Partitioning of Public Transit Networks

[Bachelor's thesis]

Matthias Hertel

Albert-Ludwigs-Universität Freiburg

11.09.2015

Contents

1. Introduction
   - Motivation
   - Goal
2. Data model
3. Algorithms
   - K-means
   - Merging algorithm
   - METIS
   - PUNCH
4. Evaluation

Transfer Patterns [1] with partitioning

Transfer Patterns = sequences of transfers on optimal routes
Freiburg $\rightarrow$ Zürich: {[Freiburg, Zürich], [Freiburg, Basel, Zürich]}

Transfer Patterns [1] with partitioning

Transfer Patterns = sequences of transfers on optimal routes
Freiburg → Zürich: {[Freiburg, Zürich], [Freiburg, Basel, Zürich]}

Compute Transfer Patterns between

- stations of the same partition
- border stations $b(C_x)$ and $b(C_y)$

⇒ reduced runtime
⇒ reduced space

Transfer Patterns [1] with partitioning

Transfer Patterns = sequences of transfers on optimal routes
Freiburg → Zürich: {[Freiburg, Zürich], [Freiburg, Basel, Zürich]}

Compute Transfer Patterns between

- stations of the same partition
- border stations $b(C_x)$ and $b(C_y)$

⇒ reduced runtime
⇒ reduced space

Query "A → B":
A → $b(C_A)$ → $b(C_B)$ → B
⇒ little slower query times

Goal

Partition the stations of a public transit network, such that

- partitions are small
- most traffic lies inside the partitions

Dataset

- schedule of Deutsche Bahn (2015)
- only local traffic (no ICEs and ICs)
- modelled as undirected weighted graph
- stations $\rightarrow$ nodes
- connections $\rightarrow$ edges
- frequencies $\rightarrow$ edge weights
- heuristical footpaths (distance $\leq$ 400 m; weight 200,000)

K-means-clustering [2]

- uses only geographic data

Algorithm 1 k-means-clustering

initialize
while assignments change **do**
  update assignments
  update means
end while

Merging algorithm [3]

- hierarchical
- merges neighboured partitions
- hyperparameter k = number of partitions
- hyperparameter U = upper bound partition size
- order distinguished by a utility function

Merging algorithm [3]

- hierarchical
- merges neighboured partitions
- hyperparameter k = number of partitions
- hyperparameter U = upper bound partition size
- order distinguished by a utility function

$$f(u,v) = \frac{1}{s(u) \cdot s(v)} \cdot \left(\frac{w(u,v)}{\sqrt{s(u)}} + \frac{w(u,v)}{\sqrt{s(v)}}\right)$$

$s(u)$ = size of u
$s(v)$ = size of v
$w(u,v)$ = sum of edge weights between u and v

METIS [4]

- graph partitioning framework
- state of the art
- can be downloaded [1]
- hyperparameter k = number of partitions
- three phases (next slide)

[1] http://glaros.dtc.umn.edu/gkhome/metis/metis/download

METIS

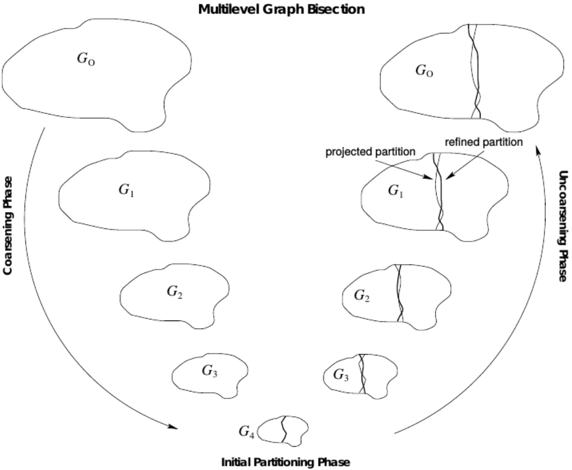


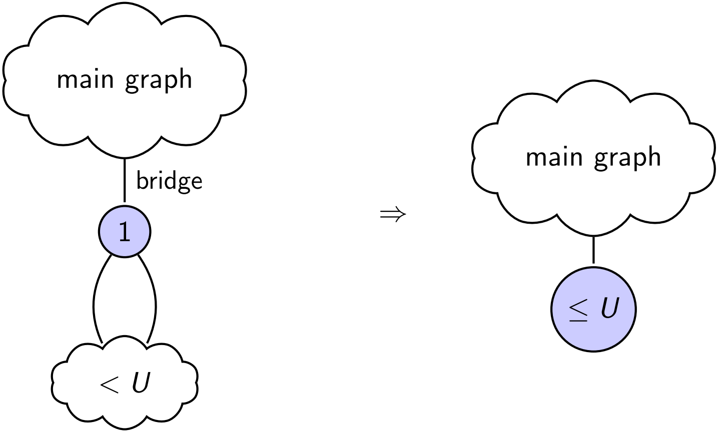
Figure : The three phases of METIS (Source: [4])

PUNCH [5]

- “partitioning using natural cut heuristics”
- hyperparameter $U =$ upper bound partition size
- two phases
  - filtering phase
  - assembly phase

PUNCH

Filtering phase: contract regions that are separated by small cuts



PUNCH

Assembly phase

- initial solution: run merging algorithm on filtered graph
- local optimization:
  - uncontract small regions
  - rerun merging algorithm
  - take better solution

Comparison: cut size

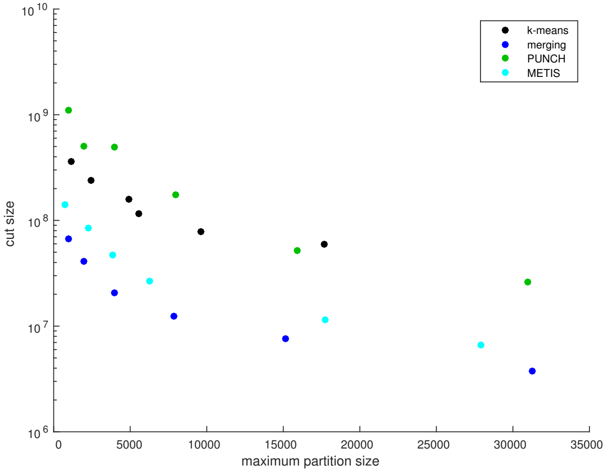


Figure : Cut size over maximum partition size.

Comparison: cut edges

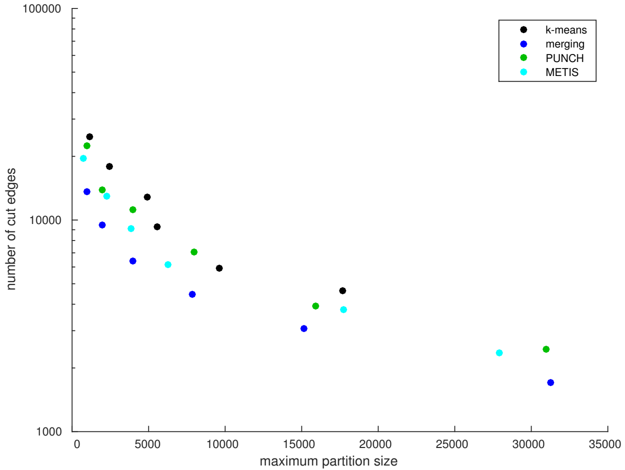
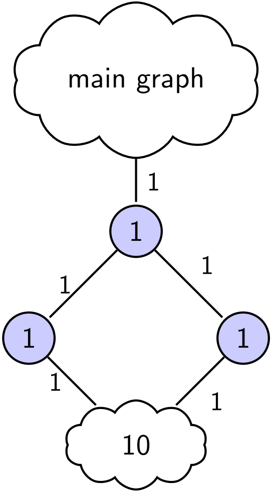
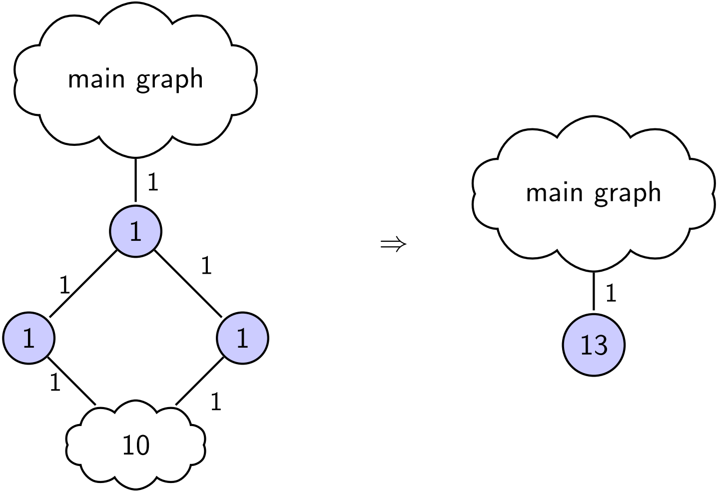


Figure : Cut edges over maximum partition size.

PUNCH - unweighted graph

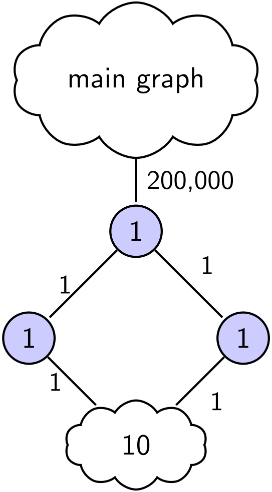


PUNCH - unweighted graph

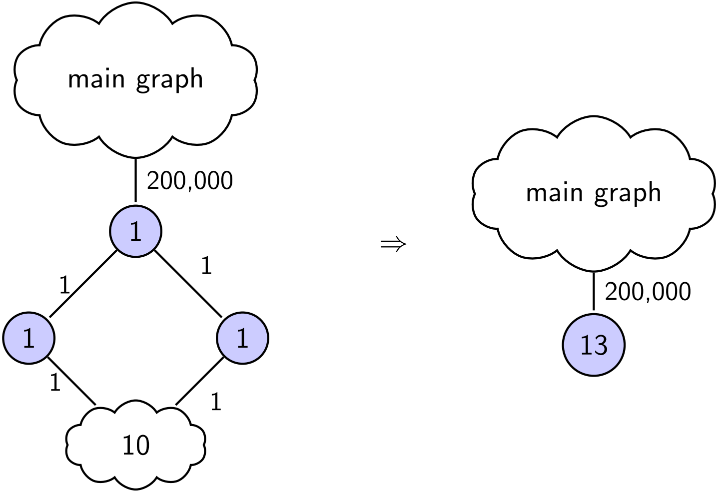


⇒ minimum cut preserved

PUNCH - weighted graph



PUNCH - weighted graph



⇒ minimum cut **not** preserved

The gain of footpaths

merging algorithm with U=4,000

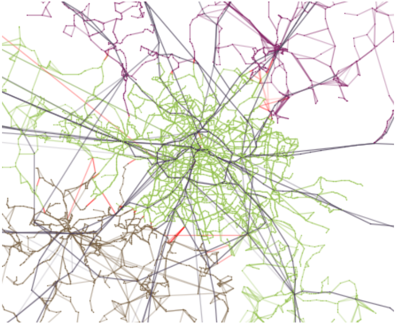


Figure : no footpaths

The gain of footpaths

merging algorithm with U=4,000

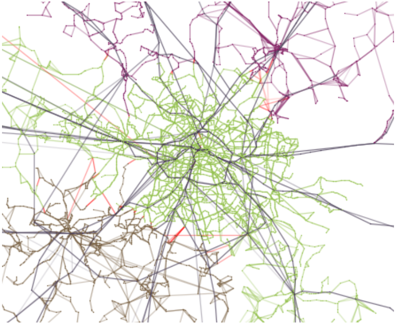


Figure : no footpaths

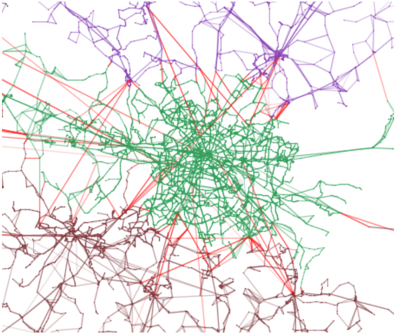


Figure : with footpaths

Conclusions

- K-means better than expected $\Rightarrow$ traffic geographically clustered

Conclusions

- K-means better than expected $\Rightarrow$ traffic geographically clustered
- merging algorithm and METIS produce good results

Conclusions

- K-means better than expected $\Rightarrow$ traffic geographically clustered
- merging algorithm and METIS produce good results
- arbitrary utility functions can be used with the merging algorithm

Conclusions

- K-means better than expected $\Rightarrow$ traffic geographically clustered
- merging algorithm and METIS produce good results
- arbitrary utility functions can be used with the merging algorithm
- PUNCH: filtering phase must use edge weights

Conclusions

- K-means better than expected $\Rightarrow$ traffic geographically clustered
- merging algorithm and METIS produce good results
- arbitrary utility functions can be used with the merging algorithm
- PUNCH: filtering phase must use edge weights
- footpaths prohibit geographically overlapping partitions

Questions?

Thank you for your attention!

Bibliography

H. Bast, E. Carlsson, A. Eigenwillig, R. Geisberger, C. Harrelson, V. Raychev, and F. Viger, "Fast routing in very large public transportation networks using transfer patterns," in *Algorithms–ESA 2010*. Springer, 2010, pp. 290–301.

J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

M. G. van der Horst, "Optimal route planning for car navigation systems," Master's thesis, Technische Universitaet Eindhoven, 2003.

G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

D. Delling, A. V. Goldberg, I. Razenshteyn, and R. F. Werneck, "Graph partitioning with natural cuts," in *Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International*. IEEE, 2011, pp. 1135–1146.

Cut size with k-means

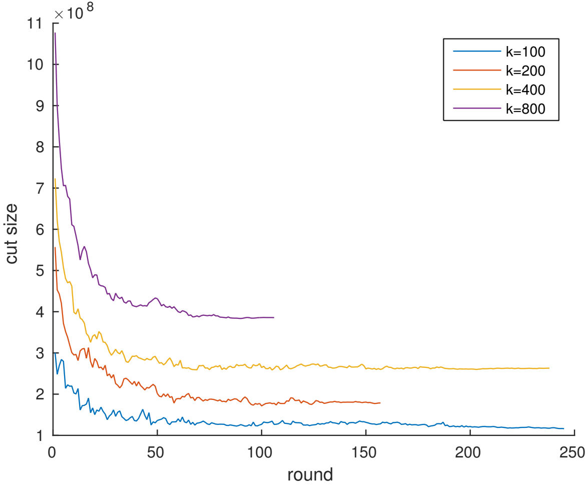


Figure : Cut size over maximum partition size.

METIS

unbalancing ratio r

$$s(p) \leq r \cdot \frac{N}{k}$$

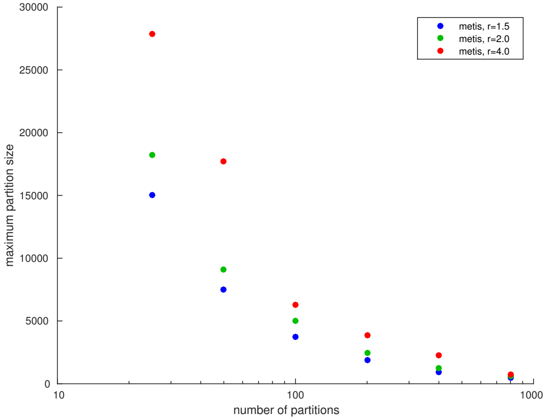


Figure : Maximum partition size over number of partitions.

METIS

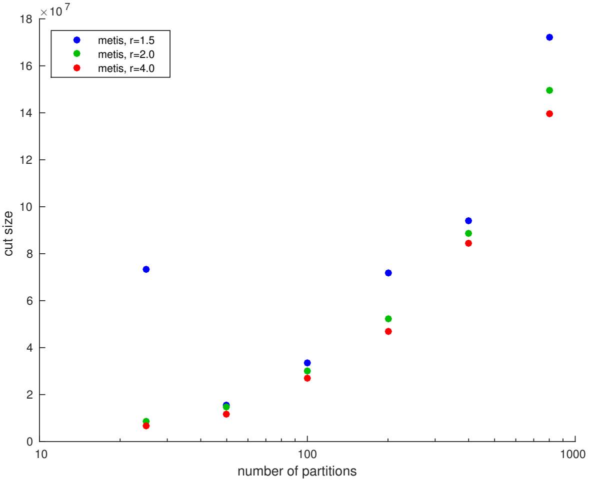
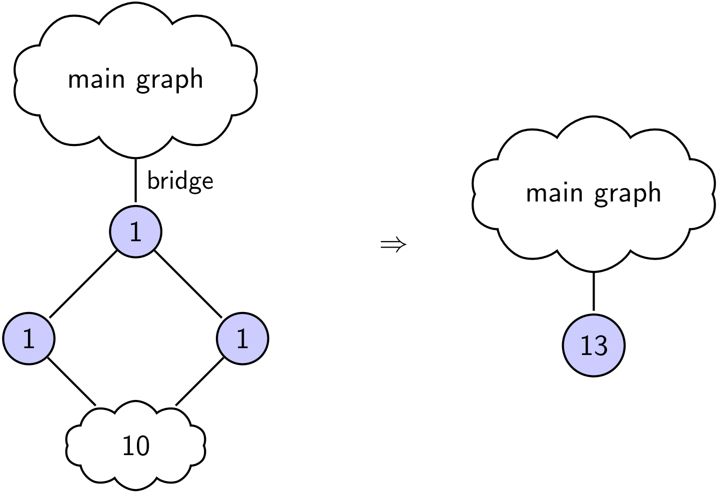


Figure : Cut size over number of partitions.

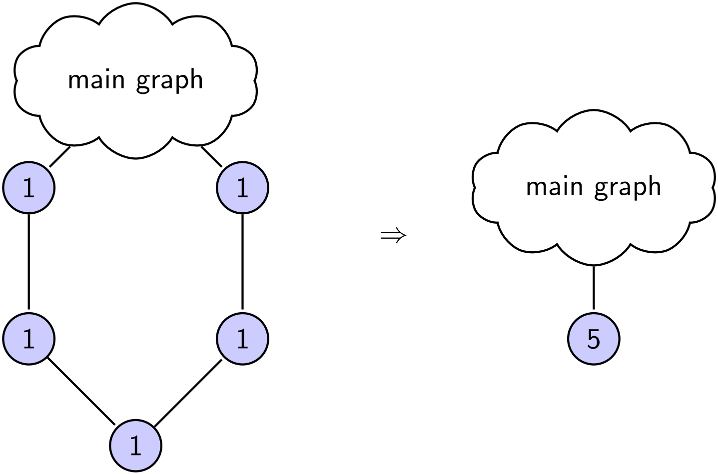
PUNCH

Filtering phase, pass 1: contract bridge-separated regions



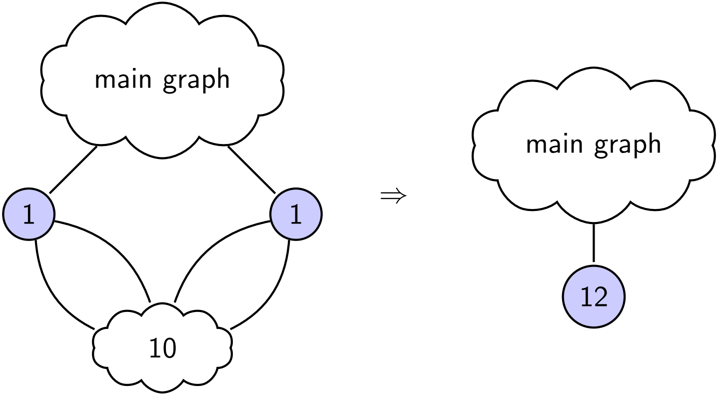
PUNCH

Filtering phase, pass 2: contract simple paths



PUNCH

Filtering phase, pass 3: contract two-cut-separated regions



PUNCH

Filtering phase, pass 4: contract "natural cut"-separated regions

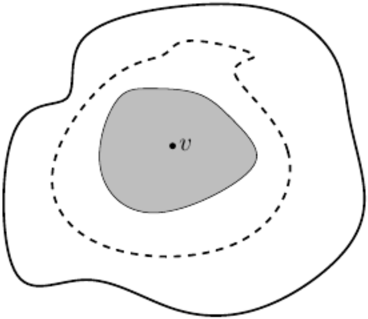


Figure : Finding a "natural cut" (Source: [5])

	k-means	merging	PUNCH	METIS
partitions	181	181	176	181
max. part. size	4,015	**1,873**	1,975	3,132
cut size	$154.7 \cdot 10^6$	$\mathbf{42.8 \cdot 10^6}$	$496.4 \cdot 10^6$	$45.5 \cdot 10^6$
cut edges	12,273	9,497	13,917	**8,562**
cut edges (%)	2.2	1.7	2.5	**1.6**
border nodes	15,564	12,954	17,669	**12,010**
border nodes (%)	6.2	5.2	7.1	**4.8**
runtime (s)	53.8	2.9	118.3	**0.3**

Table : Results of the four algorithms with about 181 partitions.