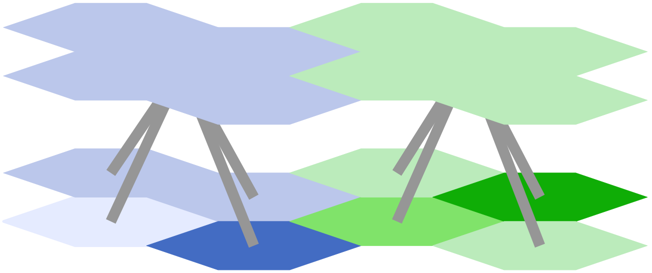
Moscow, 3.8.2007

EFFICIENT AREA AGGREGATION BY COMBINATION OF DIFFERENT TECHNIQUES

Jan-Henrik Haunert

Institut für Kartographie und Geoinformatik
Leibniz Universität Hannover



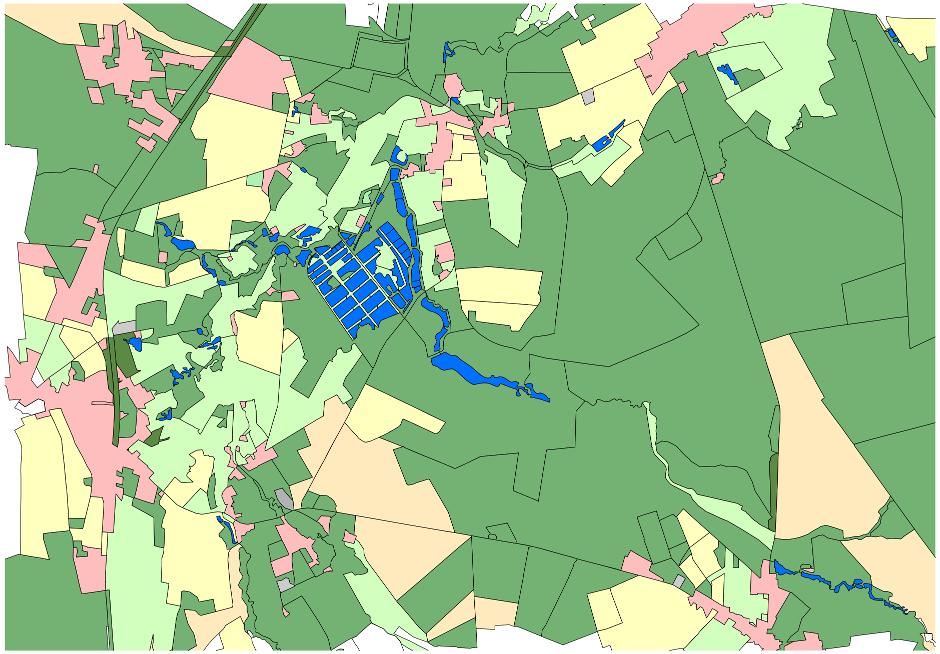
Outline

1. Introduction
2. Problem definition
3. Results by mixed-integer programming
4. Problem decomposition
5. Algorithm
6. Results

1. Introduction

Areas in topographic databases (e.g. German ATKIS)

- commonly define a planar subdivision.
- are often collected independently for different scales.



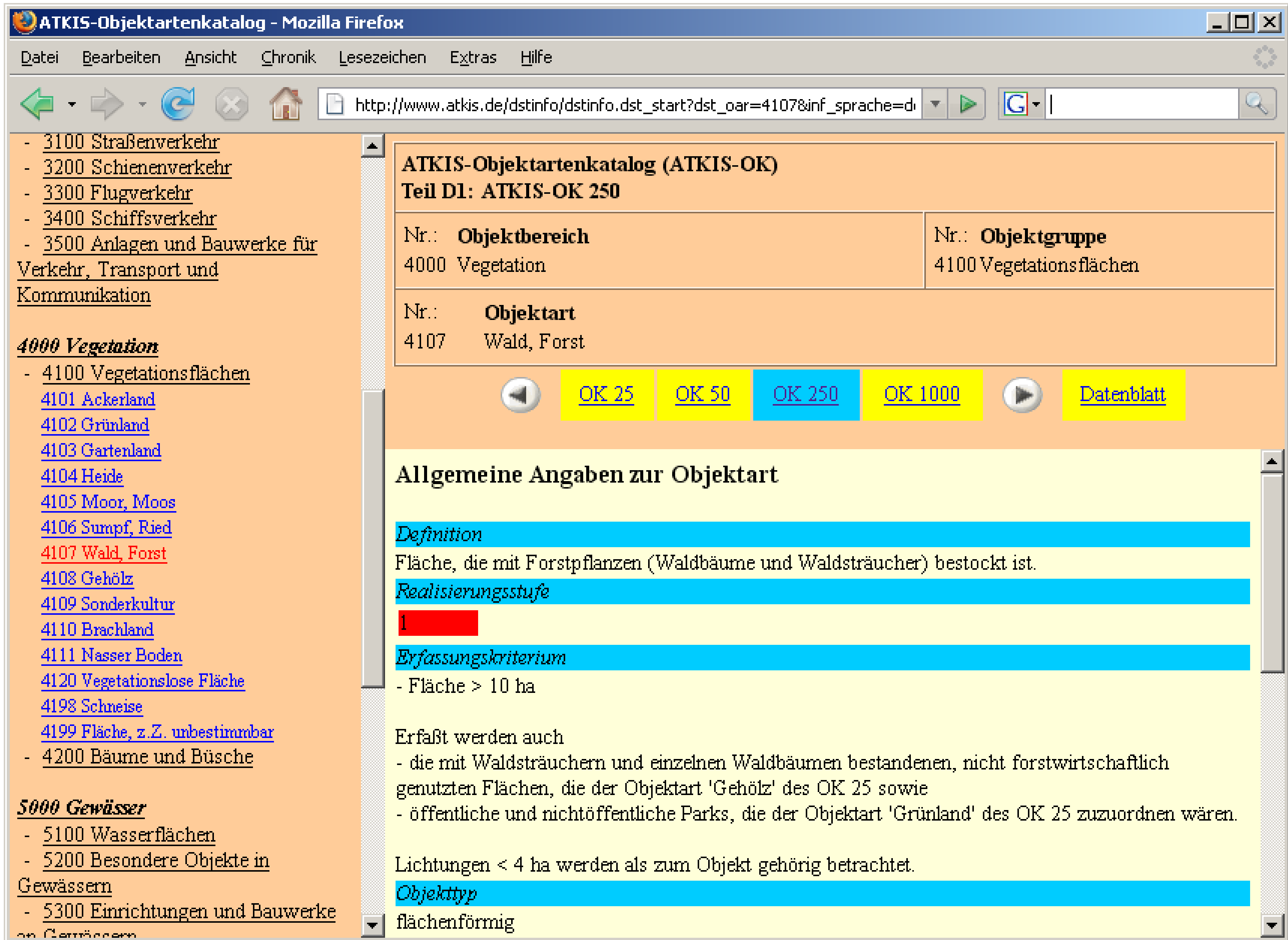
Sample from ATKIS DLM50

Automatic generalisation

- enables to collect data sets more efficiently.
- requires algorithms for different problems (area collapse, aggregation, line simplification).

1. Introduction

Generalisation must satisfy constraints defined in specifications:



1. Introduction

Existing aggregation algorithm (vanOosterom1995):

---

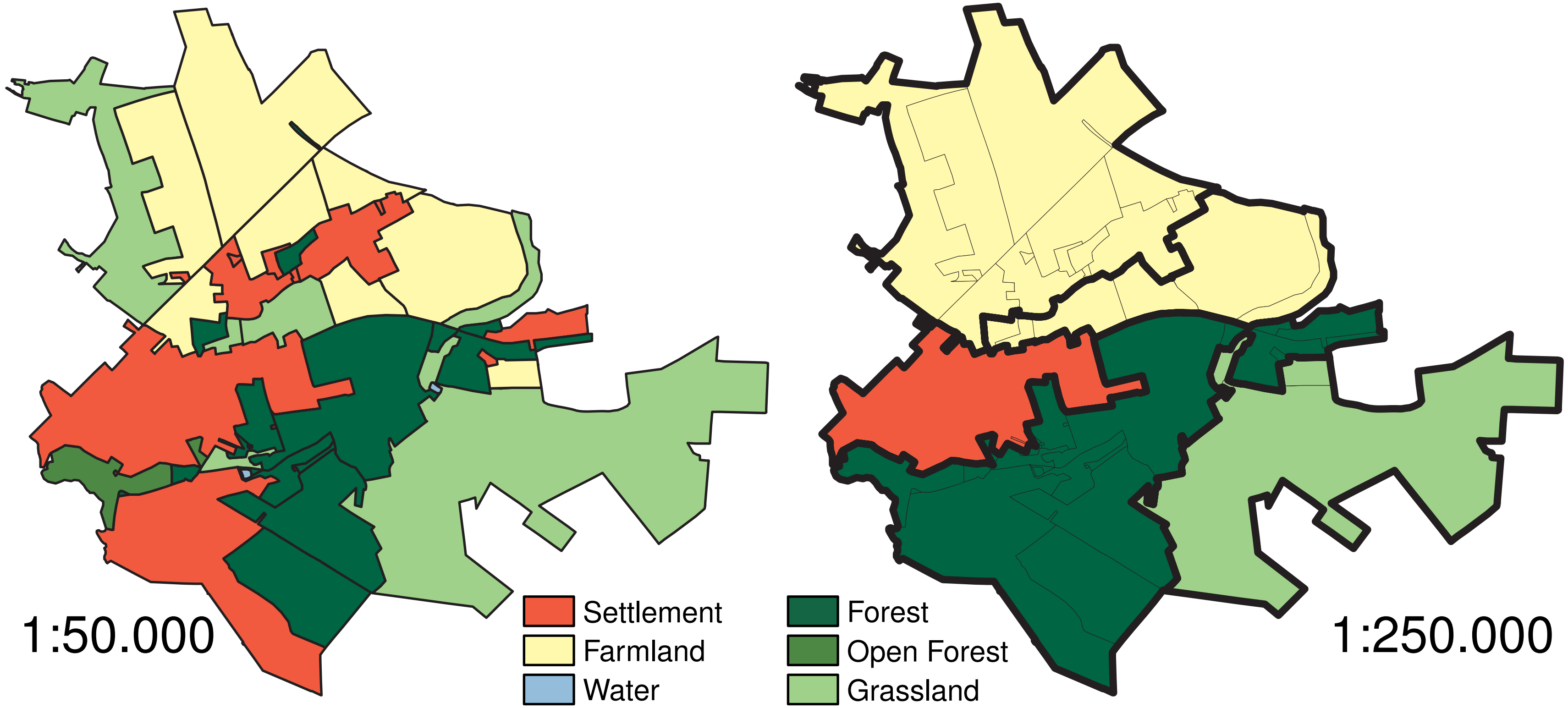
a = smallest area in data set

while a is smaller than required for target scale **do**

  merge a with best compatible neighbour

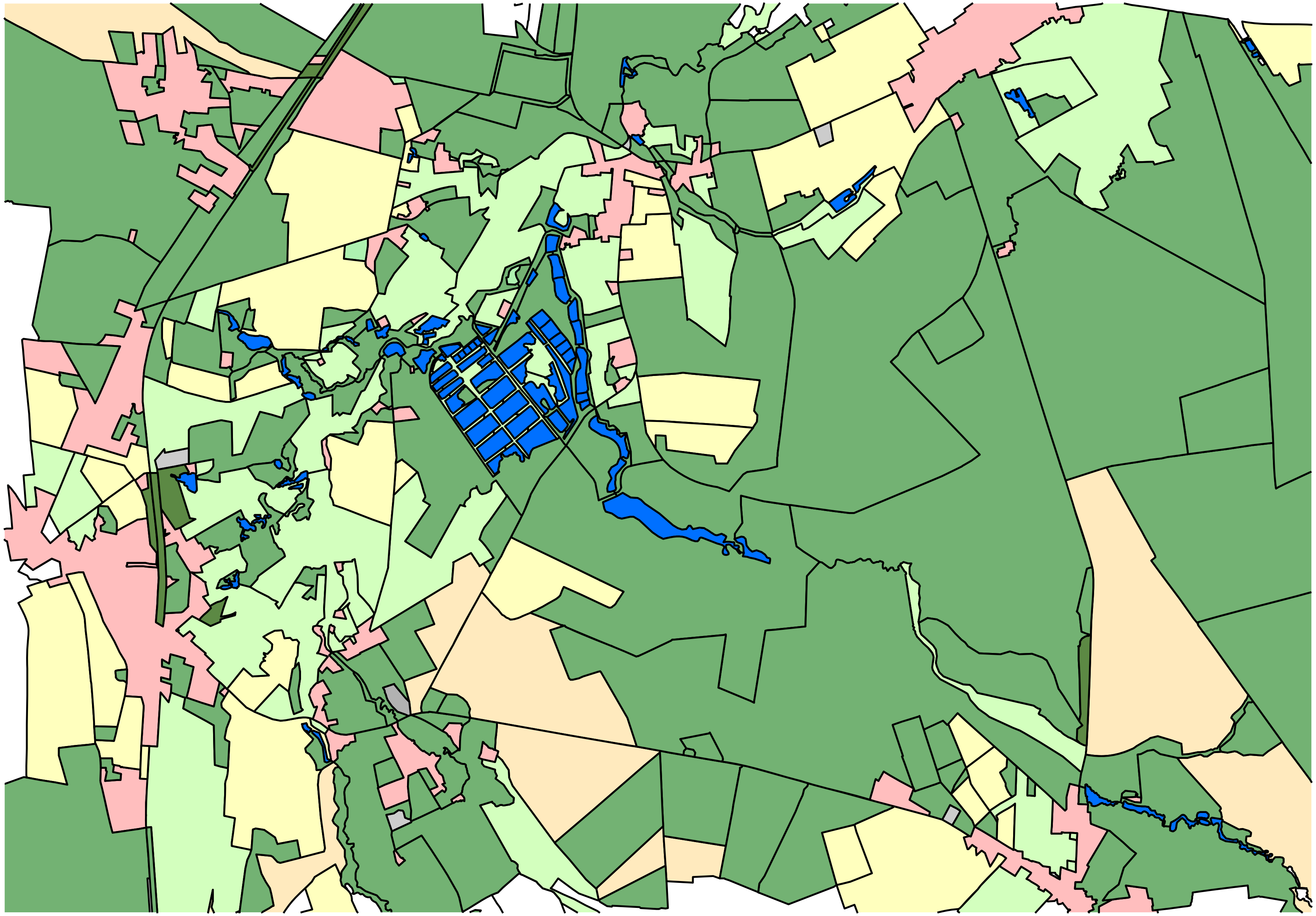
  a = smallest area in data set

end while

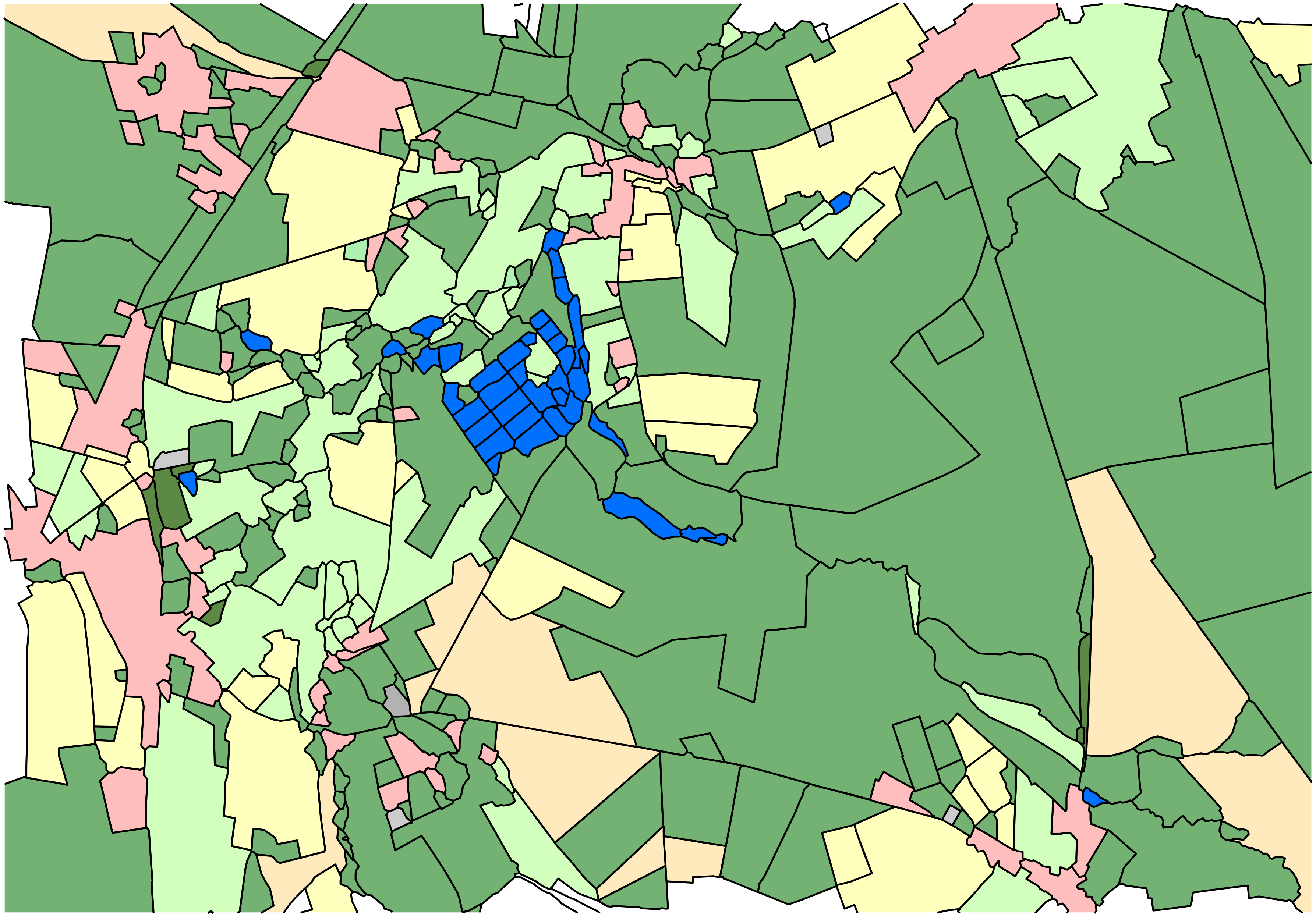


1. Introduction

Preprocessing: Input

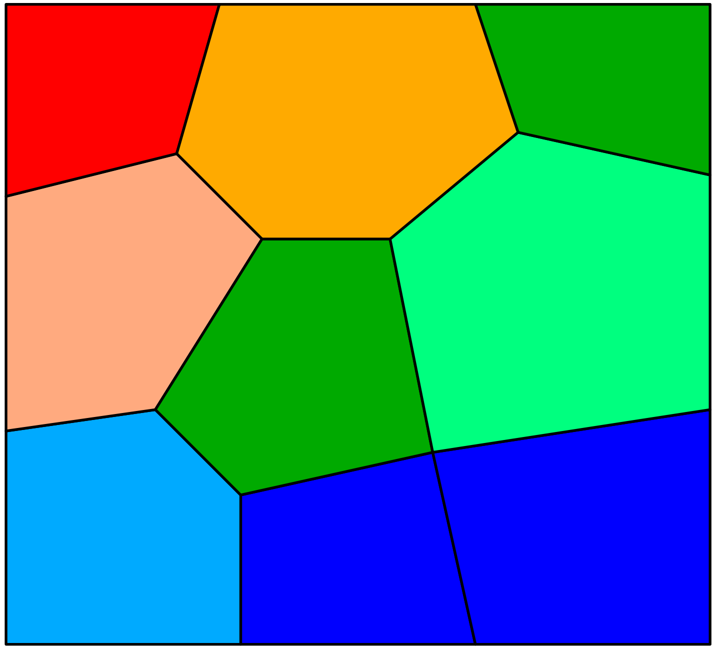


Preprocessing: Result



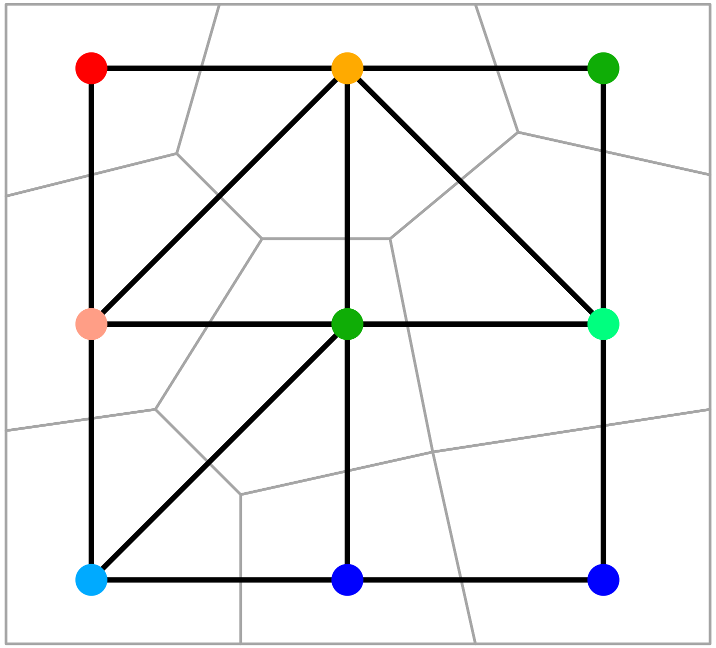
2. Problem definition

Graph representation of a planar subdivision



2. Problem definition

Graph representation of a planar subdivision

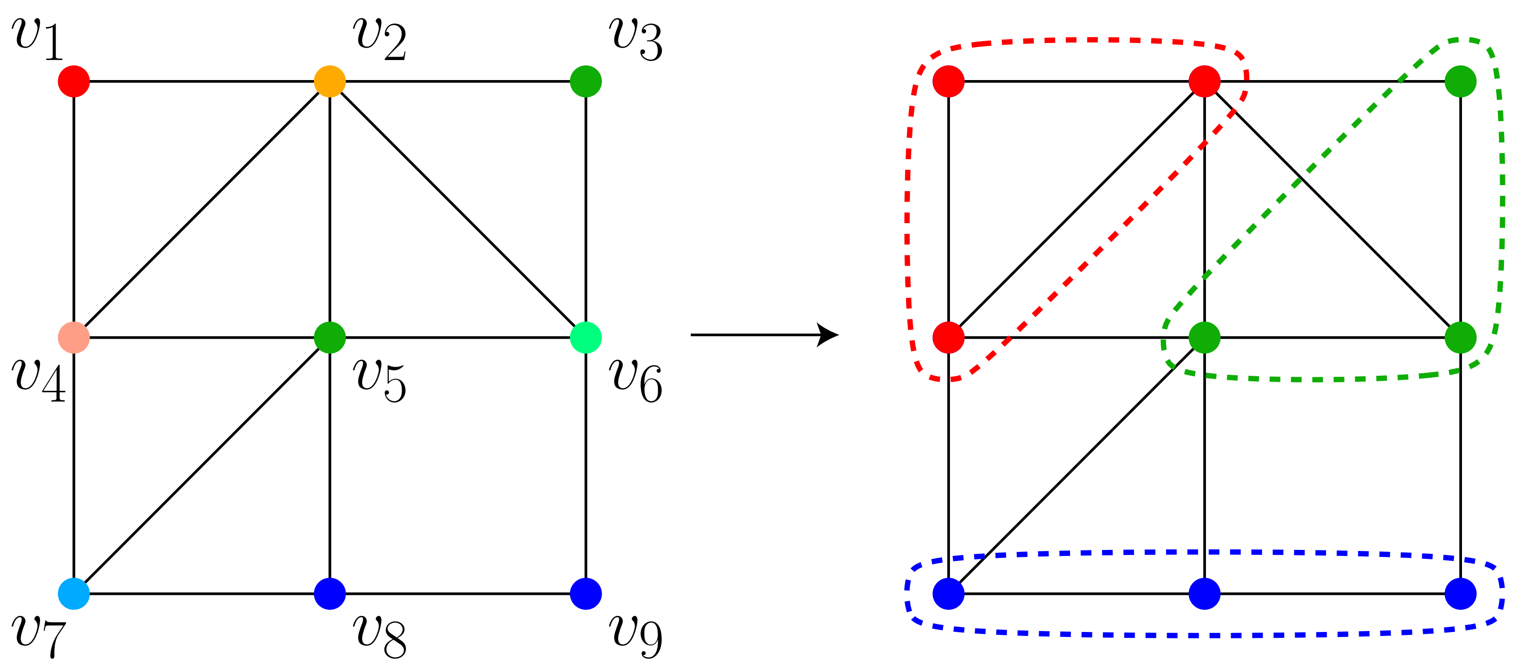


A planar subdivision is represented by

- Adjacency graph $G(V, E)$
- Node weights $w : V \rightarrow \mathbb{R}^+$ (area sizes)
- Node colours $\gamma : V \rightarrow \Gamma$ (land cover classes)

Problem:

$$P = \{\{v_1, v_2, v_4\}, \{v_3, v_5, v_6\}, \{v_7, v_8, v_9\}\}$$

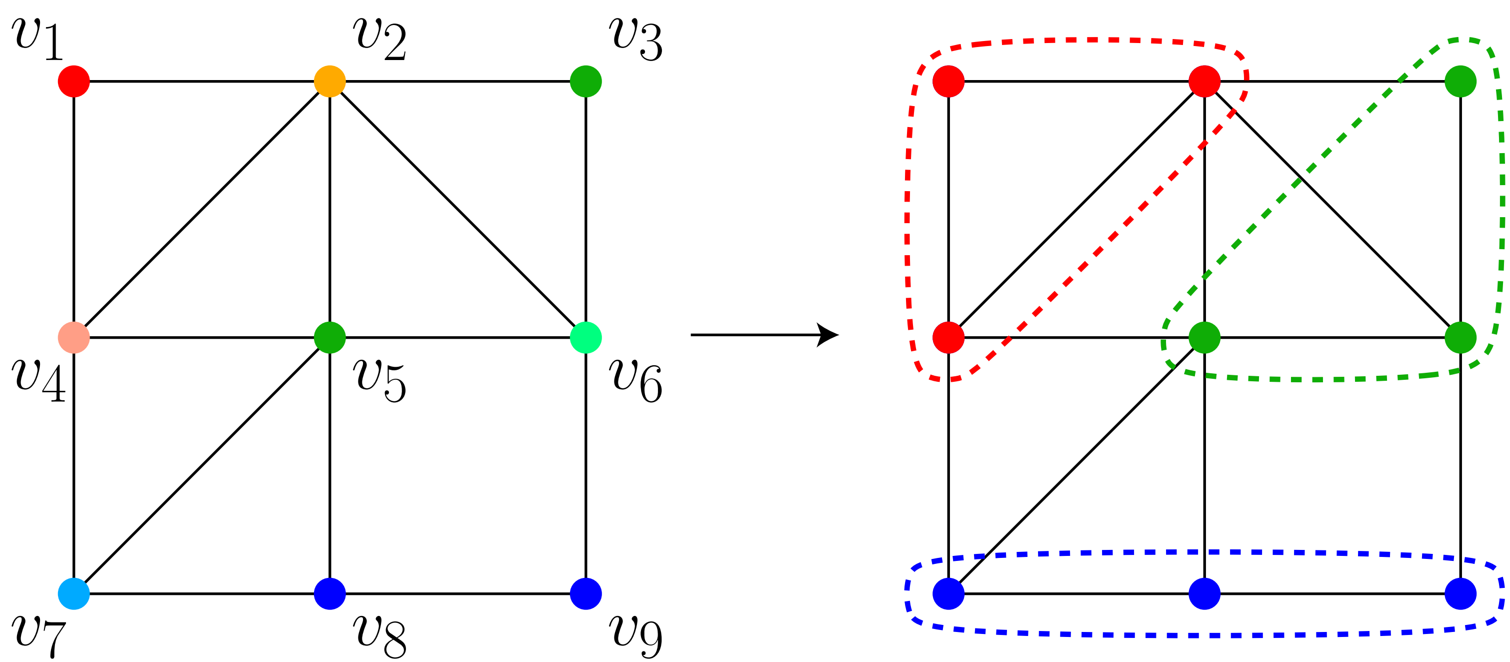


Define new colours $\gamma' : V \to \Gamma$ and a partition $P = \{V_1, V_2, \dots, V_n\}$ of V,

2. Problem definition

Problem:

$P = \{\{v_1, v_2, v_4\}, \{v_3, v_5, v_6\}, \{v_7, v_8, v_9\}\}$



Define new colours $\gamma' : V \to \Gamma$ and a partition $P = \{V_1, V_2, \ldots, V_n\}$ of V, such that for each aggregate $V_i \in P$

- the induced graph is connected,
- all contained nodes have the same new colour γ'_i,
- at least one contained node keeps its original colour, and
- the total weight of V_i is at least $\theta(\gamma'_i)$,

with $\theta : \Gamma \to \mathbb{R}^+$ defining the area thresholds for different colours.

2. Problem definition

Objective:

1. Change land cover classes as little as possible.

   Minimise $\sum_{v \in V} w(v) \cdot d(\gamma(v), \gamma'(v))$,

   with $d : \Gamma^2 \to \mathbb{R}_0^+$ defining the similarity of land cover classes.

2. Problem definition

Objective:

2. Geometrically compact regions are preferred.

Minimise $\sum_{V_i \in P} c(V_i, \gamma_i')$,

with $c : 2^V \times \Gamma \to \mathbb{R}_0^+$ defining a cost for the non-compactness of an aggregate.

Definition:
Each aggregate has a **centre**, which defines the colour of the aggregate and a central reference point, which is used to measure compactness.

Each node is a potential centre.

2. Problem definition

Objective:

1. & 2. combined

Minimise $s \cdot \sum_{v \in V} w(v) \cdot d(\gamma(v), \gamma'(v)) + (1-s) \cdot \sum_{V_i \in P} c(V_i, \gamma'_i)$,

with $s \in \mathbb{R}, 0 < s < 1$.

2. Problem definition

Complexity:

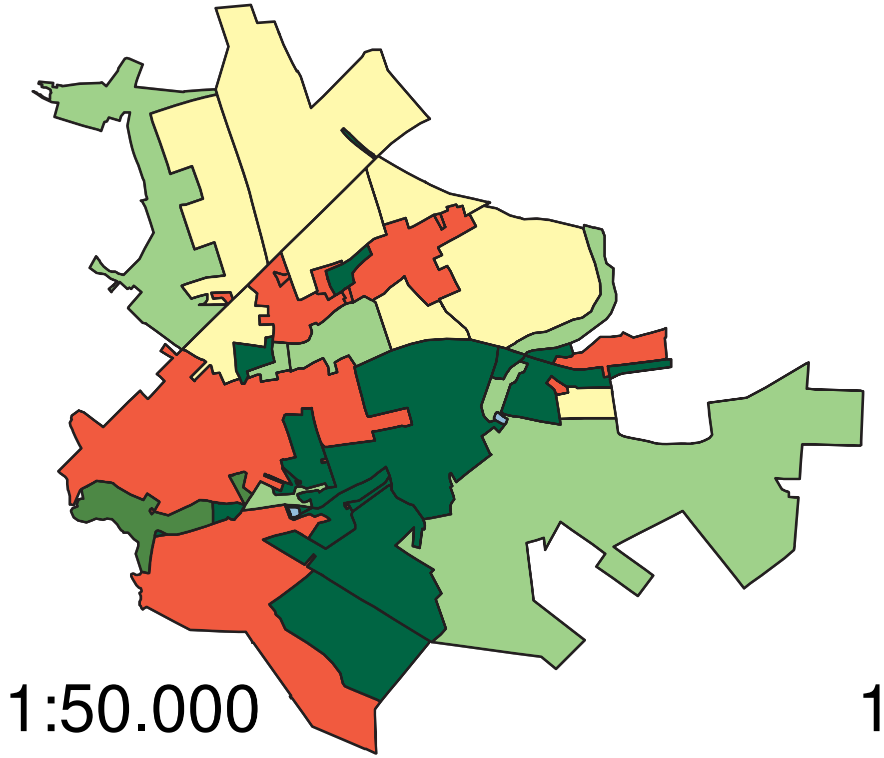
The aggregation problem ist NP-hard, even if

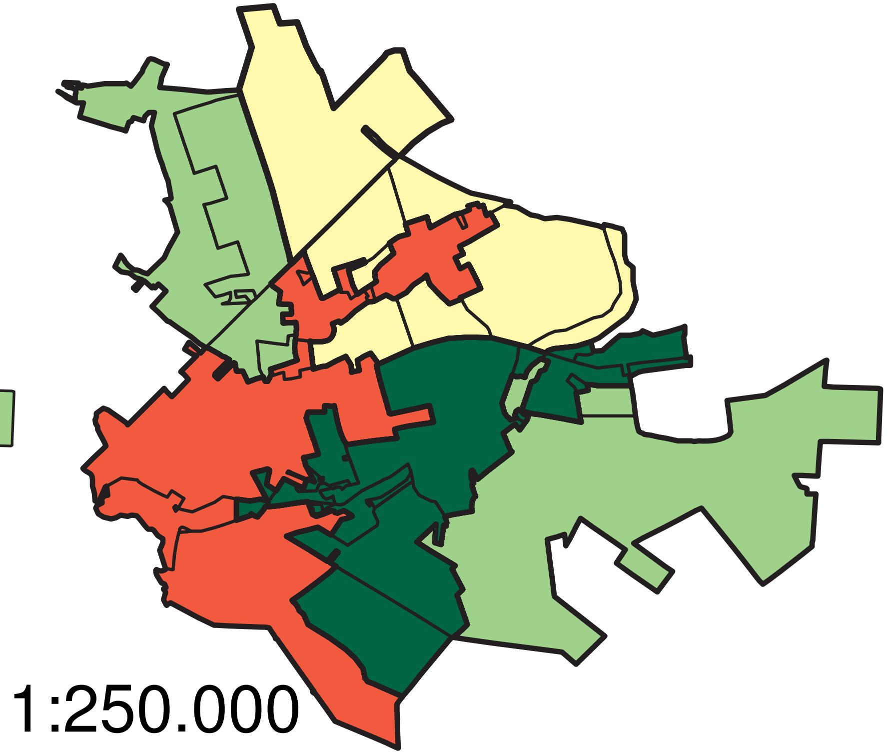
– the map has only two colours,
– the area threshold is the same for all colours,
– the distances between each two different colours are the same,
– the objective of compactness is ignored, and
– the weights of all nodes are equal.

Motivation for mixed-integer programming and heuristic methods

3. Results by mixed-integer programming

Mixed-integer programming





- Performance is not appropriate for cartographic production.
- Introduce heuristics, which allow
  - other MIP formulations
  - elimination of some variables

3. Results by mixed-integer programming

Results

exact MIP

nodes	time	objective
30	90.2s	1.73
40	12.7h	1.67
50	20.0h	2.15

MIP with heuristics

nodes	time	objective
30	0.01s	2.41
40	0.03s	1.82
50	0.45s	2.34
400	22min	19.15

Iterative algorithm

nodes	objective
30	5.51
40	5.35
50	6.35
400	29.04

MIP with heuristics

- produces near-optimal solutions and
- solves instance with 400 areas in reasonable time.

Assumption for further approach:

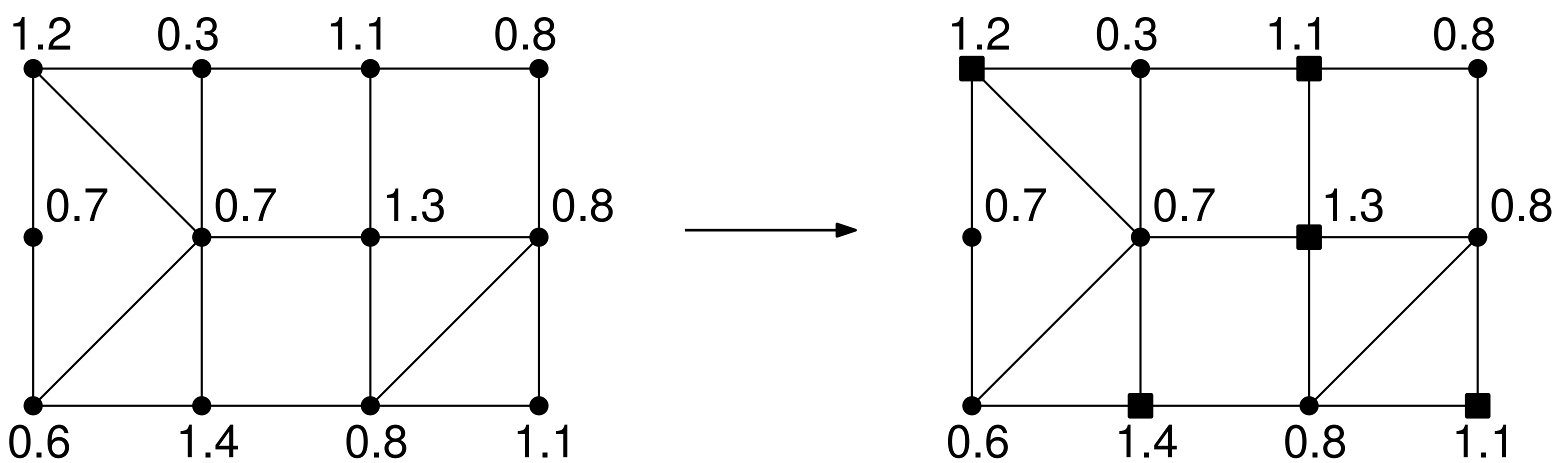
We can solve instances with k or less nodes (not restricted to any specific technique like mixed-integer programming).

4. Problem decomposition

Approach based on a heuristic

- Assumption: Large areas can be fixed as centres without loosing good solutions.
- In particular this can be done for nodes $v \in V$ with $w(v) > \theta(\gamma(v))$, i.e., areas that are sufficiently large for the target scale.

Example:



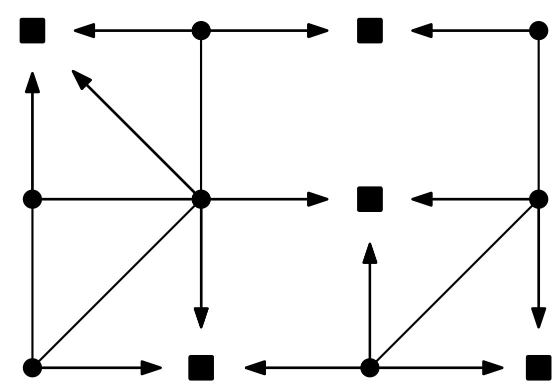
thresholds $\theta(\gamma) = 1$ for all $\gamma \in \Gamma$

4. Problem decomposition

Approach based on a heuristic

- Assumption: Large areas can be fixed as centres without loosing good solutions.
- In particular this can be done for nodes $v \in V$ with $w(v) > \theta(\gamma(v))$, i.e., areas that are sufficiently large for the target scale.

- When fixing nodes with $w(v) \geq \theta(\gamma(v))$ as centres, the aggregation problem can be solved independently for each connected component of the graph induced by the other nodes.



4. Problem decomposition

Meaning for mixed-integer programming:

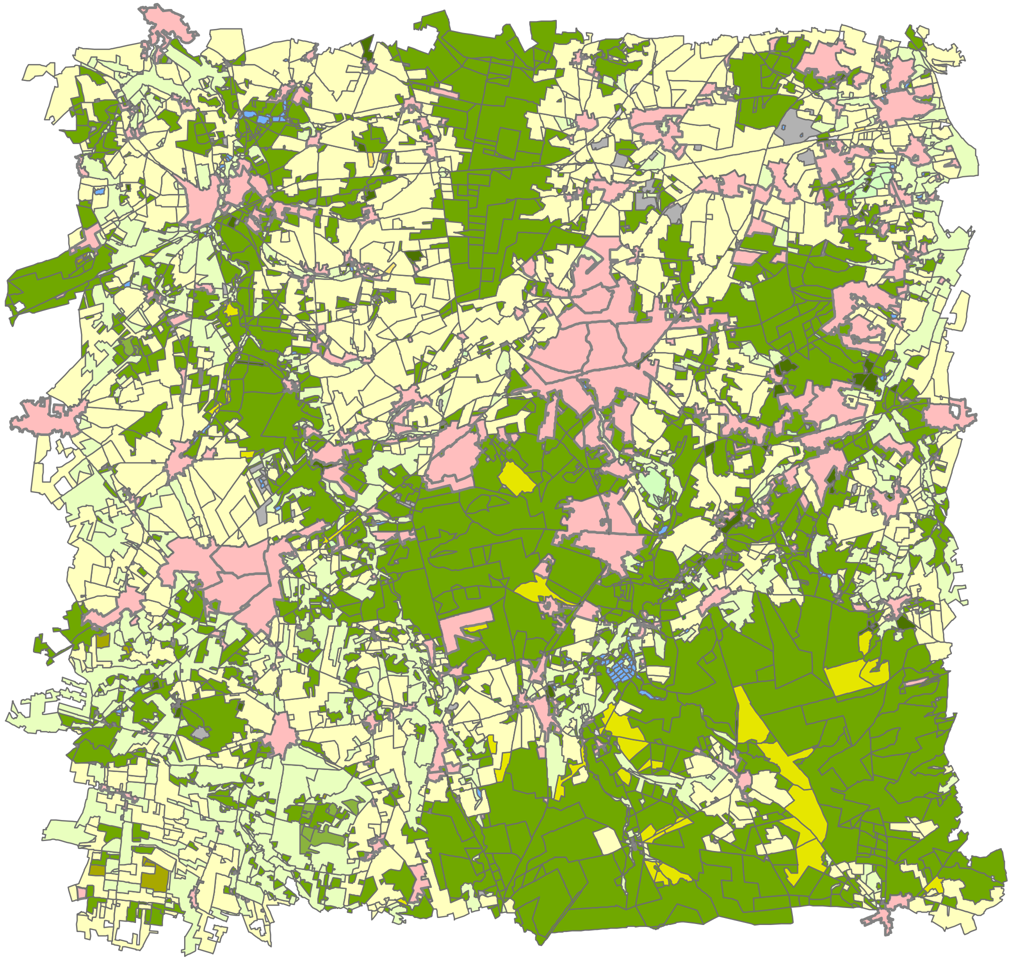
- Constraints have general form $A \cdot x \leq b$
- If the graph has two connected components, the matrix A will has form $\begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$

4. Problem decomposition

Just theory?

4. Problem decomposition

ATKIS DLM 1:50.000
BUCHHOLZ IN DER NORDHEIDE
5537 polygons (after preprocessing)
$20 \times 20\ \mathrm{km}^2$

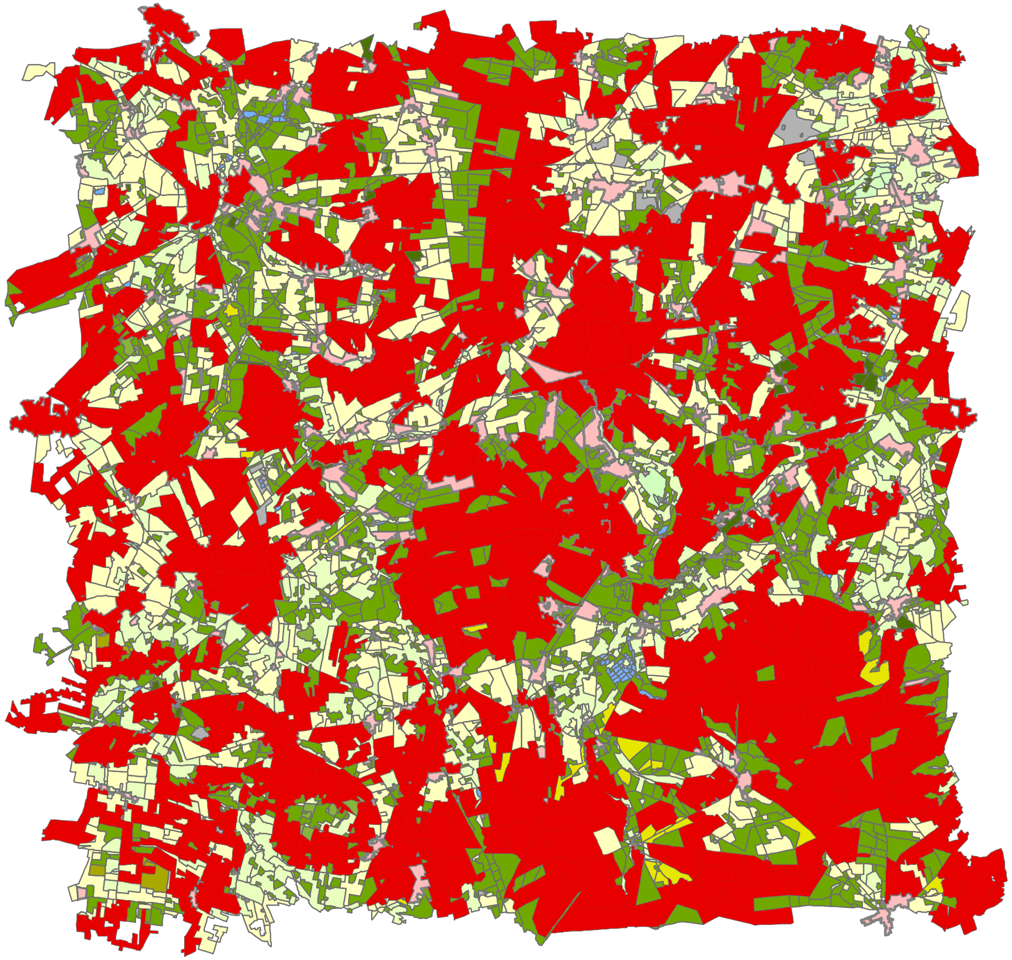


4. Problem decomposition

ATKIS DLM 1:50.000
BUCHHOLZ IN DER NORDHEIDE
5537 polygons (after preprocessing)
20×20 km^2

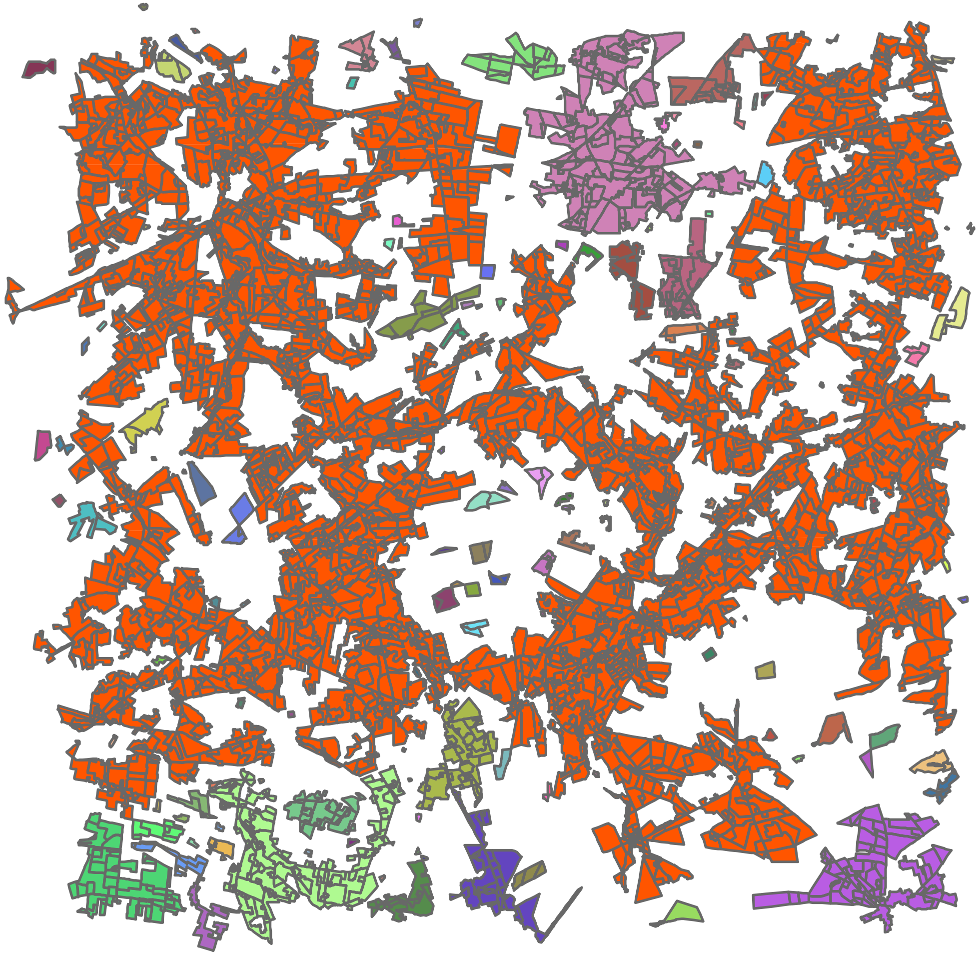
Areas of sufficient size for scale 1:250.000 (red):

7% of all polygons
49% of area coverage



4. Problem decomposition

- 145 independent problem instances
- 1 big component with 76% of all polygons
- all other instances can be solved with heuristic MIP ($|V| \leq 198$)



4. Problem decomposition

Idea:

- Introduce intermediate scales/thresholds
- The number of predefined centres will increase
- The resulting instances will become smaller

- Intermediate scales should only be introduced if needed.
- It needs to be ensured that the instances have size at most k.

5. Algorithm

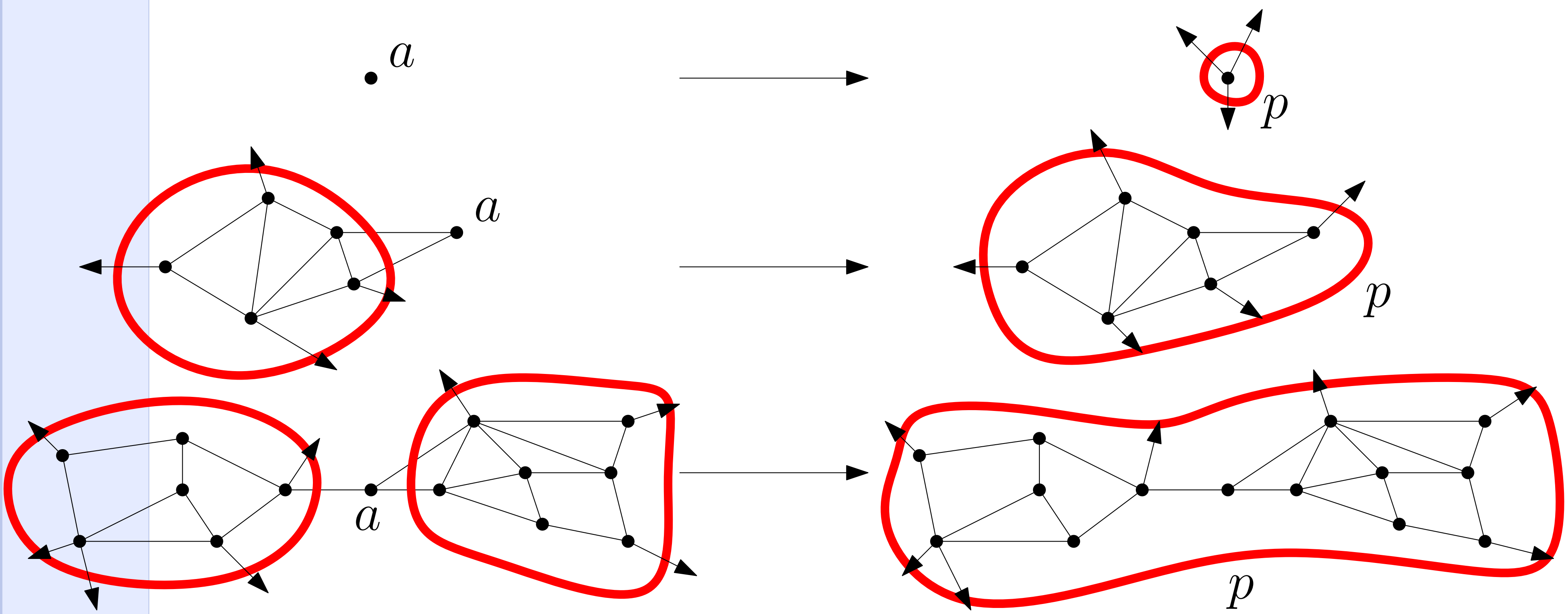
P = a set of open problem instances, initially empty
a = smallest area in data set
while a is smaller than required for the target scale **do**

P'= the set of problem instances in P containing a neighbour of a

if total number of areas contained in instances $P' < k$ **then**

Remove all instances in P' from P

Create a new instance p comprising all areas in instances P' plus a



5. Algorithm

P = a set of open problem instances, initially empty
a = smallest area in data set
while a is smaller than required for the target scale **do**

P'= the set of problem instances in P containing a neighbour of a

if total number of areas contained in instances $P' < k$ **then**

Remove all instances in P' from P

Create a new instance p comprising all areas in instances P' plus a

if p contains k areas **then**

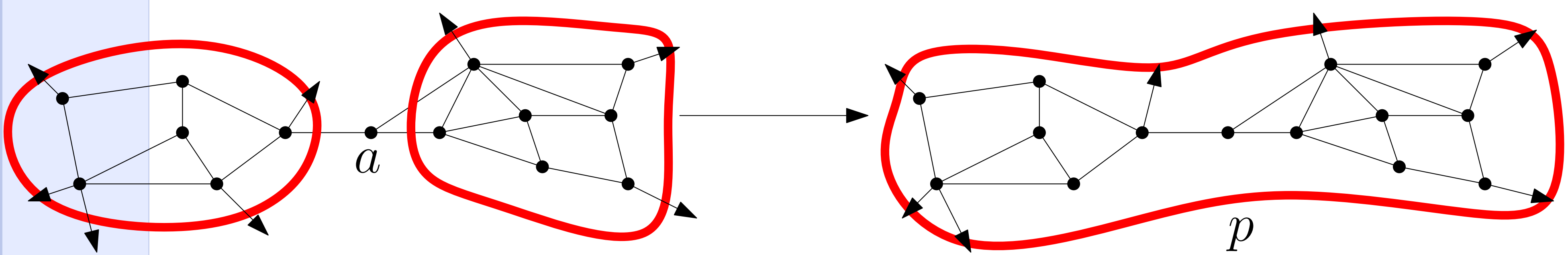
solve p (θ = size of smallest centre in neighbourhood)

else

insert p to P

end if

else



5. Algorithm

P = a set of open problem instances, initially empty
a = smallest area in data set
while a is smaller than required for the target scale **do**

  P'= the set of problem instances in P containing a neighbour of a

  if total number of areas contained in instances $P' < k$ **then**

    Remove all instances in P' from P

    Create a new instance p comprising all areas in instances P' plus a

    if p contains k areas **then**

      solve p (θ = size of smallest centre in neighbourhood)

    else

      insert p to P

    end if

  else

    Solve the instance in P' containing most areas ($\theta = w(a)$).

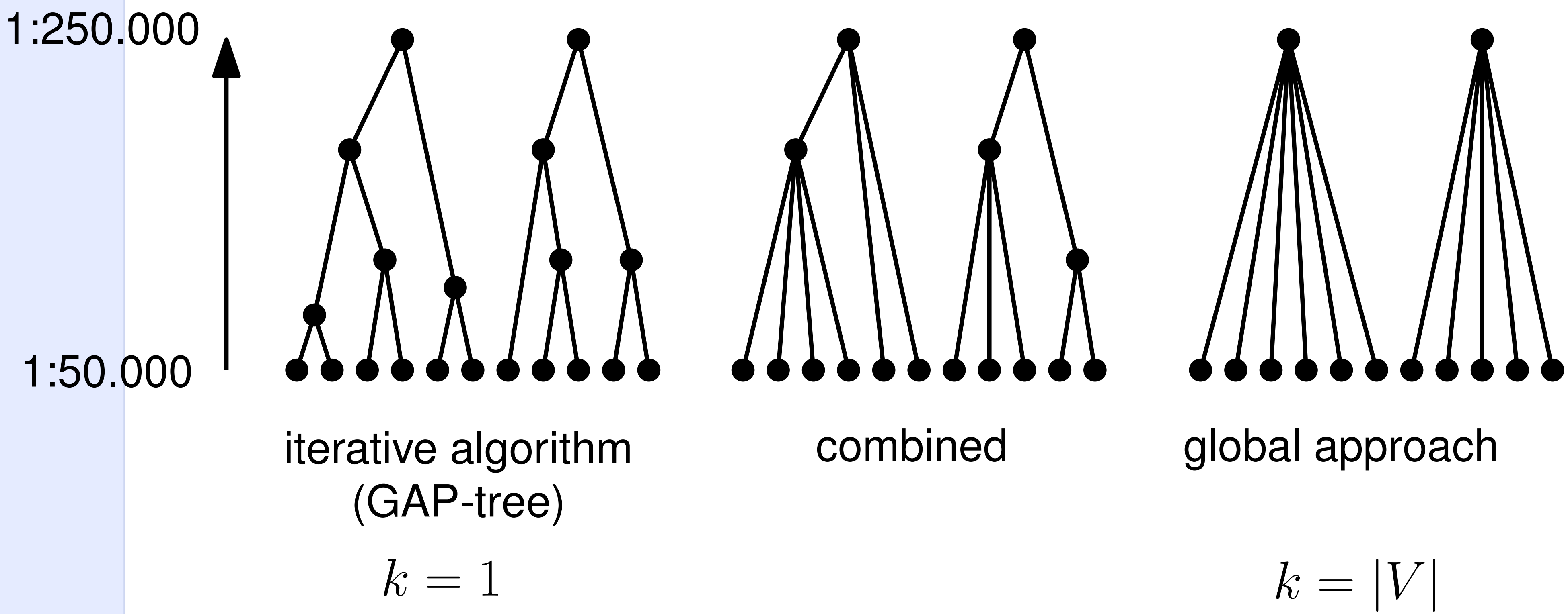
    Remove this instance from P.

  end if

  a = smallest area in data set, not contained in instances P

end while
Solve all remaining instances in P, applying threshold for target scale.

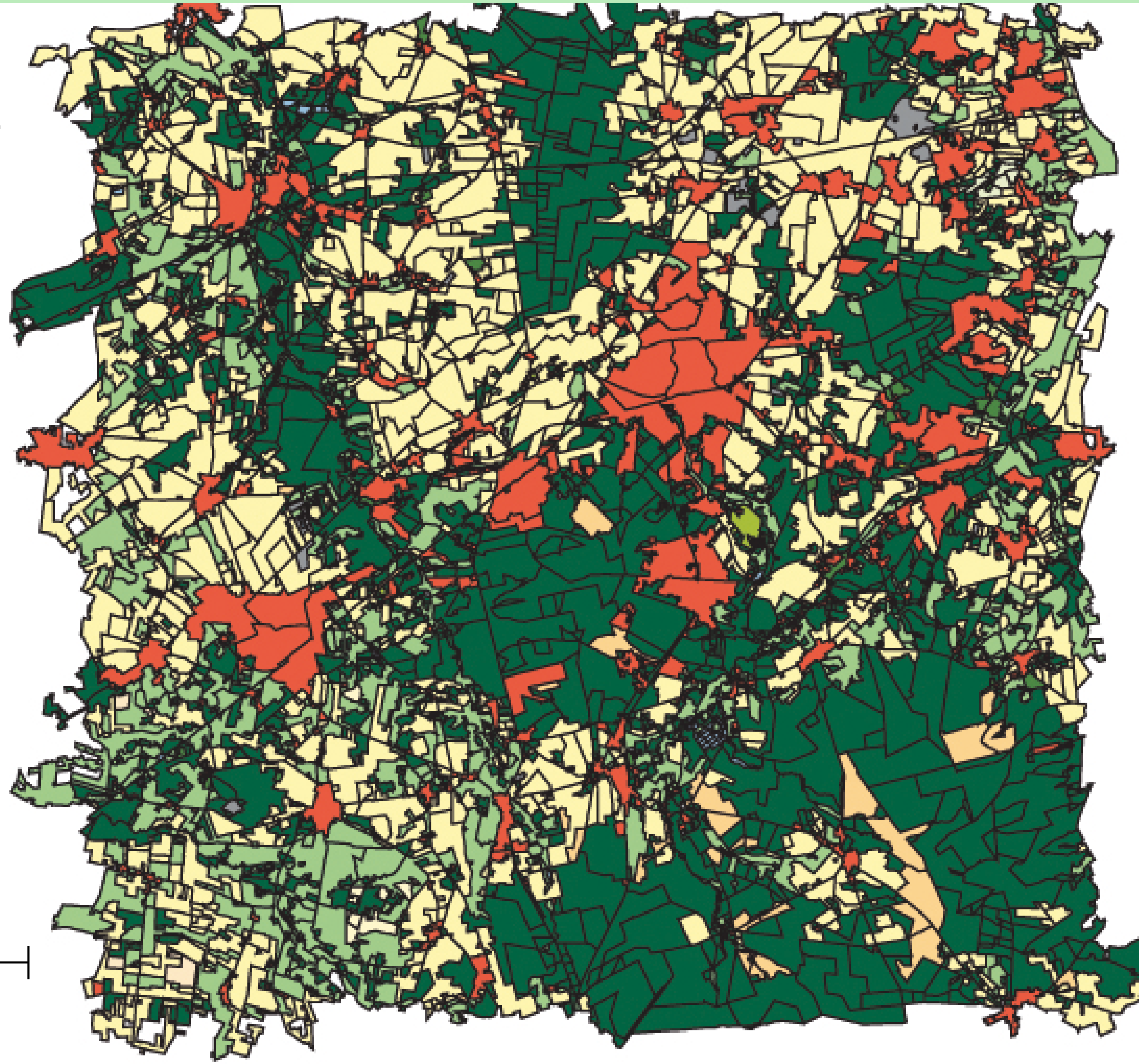
Interpretation of proposed algorithm:



It is likely that good solutions can be obtained when foreseeing more merges than one.

6. Results

ATKIS DLM50
Buchholz in der
Nordheide



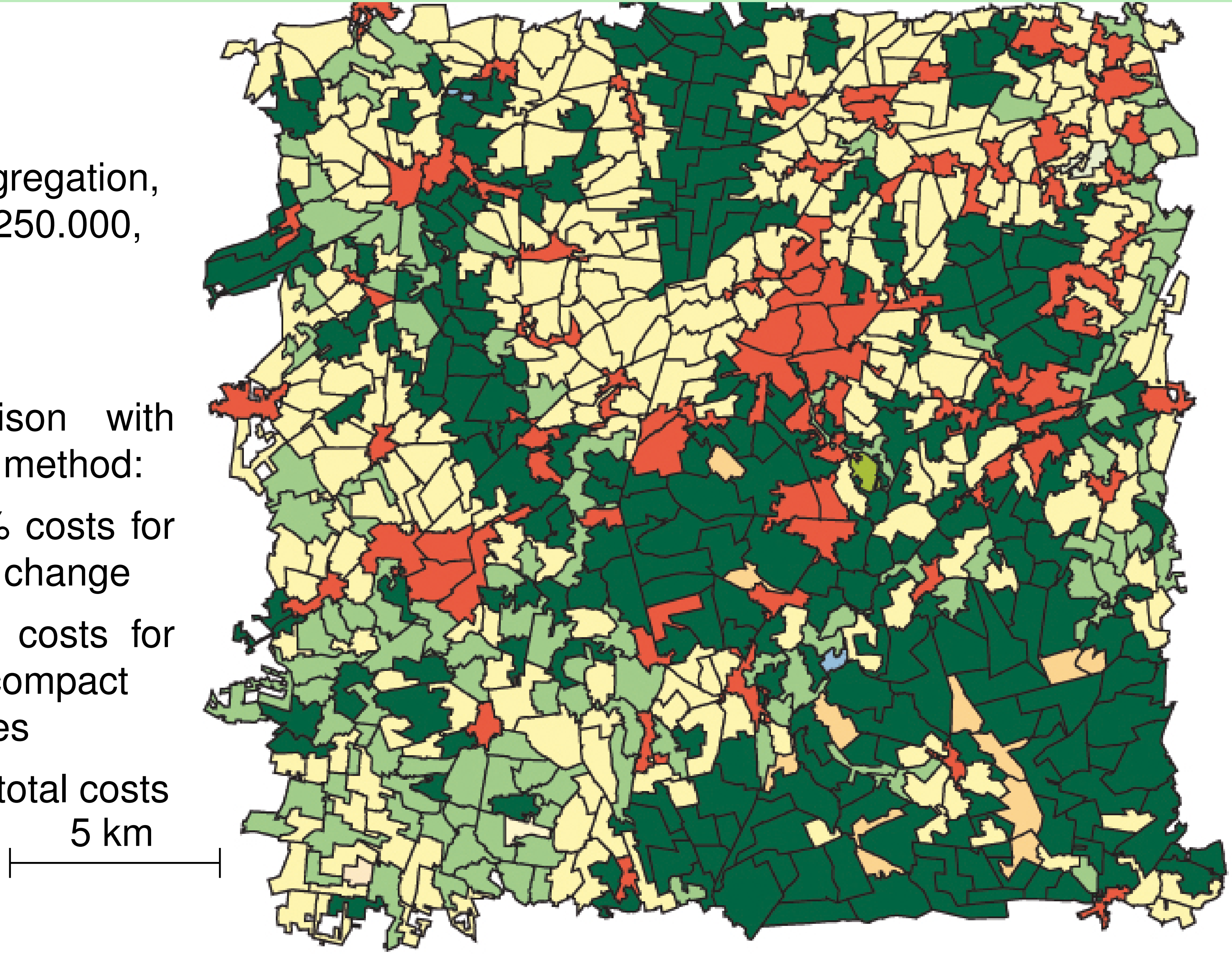
5 km

6. Results

after aggregation, scale 1:250.000, k = 200
82 min

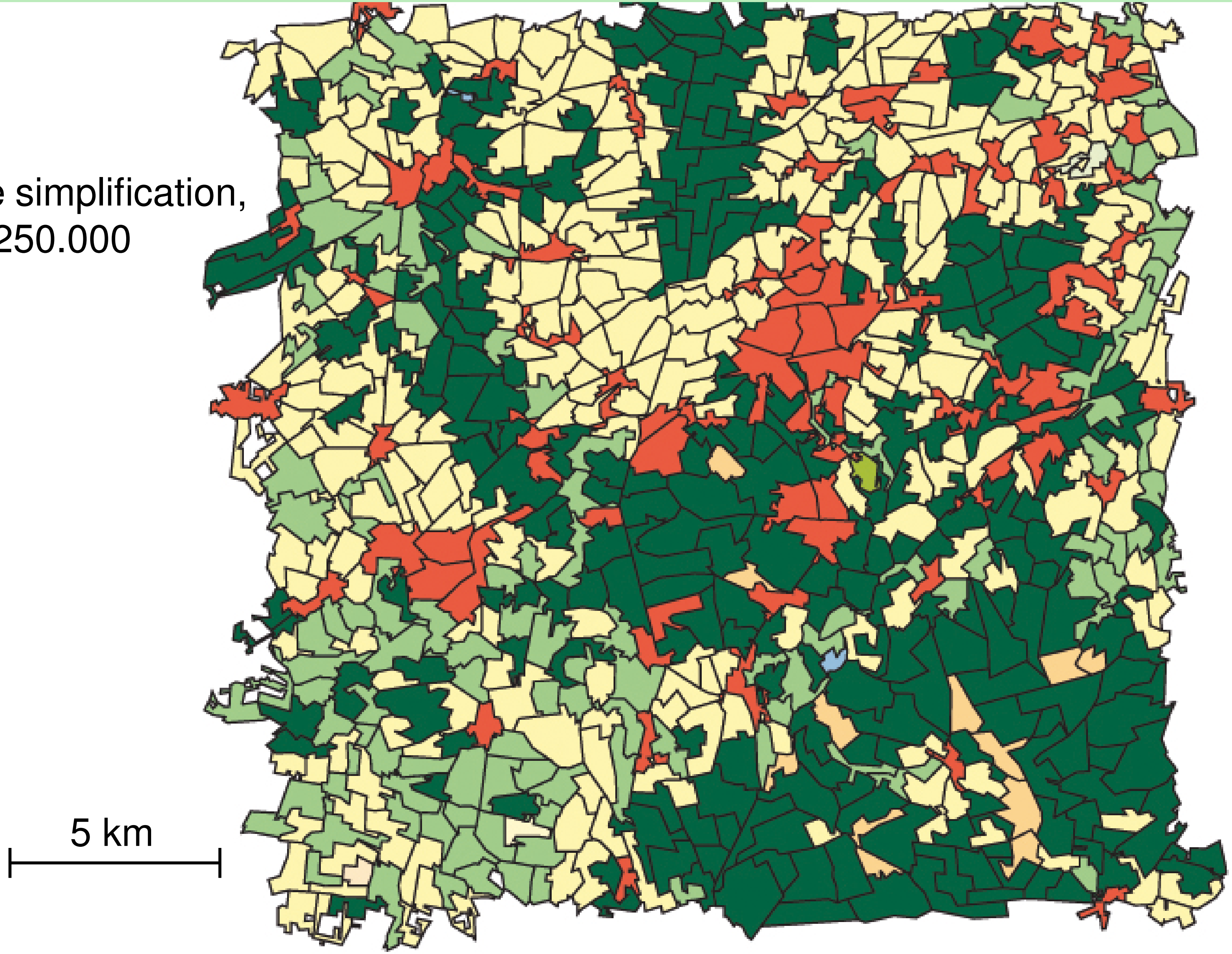
Comparison with iterative method:

- -20 % costs for class change
- -2 % costs for non-compact shapes
- -8 % total costs



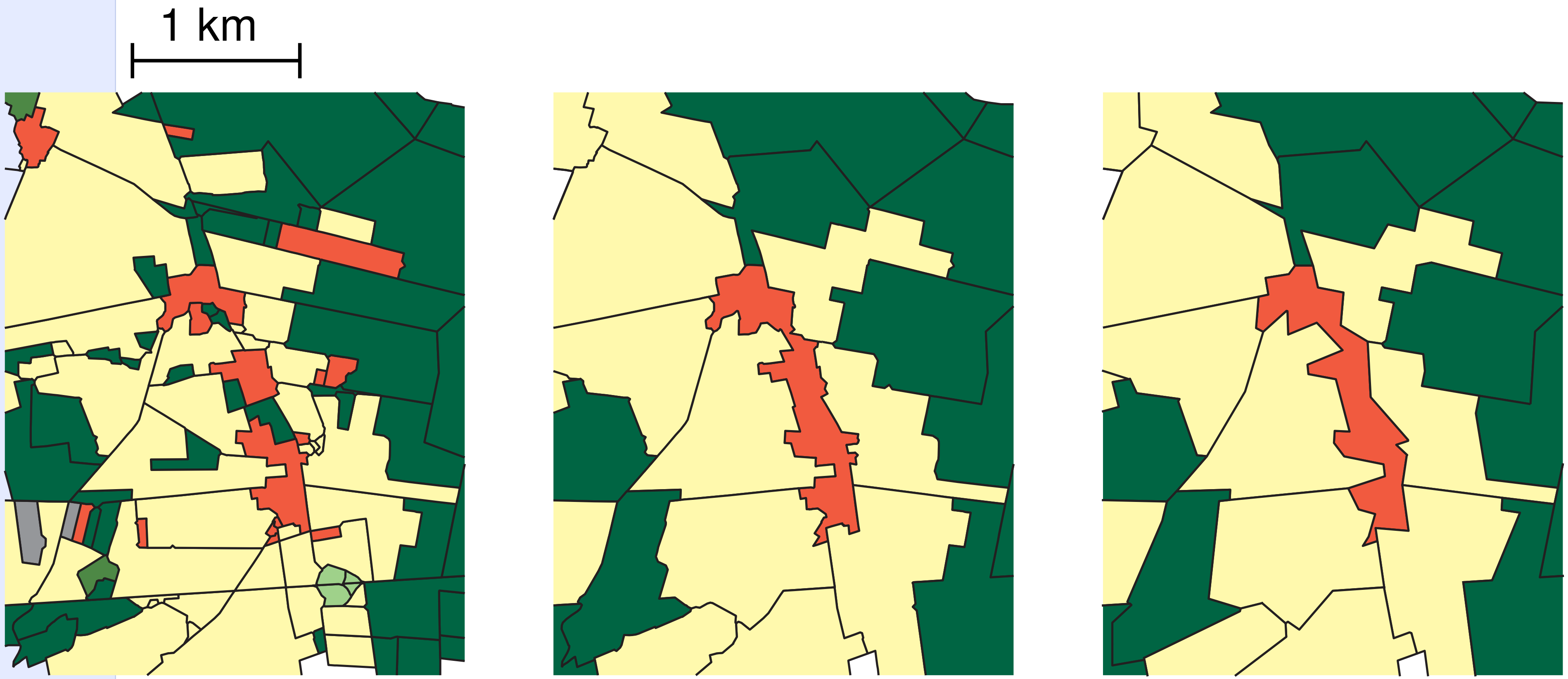
6. Results

after line simplification,
scale 1:250.000



6. Results

Details



ATKIS DLM50
Buchholz in der
Nordheide

after aggregation
scale 1:250.000

after line simplification
scale 1:250.000

6. Results

Details

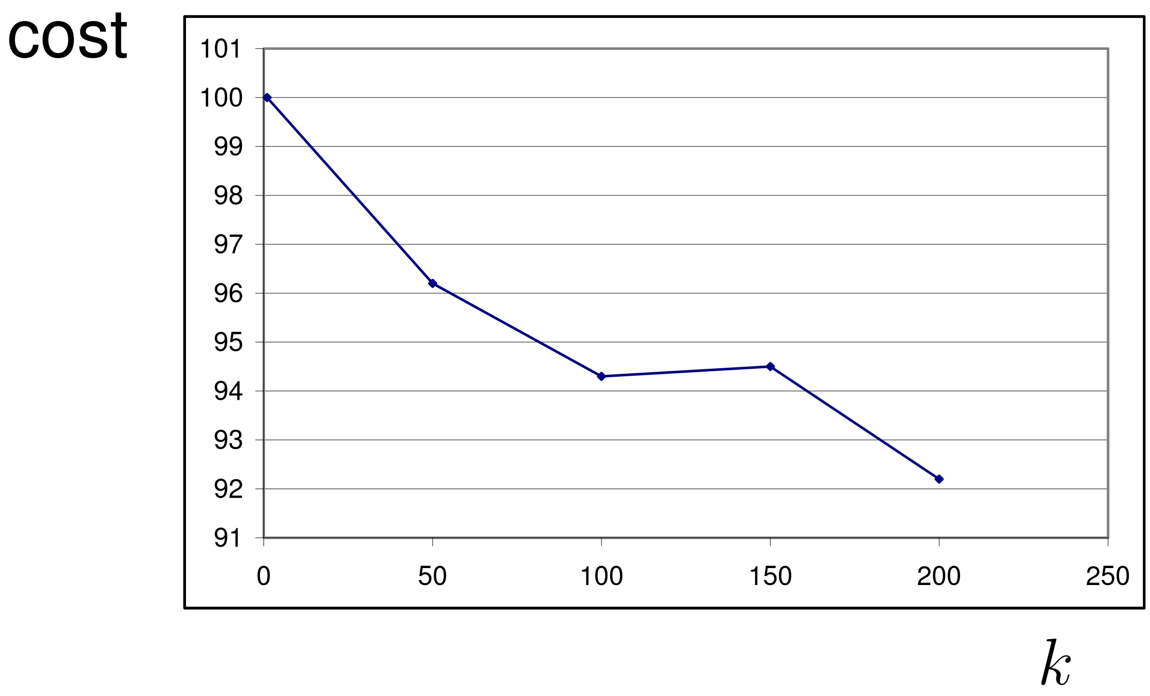
ATKIS DLM50 Buchholz in der Nordheide

after aggregation scale 1:250.000

after line simplification scale 1:250.000

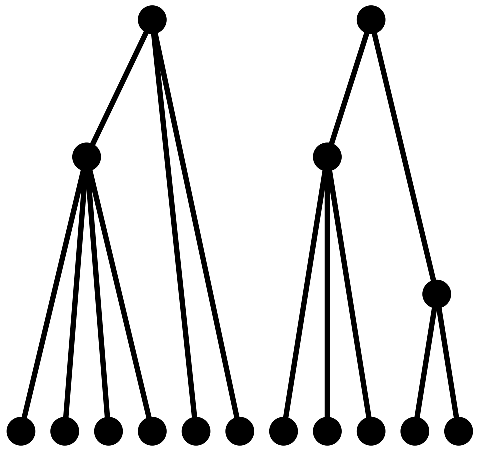
6. Results

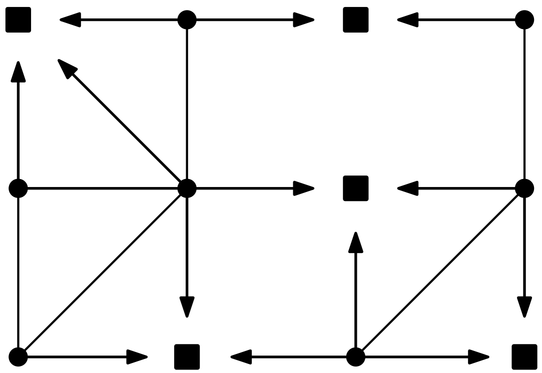
Results for different values of k:



6. Results

Other benefits:





- Intermediate scales can be used for continuous generalisation
- Further scales need to be interpolated

- Decomposition of problem is useful for incremental updating
- Decomposition allows to process in parallel

6. Results

Conclusion:

- A new efficient heuristic for area aggregation:
  - large areas are fixed as centres
  - intermediate scales are introduced
- In terms of class change, the method results in significantly better results than the purely iterative method (-20% cost).
- In terms of compactness only marginal improvements were made.
- The method generalises an existing algorithm for the same problem.
- Decomposition also allows incremental update.
- Intermediate scales can be exploited for continuous generalization.

Moscow, 3.8.2007

EFFICIENT AREA AGGREGATION BY COMBINATION OF DIFFERENT TECHNIQUES

Jan-Henrik Haunert

Institut für Kartographie und Geoinformatik
Leibniz Universität Hannover