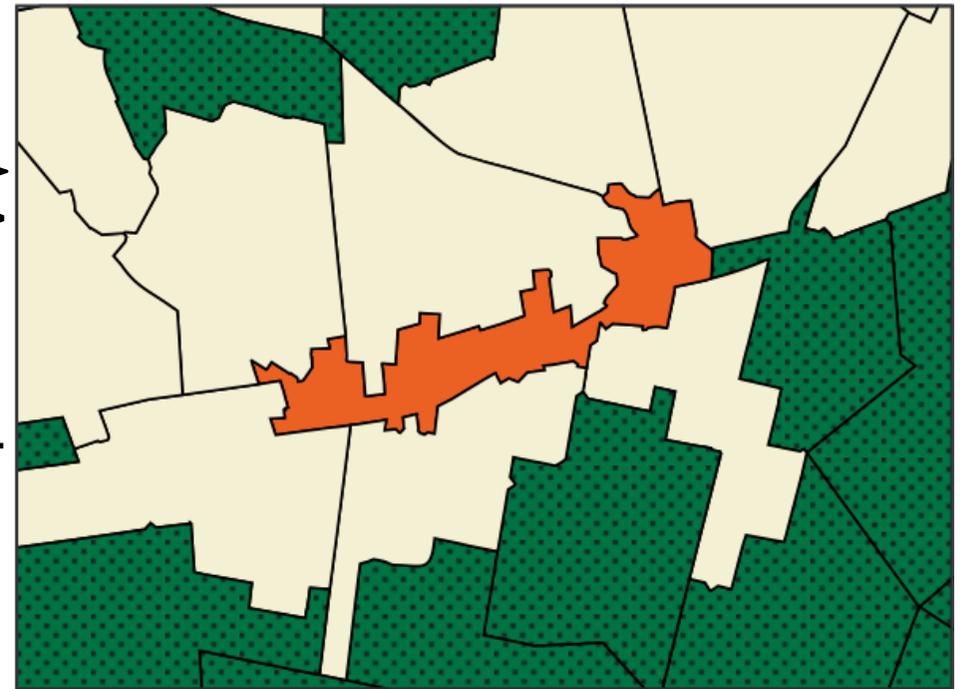


Flächennutzung Maßstab 1:250.000

Flächenaggregation



↑ Aggregation



Flächennutzung Maßstab 1:50.000

 Grassland

 Forest

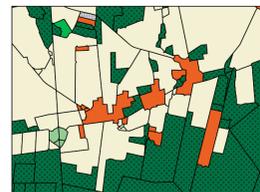
 Farmland

 Settlement

 Industry

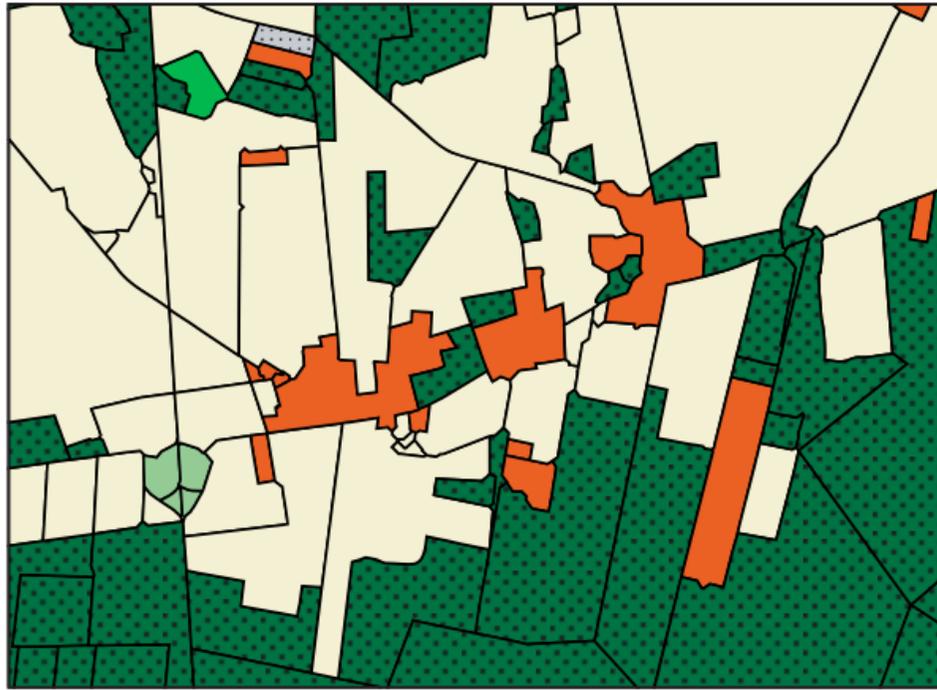
direktes
Skalieren ↓

?

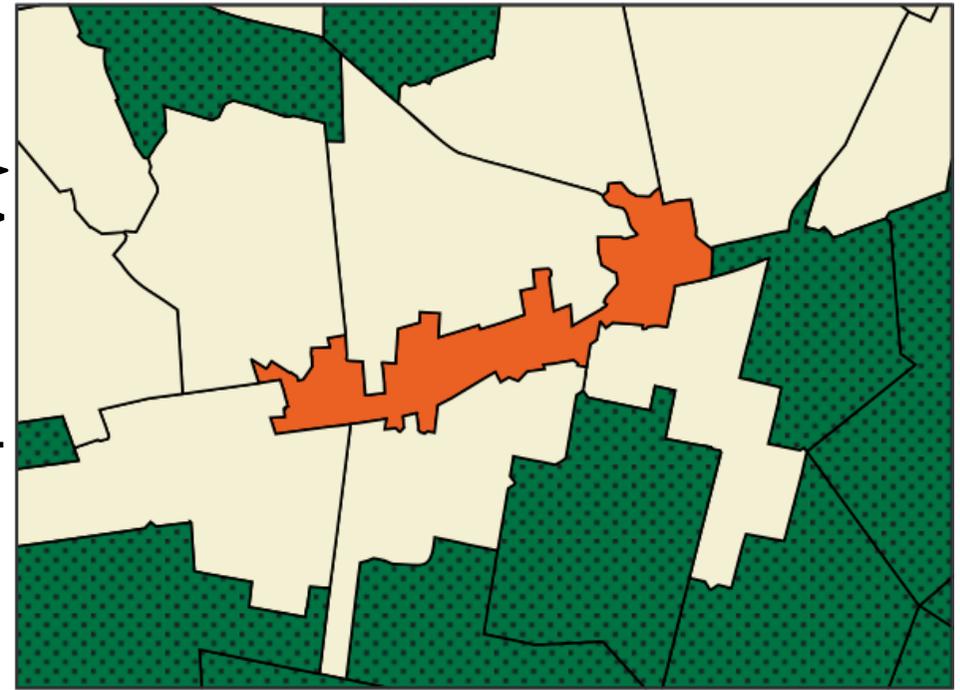


Flächennutzung Maßstab 1:250.000

Flächenaggregation



↑ Aggregation



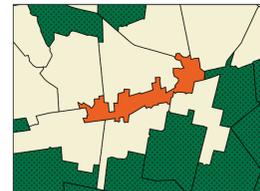
Flächennutzung Maßstab 1:50.000

-  Grassland
-  Forest
-  Farmland
-  Settlement
-  Industry

direktes
Skalieren ↓

Skalieren ↓

?

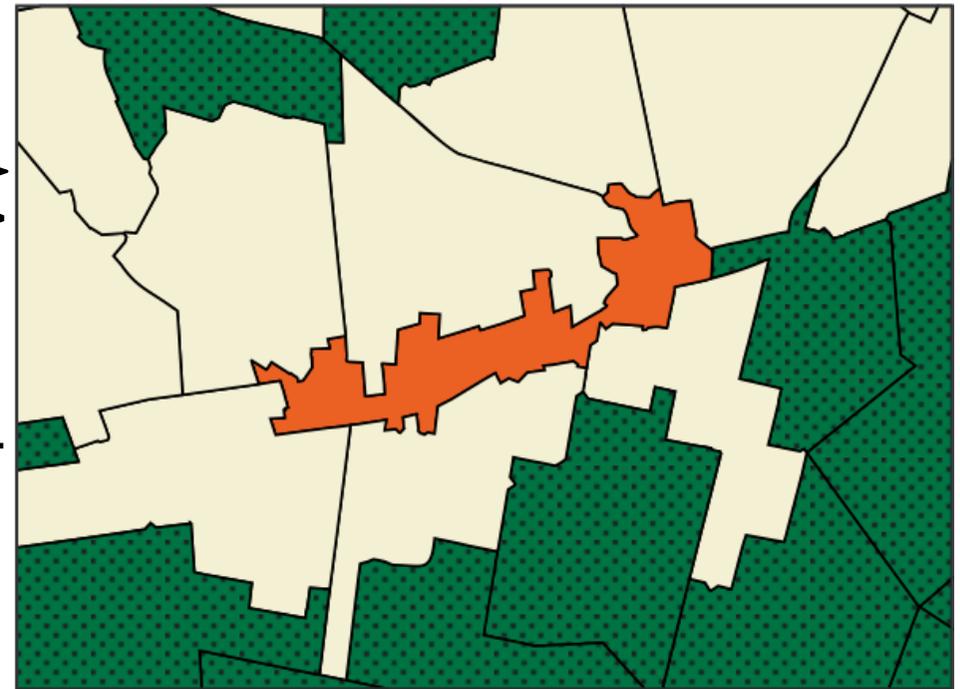


Flächennutzung Maßstab 1:250.000

Flächenaggregation



↓ Aggregation



Flächennutzung Maßstab 1:50.000

 Grassland

 Forest

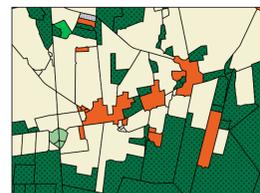
 Farmland

 Settlement

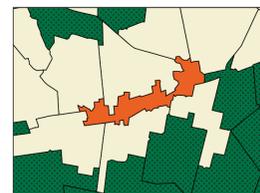
 Industry

↓ direktes
Skalieren

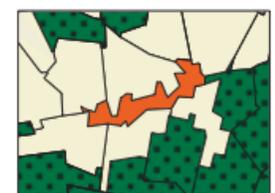
?



↓ Skalieren

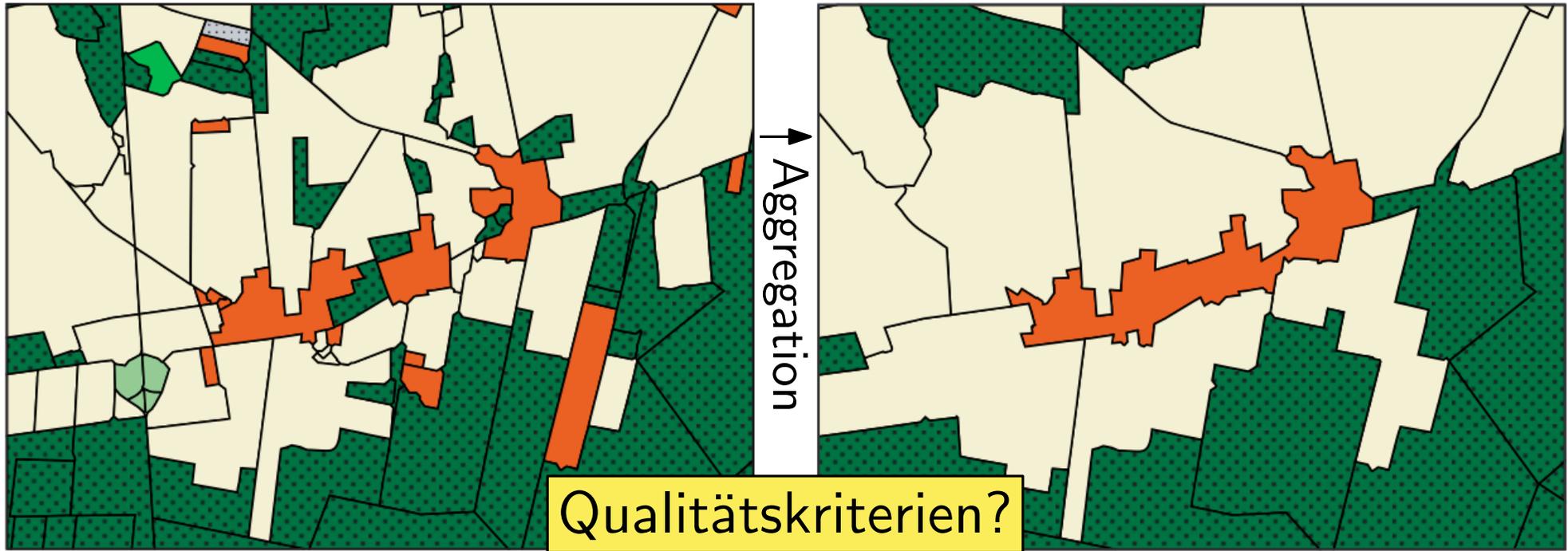


→ Vereinfachen



Flächennutzung Maßstab 1:250.000

Flächenaggregation



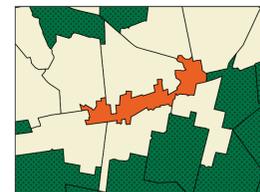
Flächennutzung Maßstab 1:50.000

-  Grassland
-  Forest
-  Farmland
-  Settlement
-  Industry

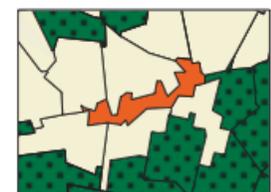
direktes
Skalieren

Skalieren

?



Vereinfachen



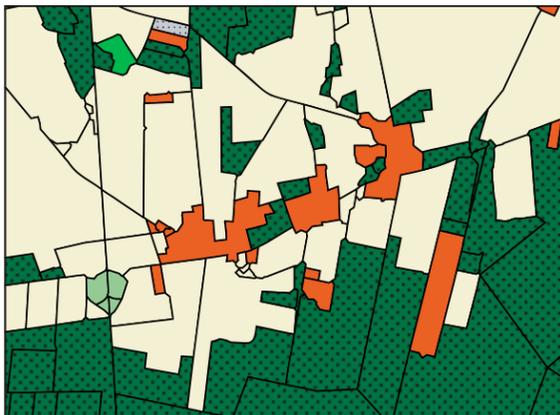
Flächennutzung Maßstab 1:250.000

Daten: topographische Datenbank mit Flächennutzungsdaten

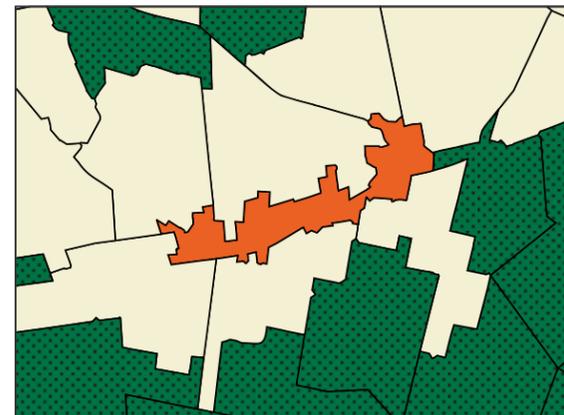
- Unterteilung der Ebene in Polygone bestimmter Nutzungskategorien
- z.B.: Siedlung, Gewerbe, Ackerland, Wald, ...

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

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



↑ Aggregation



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

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?

Algorithmus RegionGrowing

$S \leftarrow$ Menge der Regionen kleiner als Mindestfläche

while $S \neq \emptyset$ **do**

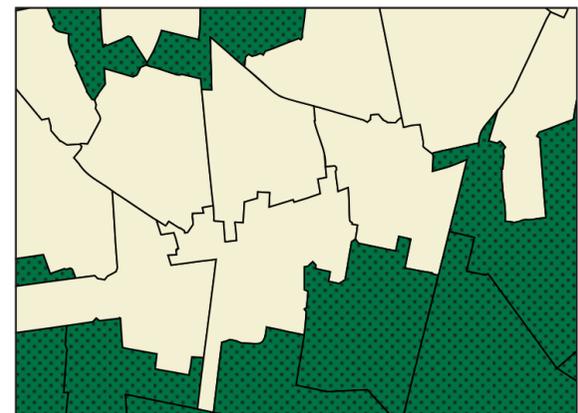
$a \leftarrow$ kleinste Region in S

 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



1) Logische Konsistenz

- Ausgabe ist Unterteilung der Ebene
- Einhaltung von Mindestflächen, ggf. kategorienabhängig (z.B. Wald \geq 40 ha für 1:250K)

1) Logische Konsistenz

- Ausgabe ist Unterteilung der Ebene
- Einhaltung von Mindestflächen, ggf. kategorienabhängig (z.B. Wald \geq 40 ha für 1:250K)

2) Vollständigkeit

- jede Eingaberegion, die groß genug ist muss korrekt in ihrer Kategorie dargestellt sein

1) Logische Konsistenz

- Ausgabe ist Unterteilung der Ebene
- Einhaltung von Mindestflächen, ggf. kategorienabhängig (z.B. Wald ≥ 40 ha für 1:250K)

2) Vollständigkeit

- jede Eingaberegion, die groß genug ist muss korrekt in ihrer Kategorie dargestellt sein

3) Semantische Genauigkeit

- semantische Distanz $d: \Gamma \times \Gamma \rightarrow \mathbb{R}_0^+$ zwischen Kategorien
- minimiere $\bar{d} = \sum_{v \in V} w(v) \cdot d(\gamma(v), \gamma'(v)) / \sum_{v \in V} w(v)$

1) Logische Konsistenz

- Ausgabe ist Unterteilung der Ebene
- Einhaltung von Mindestflächen, ggf. kategorienabhängig (z.B. Wald ≥ 40 ha für 1:250K)

2) Vollständigkeit

- jede Eingaberegion, die groß genug ist muss korrekt in ihrer Kategorie dargestellt sein

3) Semantische Genauigkeit

- semantische Distanz $d: \Gamma \times \Gamma \rightarrow \mathbb{R}_0^+$ zwischen Kategorien
- minimiere $\bar{d} = \sum_{v \in V} w(v) \cdot d(\gamma(v), \gamma'(v)) / \sum_{v \in V} w(v)$

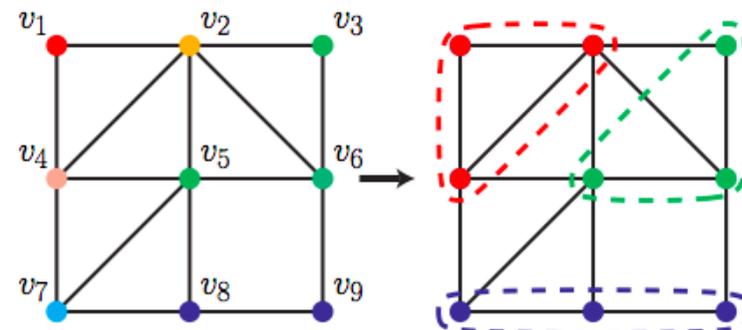
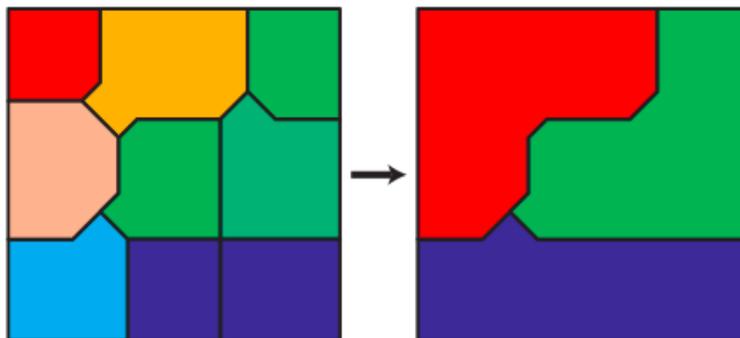
4) Kompaktheit

- aggregierte Regionen sollen möglichst kompakt sein, gemessen an einem Kompaktheitsmaß $c: 2^V \times \Gamma \rightarrow \mathbb{R}_0^+$

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

Ges: Knotenfärbung $\gamma': V \rightarrow \Gamma$ und Partitionierung $P = \{V_1, \dots, V_p\}$ 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. $\gamma(v) = \gamma'(v)$



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

Ges: Knotenfärbung $\gamma': V \rightarrow \Gamma$ und Partitionierung $P = \{V_1, \dots, V_p\}$ 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 $f = s \cdot f_{\text{col}} + (1 - s) \cdot f_{\text{comp}}$
- $f_{\text{col}} = \sum_{v \in V} w(v) \cdot d(\gamma(v), \gamma'(v))$
- $f_{\text{comp}} = \sum_{V_i \in P} c(V_i, \gamma'_i)$

Umfangsbasiert

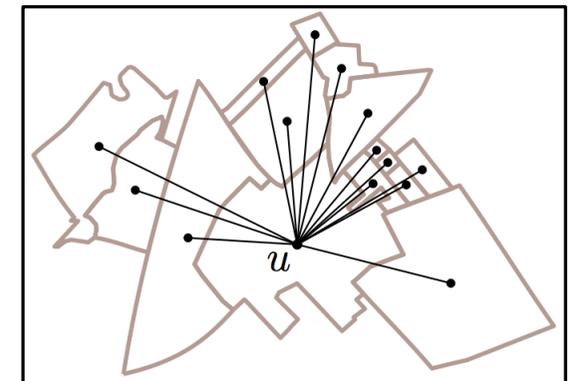
- je kleiner der Umfang einer Region desto kompakter
- $c_p(V') = \text{Umfang}(V')$
- präferiert in der Summe über P größere Regionen
- Alternativen wie $c_p / (2\sqrt{\pi}\text{Fläche})$ oder $\sqrt{\text{Fläche}} \cdot c_p / 2\sqrt{\pi}$ sind problematisch für lineare Optimierung

Umfangsbasiert

- je kleiner der Umfang einer Region desto kompakter
- $c_p(V') = \text{Umfang}(V')$
- präferiert in der Summe über P größere Regionen
- Alternativen wie $c_p/(2\sqrt{\pi\text{Fläche}})$ oder $\sqrt{\text{Fläche}} \cdot c_p/2\sqrt{\pi}$ sind problematisch für lineare Optimierung

Distanz zum Zentrum

- sei $\delta: V \times V \rightarrow \mathbb{R}_0^+$ Euklidischer Abstand zwischen Schwerpunkten
- $c_d(V', k) = \min\{\sum_{v \in V'} w(v) \cdot \delta(v, u) \mid u \in V' \text{ und } \gamma(u) = k\}$
- Flächen sammeln sich rund um Zentrum mit unverändertem Typ
- präferiert kleinere Regionen, jedoch beschränkt durch θ
- Kombination mit c_p möglich

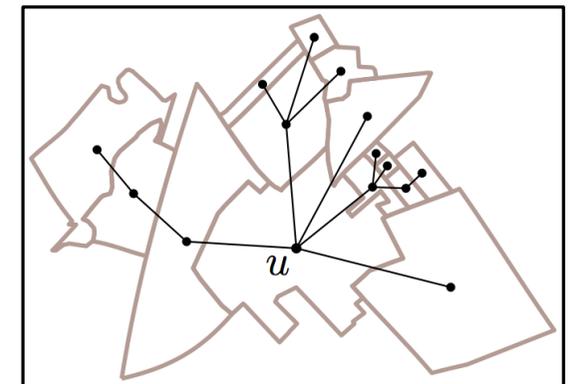


Umfangsbasiert

- je kleiner der Umfang einer Region desto kompakter
- $c_p(V') = \text{Umfang}(V')$
- präferiert in der Summe über P größere Regionen
- Alternativen wie $c_p/(2\sqrt{\pi\text{Fläche}})$ oder $\sqrt{\text{Fläche}} \cdot c_p/2\sqrt{\pi}$ sind problematisch für lineare Optimierung

Distanz zum Zentrum

- sei $\delta: V \times V \rightarrow \mathbb{R}_0^+$ Euklidischer Abstand zwischen Schwerpunkten
- $c_d(V', k) = \min\{\sum_{v \in V'} w(v) \cdot \delta(v, u) \mid u \in V' \text{ und } \gamma(u) = k\}$
- Flächen sammeln sich rund um Zentrum mit unverändertem Typ
- präferiert kleinere Regionen, jedoch beschränkt durch θ
- Kombination mit c_p möglich
- nutze alternativ kürzeste Wege $\delta'_{V'}(u, v)$ in G , die innerhalb V' verlaufen $\rightarrow c_{sp}(V', k)$



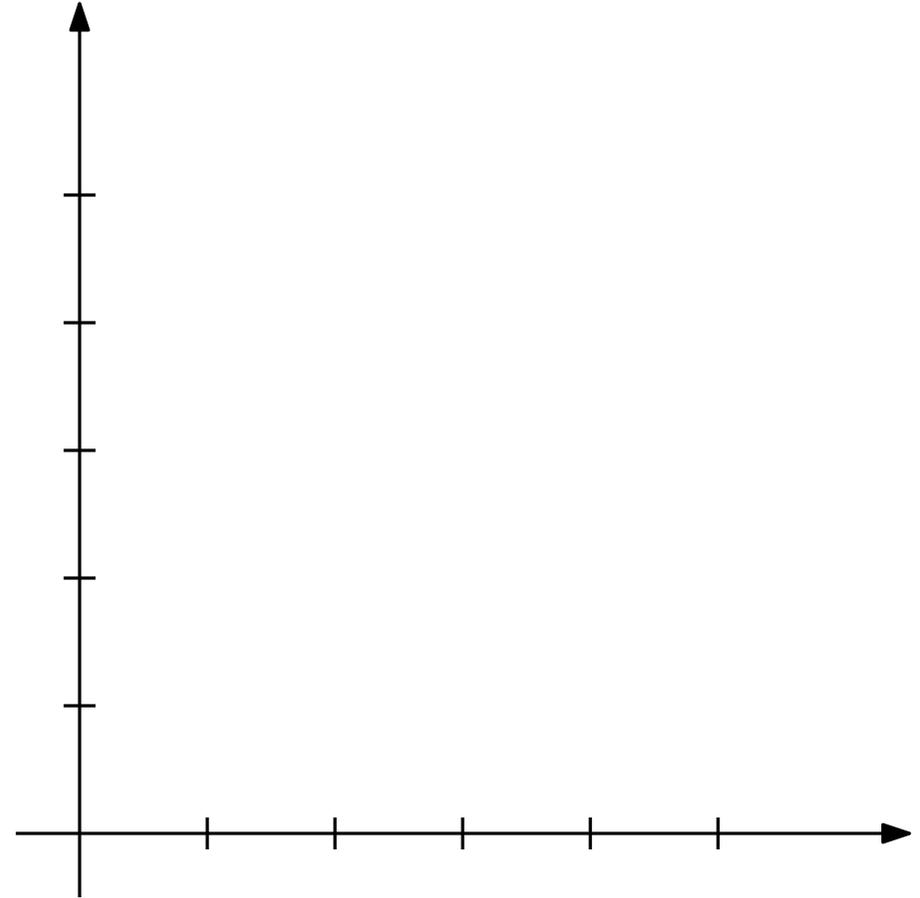
Satz 1: Für eine gegebene Instanz von AREAAGGREGATION und einen Wert $C \in \mathbb{Z}$ ist es NP-schwer zu entscheiden, ob eine Lösung mit Kosten $\leq C$ existiert, selbst wenn $|\Gamma| = 2$, $w \equiv 1$, θ für alle Farben gleich, Distanz $d \in \{0, 1\}$ und $s = 1$.

Beweis: Reduktion von planarem Vertex Cover (s. Tafel)

(Gemischt) Ganzzahlige Programmierung

Lineare Programmierung (LP) ist ein effizientes Optimierungsverfahren für

- lineare Nebenbedingungen
- lineare Zielfunktion
- reellwertige Variablen



(Gemischt) Ganzzahlige Programmierung

Lineare Programmierung (LP) ist ein effizientes Optimierungsverfahren für

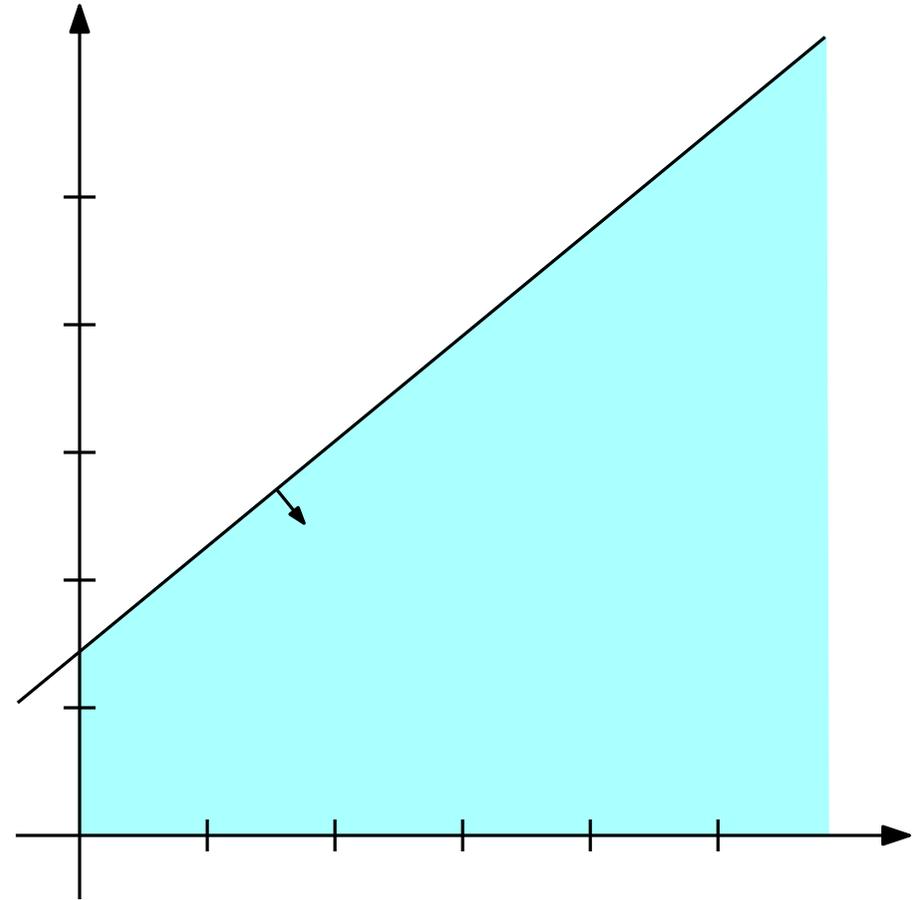
- lineare Nebenbedingungen
- lineare Zielfunktion
- reellwertige Variablen

Beispiel:

maximiere $x + 2y$ sodass

$$y \leq 0,9x + 1,5$$

$$y \geq 1,4x - 1,3$$



(Gemischt) Ganzzahlige Programmierung

Lineare Programmierung (LP) ist ein effizientes Optimierungsverfahren für

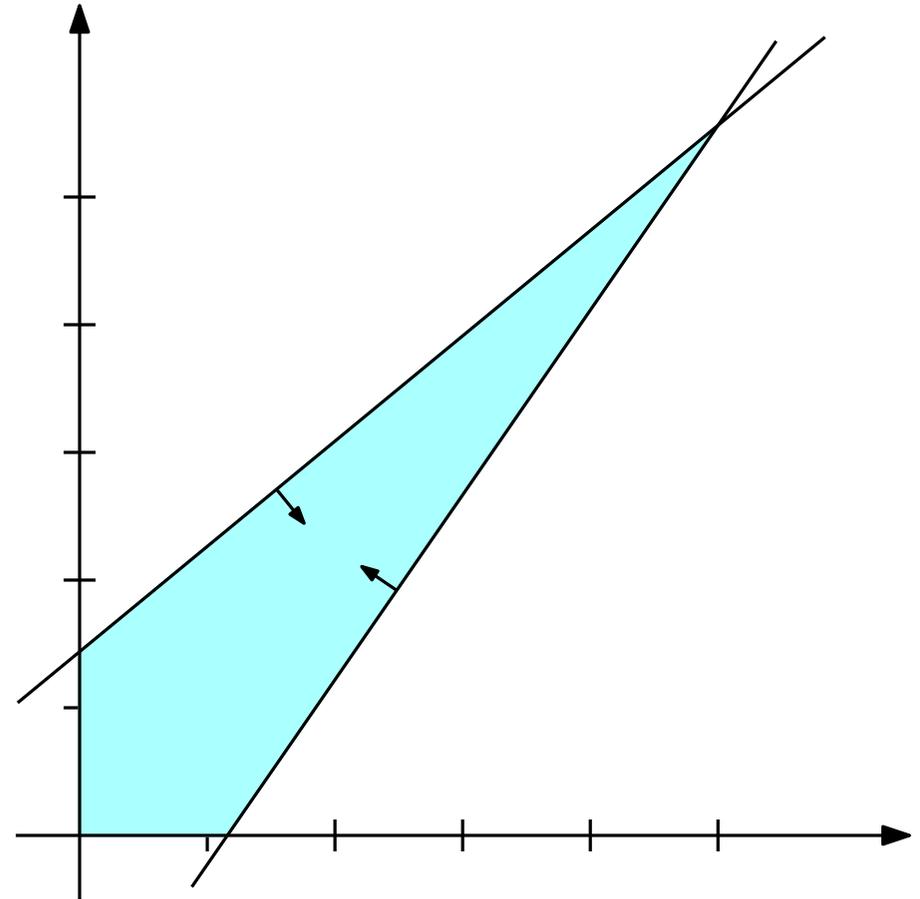
- lineare Nebenbedingungen
- lineare Zielfunktion
- reellwertige Variablen

Beispiel:

maximiere $x + 2y$ sodass

$$y \leq 0,9x + 1,5$$

$$y \geq 1,4x - 1,3$$



(Gemischt) Ganzzahlige Programmierung

Lineare Programmierung (LP) ist ein effizientes Optimierungsverfahren für

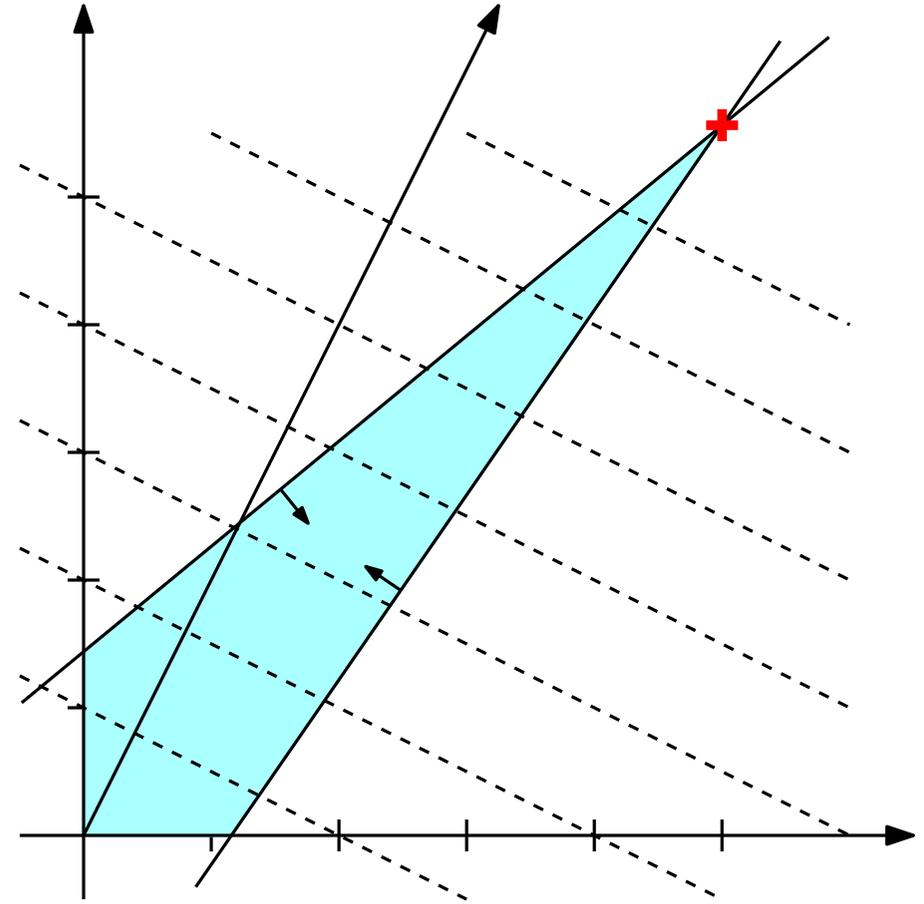
- lineare Nebenbedingungen
- lineare Zielfunktion
- reellwertige Variablen

Beispiel:

maximiere $x + 2y$ sodass

$$y \leq 0,9x + 1,5$$

$$y \geq 1,4x - 1,3$$



(Gemischt) Ganzzahlige Programmierung

Lineare Programmierung (LP) ist ein effizientes Optimierungsverfahren für

- lineare Nebenbedingungen
- lineare Zielfunktion
- reellwertige Variablen

Beispiel:

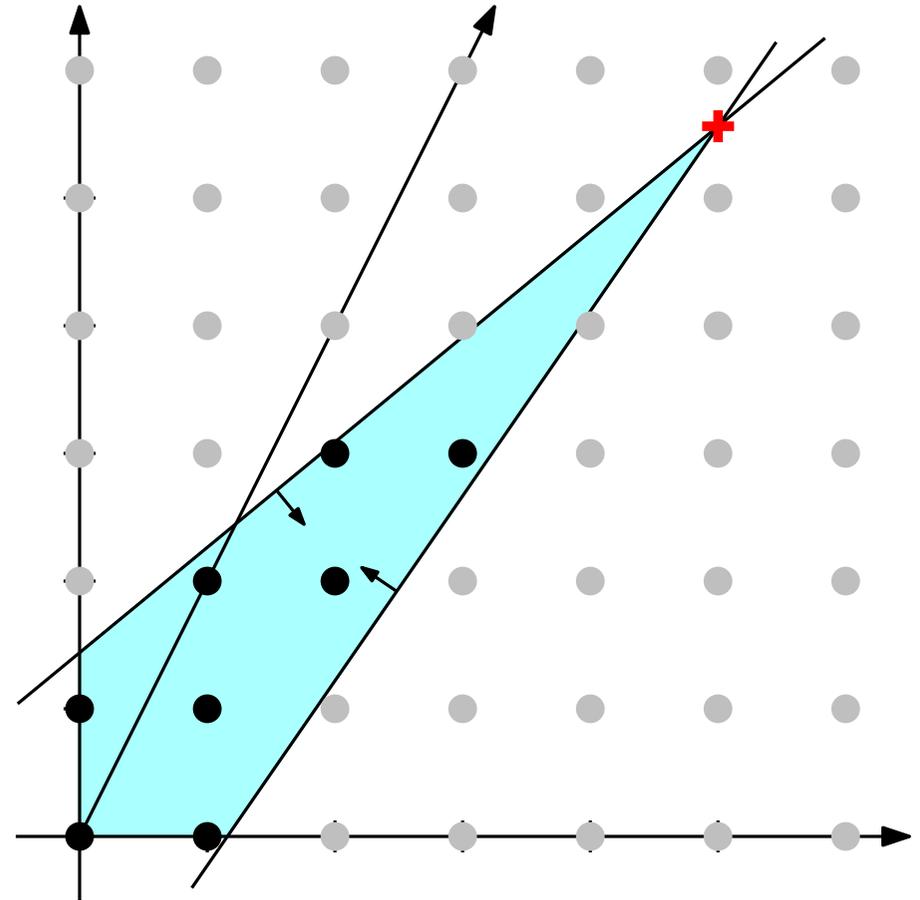
maximiere $x + 2y$ sodass

$$y \leq 0,9x + 1,5$$

$$y \geq 1,4x - 1,3$$

Gemischt ganzzahlige Programmierung (MIP)

- erlaubt binäre und ganzzahlige Variablen
- ist im Allg. NP-schwer
- liefert dennoch oft brauchbare und praktikable Lösungen in der Praxis



(Gemischt) Ganzzahlige Programmierung

Lineare Programmierung (LP) ist ein effizientes Optimierungsverfahren für

- lineare Nebenbedingungen
- lineare Zielfunktion
- reellwertige Variablen

Beispiel:

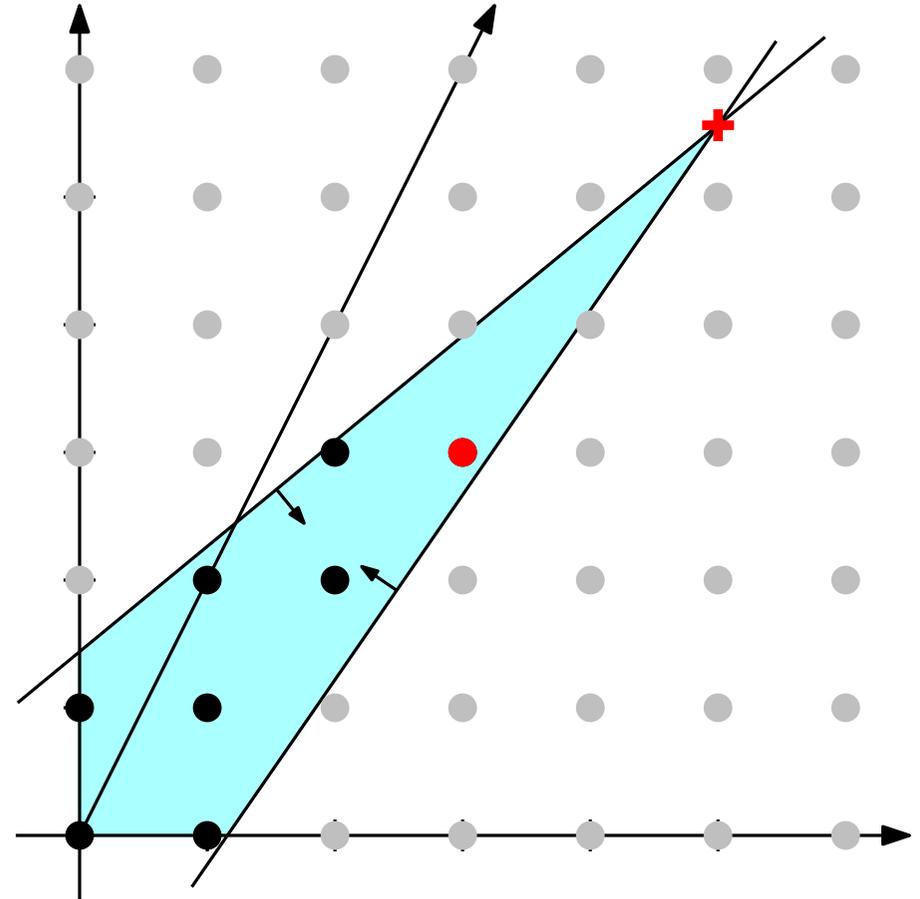
maximiere $x + 2y$ sodass

$$y \leq 0,9x + 1,5$$

$$y \geq 1,4x - 1,3$$

Gemischt ganzzahlige Programmierung (MIP)

- erlaubt binäre und ganzzahlige Variablen
- ist im Allg. NP-schwer
- liefert dennoch oft brauchbare und praktikable Lösungen in der Praxis



Variablen:

- $f_a \in [0, M]$ definiert den Fluss über Kante a
($M = \sum_{v \in V} w(v)$)
- $F_a \in \{0, 1\}$ mit $F_a = 1$ falls $f_a > 0$
- $b_{vk} \in \{0, 1\}$ mit $b_{vk} = 1$ falls $\gamma'(v) = k$
- $s_v \in \{0, 1\}$ mit $s_v = 1$ falls v eine Senke ist

Variablen:

- $f_a \in [0, M]$ definiert den Fluss über Kante a
($M = \sum_{v \in V} w(v)$)
- $F_a \in \{0, 1\}$ mit $F_a = 1$ falls $f_a > 0$
- $b_{vk} \in \{0, 1\}$ mit $b_{vk} = 1$ falls $\gamma'(v) = k$
- $s_v \in \{0, 1\}$ mit $s_v = 1$ falls v eine Senke ist

Zielfunktion:

minimiere

$$s \sum_{v \in V} \sum_{k \in \Gamma} w(v) \cdot d(\gamma(v), k) \cdot b_{vk} + (1 - s) \sum_{a=uv \in A} \delta(u, v) \cdot f_a$$

Nebenbedingungen:

$$M \cdot F_a \geq f_a \quad \text{für alle } a \in A \quad (1)$$

$$\sum_{k \in \Gamma} b_{vk} = 1 \quad \text{für alle } v \in V \quad (2)$$

$$b_{uk} \geq b_{vk} + (F_{uv} - 1) \quad \text{für alle } uv \in A, k \in \Gamma \quad (3)$$

$$b_{v\gamma(v)} \geq s_v \quad \text{für alle } v \in V \quad (4)$$

$$\sum_{a=vu \in A} f_a - \sum_{a=uv \in A} f_a \geq w(v) - Ms_v \quad \text{für alle } v \in V \quad (5)$$

$$\sum_{a=vu \in A} f_a - \sum_{a=uv \in A} f_a \leq w(v) - s_v \theta(\gamma(v)) \quad \text{für alle } v \in V \quad (6)$$

$$\sum_{a=uv \in A} F_a \leq 1 - s_u \quad \text{für alle } u \in V \quad (7)$$

Nebenbedingungen:

$$M \cdot F_a \geq f_a \quad \text{für alle } a \in A \quad (1)$$

$$\sum_{k \in \Gamma} b_{vk} = 1 \quad \text{für alle } v \in V \quad (2)$$

$$b_{uk} \geq b_{vk} + (F_{uv} - 1) \quad \text{für alle } uv \in A, k \in \Gamma \quad (3)$$

$$b_{v\gamma(v)} \geq s_v \quad \text{für alle } v \in V \quad (4)$$

$$\sum_{a=vu \in A} f_a - \sum_{a=uv \in A} f_a \geq w(v) - Ms_v \quad \text{für alle } v \in V \quad (5)$$

$$\sum_{a=vu \in A} f_a - \sum_{a=uv \in A} f_a \leq w(v) - s_v \theta(\gamma(v)) \quad \text{für alle } v \in V \quad (6)$$

$$\sum_{a=uv \in A} F_a \leq 1 - s_u \quad \text{für alle } u \in V \quad (7)$$

Wie lässt sich Vollständigkeit modellieren?

Variablen:

- $x_{uv} \in \{0, 1\}$ mit $x_{uv} = 1$ gdw. Knoten v zum Zentrum u gehört

Variablen:

- $x_{uv} \in \{0, 1\}$ mit $x_{uv} = 1$ gdw. Knoten v zum Zentrum u gehört

Zielfunktion:

minimiere

$$s \sum_{(u,v) \in V^2} w(v) \cdot d(\gamma(v), \gamma(u)) \cdot x_{uv} + (1 - s) \sum_{(u,v) \in V^2} w(v) \cdot \delta(u, v) \cdot x_{uv}$$

Variablen:

- $x_{uv} \in \{0, 1\}$ mit $x_{uv} = 1$ gdw. Knoten v zum Zentrum u gehört

Zielfunktion:

minimiere

$$s \sum_{(u,v) \in V^2} w(v) \cdot d(\gamma(v), \gamma(u)) \cdot x_{uv} + (1 - s) \sum_{(u,v) \in V^2} w(v) \cdot \delta(u, v) \cdot x_{uv}$$

Nebenbedingungen:

$$\sum_{u \in V} x_{uv} = 1 \quad \text{für alle } v \in V \quad (8)$$

$$\sum_{v \in V} w(v) \cdot x_{uv} \geq \theta(\gamma(u)) \cdot x_{uu} \quad \text{für alle } u \in V \quad (9)$$

$$\sum_{w \in Pr_u(v)} x_{uw} \geq x_{uv} \quad \text{für alle } (u, v) \in V^2, u \neq v \quad (10)$$

Variablen:

- $x_{uv} \in \{0, 1\}$ mit $x_{uv} = 1$ gdw. Knoten v zum Zentrum u gehört

Zielfunkt **Wie lässt sich Vollständigkeit modellieren?**

minimiere

$$s \sum_{(u,v) \in V^2} w(v) \cdot d(\gamma(v), \gamma(u)) \cdot x_{uv} + (1 - s) \sum_{(u,v) \in V^2} w(v) \cdot \delta(u, v) \cdot x_{uv}$$

Nebenbedingungen:

$$\sum_{u \in V} x_{uv} = 1 \quad \text{für alle } v \in V \quad (8)$$

$$\sum_{v \in V} w(v) \cdot x_{uv} \geq \theta(\gamma(u)) \cdot x_{uu} \quad \text{für alle } u \in V \quad (9)$$

$$\sum_{w \in Pr_u(v)} x_{uw} \geq x_{uv} \quad \text{für alle } (u, v) \in V^2, u \neq v \quad (10)$$

MIP C: Umfangskosten [Haunert, Wolff 2010]

wie MIP B, aber zusätzlich

Variablen:

- $y_{ue} \in [0, 1]$ mit $y_{ue} = 0$ falls mind. ein Endpunkt von $e \in E$ nicht zum Zentrum u gehört

MIP C: Umfangskosten [Hauert, Wolff 2010]

wie MIP B, aber zusätzlich

Variablen:

- $y_{ue} \in [0, 1]$ mit $y_{ue} = 0$ falls mind. ein Endpunkt von $e \in E$ nicht zum Zentrum u gehört

Nebenbedingungen:

$$y_{ue} \leq x_{uw} \quad y_{ue} \leq x_{uv} \quad \text{für alle } u \in V, e = \{v, w\} \in E \quad (11)$$

wie MIP B, aber zusätzlich

Variablen:

- $y_{ue} \in [0, 1]$ mit $y_{ue} = 0$ falls mind. ein Endpunkt von $e \in E$ nicht zum Zentrum u gehört

Nebenbedingungen:

$$y_{ue} \leq x_{uw} \quad y_{ue} \leq x_{uv} \quad \text{für alle } u \in V, e = \{v, w\} \in E \quad (11)$$

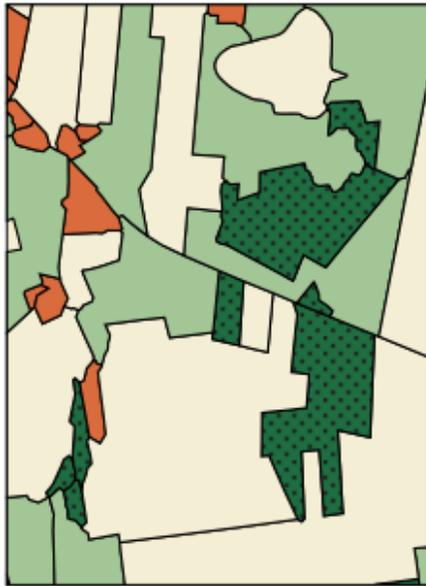
Zielfunktion:

minimiere

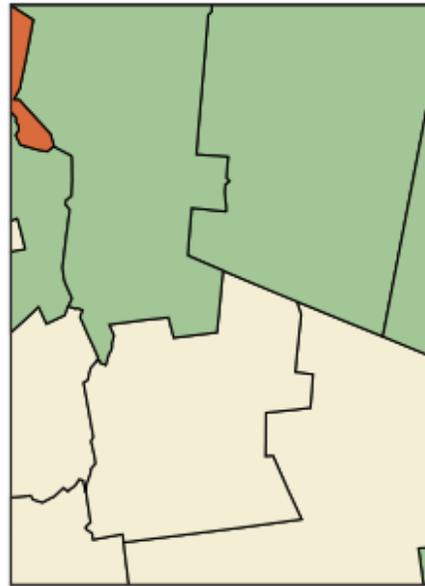
$$s \sum_{(u,v) \in V^2} w(v) \cdot d(\gamma(v), \gamma(u)) \cdot x_{uv} - (1 - s) \cdot 2 \sum_{u \in V} \sum_{e \in E} \lambda(e) \cdot y_{ue}$$

wobei $\lambda: E \rightarrow \mathbb{R}^+$ die Kantenlängen angibt

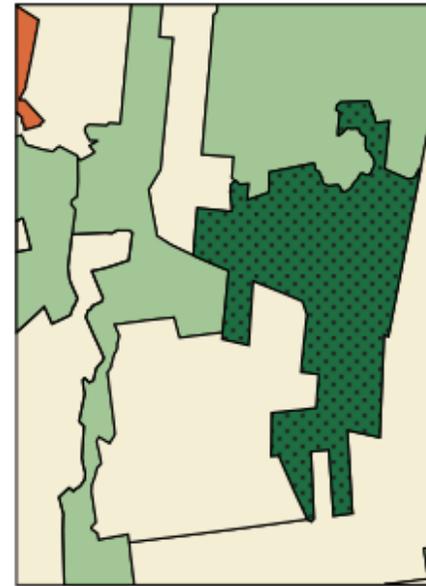
Visueller Eindruck:



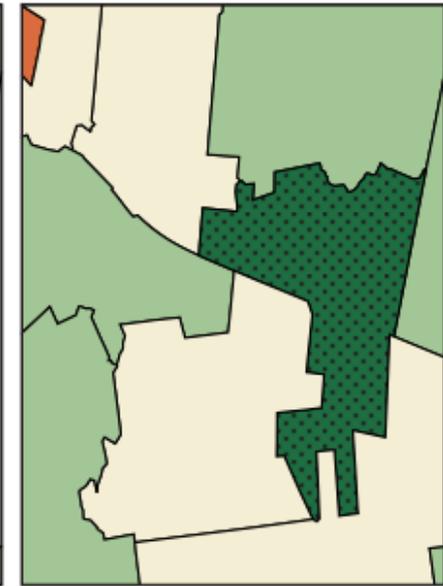
Eingabe



Region Growing



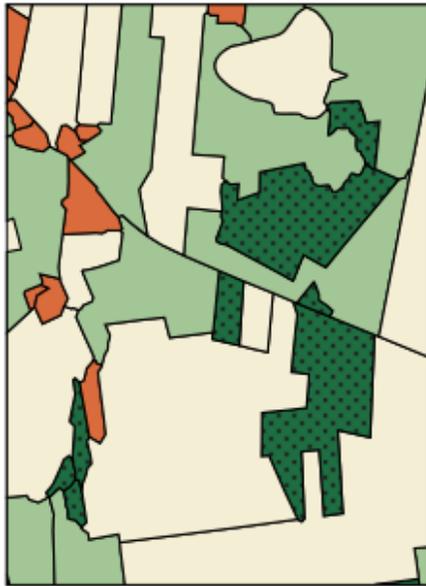
Semantik



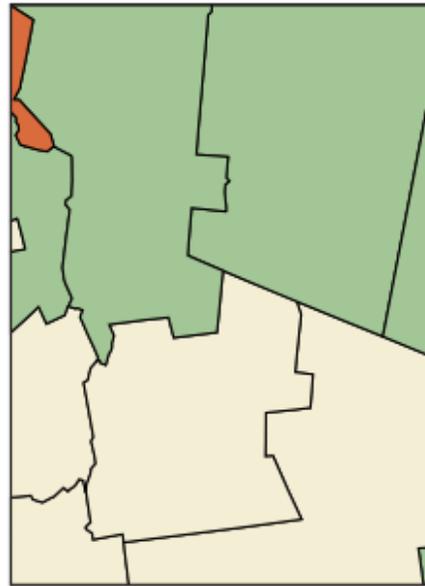
Semantik + Kompaktheit

Evaluation

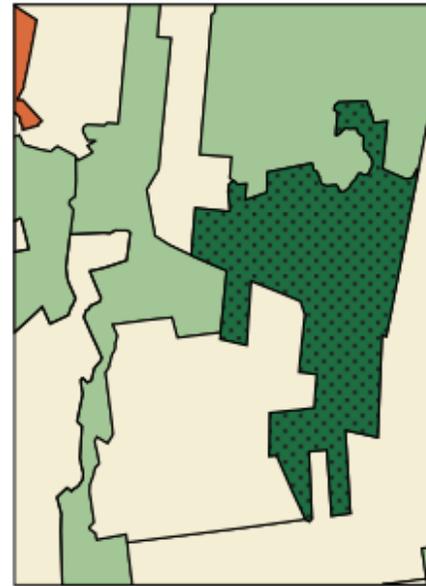
Visueller Eindruck:



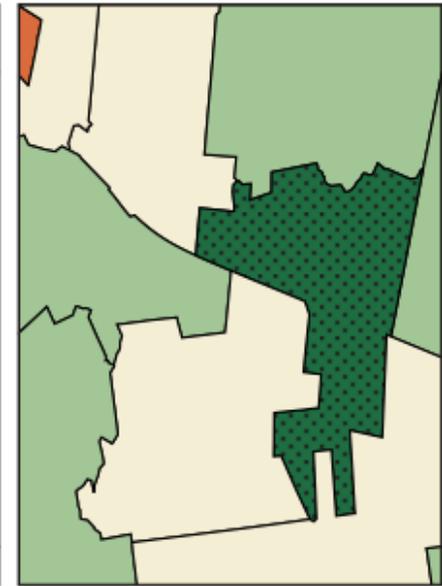
Eingabe



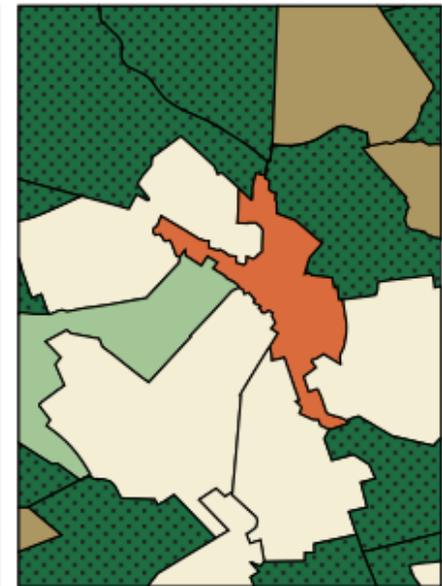
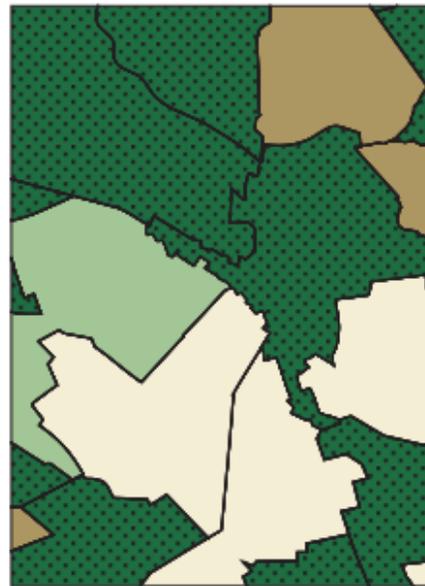
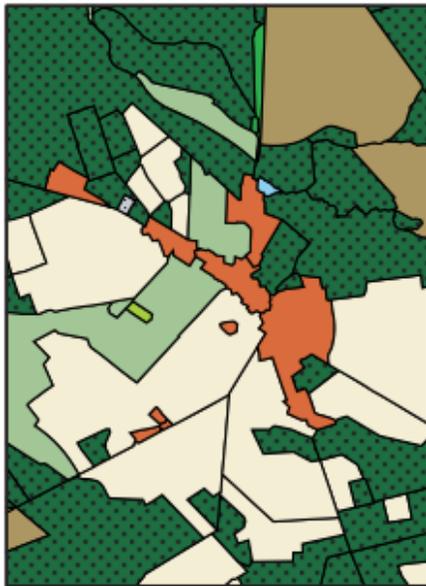
Region Growing



Semantik



Semantik + Kompaktheit



Performance:

- MIP A: recht langsam, selbst für kleinere Instanzen
- MIP B/C: deutlich schneller, aber erst durch zusätzliche Heuristiken werden realistische Instanzgrößen erreicht

nodes	Alg. 1 \bar{d}	flow MIP				P-R MIP							
		pure		centre heuristic		pure		distance heuristic		centre heuristic		centre + dist. heuristic	
		time	\bar{d}	time	\bar{d}	time	\bar{d}	time	\bar{d}	time	\bar{d}	time	\bar{d}
30	29.30	223.2	9.20	8.8	9.20	2.8	9.20	3.0	9.20	0.04	12.81	0.01	12.81
40	22.31	3.9 h	6.97	41.6	6.97	31.1	7.34	16.8	7.34	0.04	7.59	0.03	7.59
50	15.30	*20.0 h	5.18	432.1	5.18	60.9	5.18	62.5	5.18	0.34	5.64	0.15	5.64
60	15.49					253.0	5.85	236.0	5.85	0.45	6.66	0.72	6.66
100	12.48									33.1	5.93	12.5	5.93
200	9.85									84.3	4.68	87.0	4.68
300	6.92									267.2	4.50	363.3	4.50
400	7.04									397.7	4.64	346.1	4.64

aus [Hauert, Wolff 2010]