Karlsruhe Institute of Technology

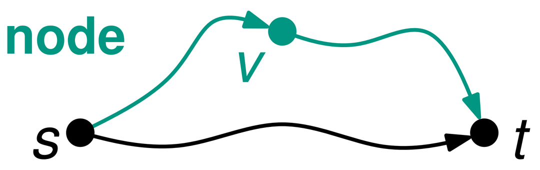
Institute of Theoretical Informatics
Algorithmics

Candidate Sets for Alternative Routes in Road Networks [1]

Dennis Luxen and Dennis Schieferdecker – {*luxen,schieferdecker@kit.edu*}
http://algo2.iti.kit.edu

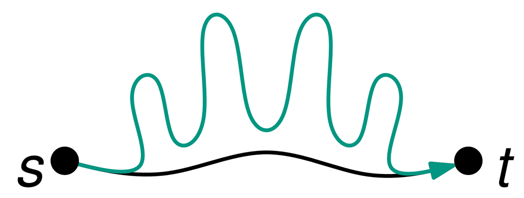
Modelling Alternatives

- concatenation of two shortest paths at **via node**
- must adhere to **quality criteria**



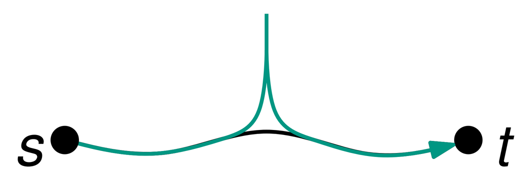
Quality Criteria

- **uniformly bounded stretch**
  each subpath should not be too much longer than a shortest path [$\epsilon = 25\%$]



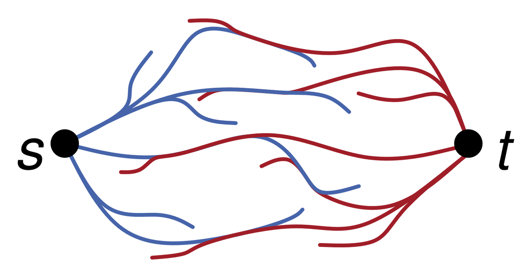
- **maximum overlap**
  paths should not have too many subpaths in common [$\gamma = 80\%$]

- **local optimality**
  all local decisions along a path should make sense [$\alpha = 25\%$]



Baseline Algorithm [2]

- grow search spaces from s and t
- **plateaus** yield candidates for via nodes



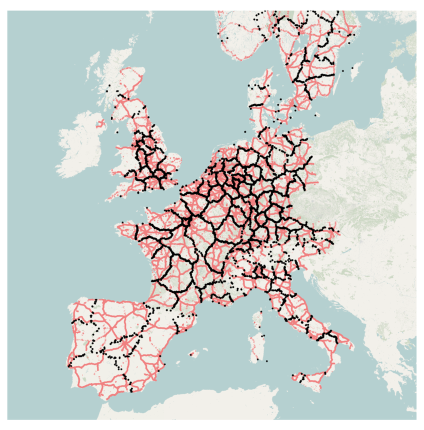
Algorithm is named according to underlying shortest path technique, e.g. **X-BDV**, **X-CHV**, **X-CHASEV**, . . .

Conjecture (limited number of alternative paths)

If the number of shortest paths between two regions of a road network is small, so is the number of plateaus. Likewise, the number of admissible alternatives is small and can be covered by few via nodes.

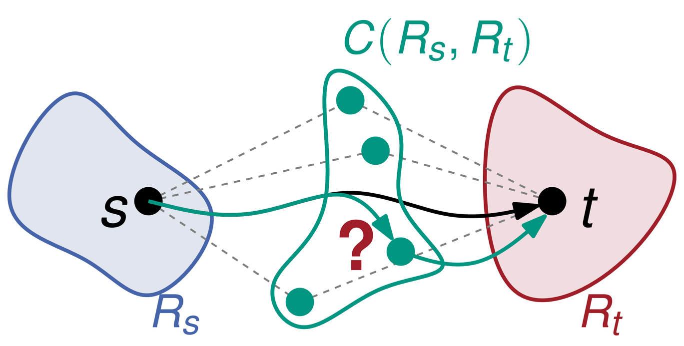
Idea (Single-Level)

- partition graph into regions
- compute **via node candidate set** for each pair of regions
- examine sets during query



Query

- determine via node candidate set for the considered region pair
- check via node candiates in set
- stop when quality criteria are fulfilled

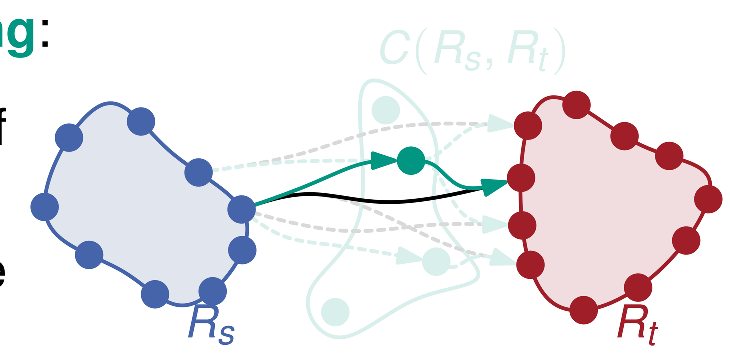


If regions are neighboring or the same, baseline algorithm is used.

Preprocessing

Our algorithm is used for **bootstrapping**:

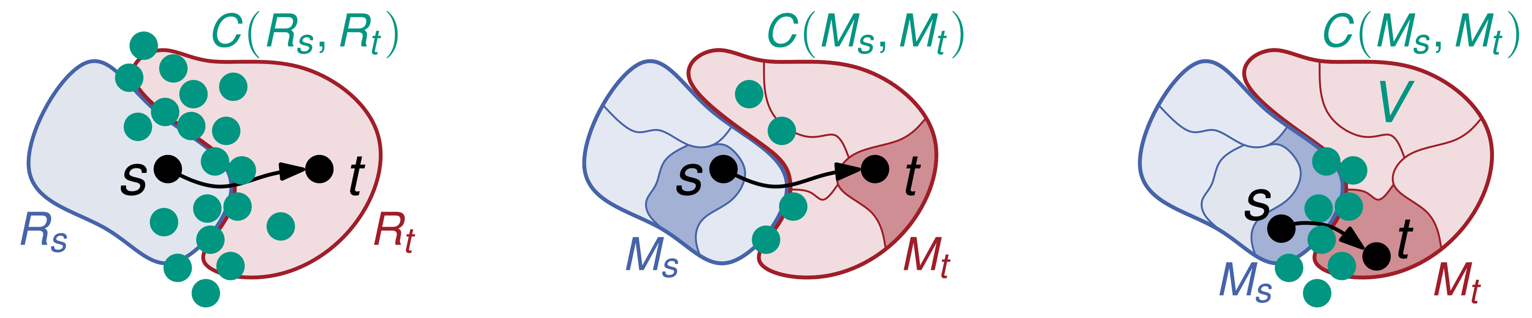
- compute alternatives for all pairs of boundary nodes for all region pairs
- if no alternative is found, run baseline algorithm to compute new via node
- add node to respective via node candidate set



Engineering: parallelization, sampling, storing search spaces, . . .

Extension (Multi-Level)

Partitioning can be done in multiple levels. If source and target regions are neighboring or the same, the algorithm recurses to a finer level.

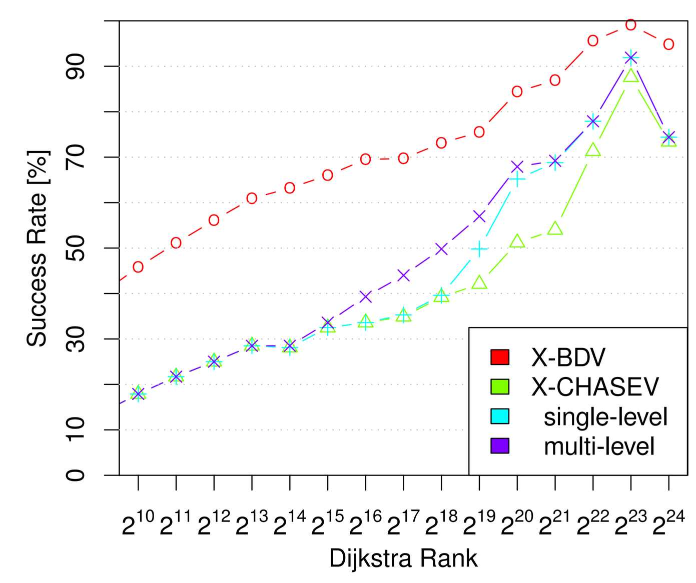


Experimental Evaluation

Algorithms are implemented in C++ and compiled with g++ 4.5 using full optimizations. Queries use a Core i7-920 at 2.66 GHz (12 GiB). Preprocessing uses 4 Opteron 6168 at 1.90 Ghz (256 GiB). Experiments are done on the road network of Western Europe, as provided by PTV AG for the 9th DIMACS Implementation Challenge.

no relaxation

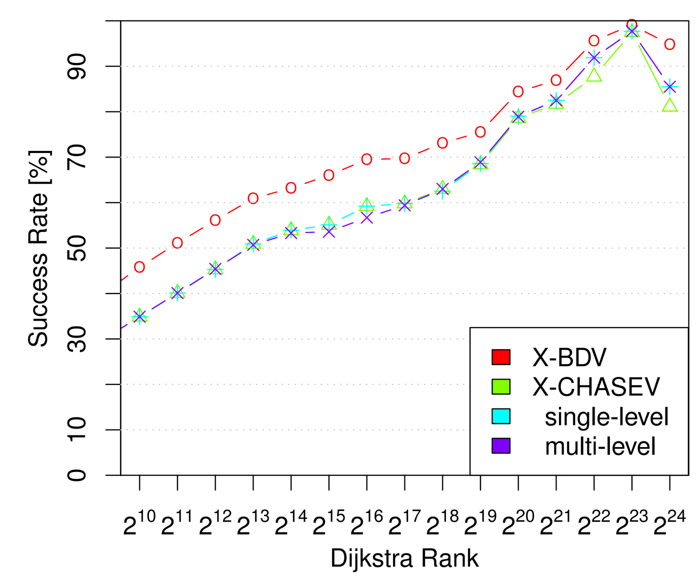
query	p=1 time [ms]	p=1 success rate [%]	p=2 time [ms]	p=2 success rate [%]
X-BDV	11.5 s	94.51	12.3 s	80.60
X-CHV	1.218	75.56	1.771	40.25
X-CHASEV	0.581	75.56	0.797	40.25
single-level	0.167	80.73	0.304	50.87
multi-level	0.162	81.20	0.304	51.25



preprocessing	time [h]	size [kiB]	average size p=1	average size p=2
single-level	1.11	858.42	4.44	5.17
+ multi-level	0.66	2 809.72	6.12	5.92

3-relaxtion

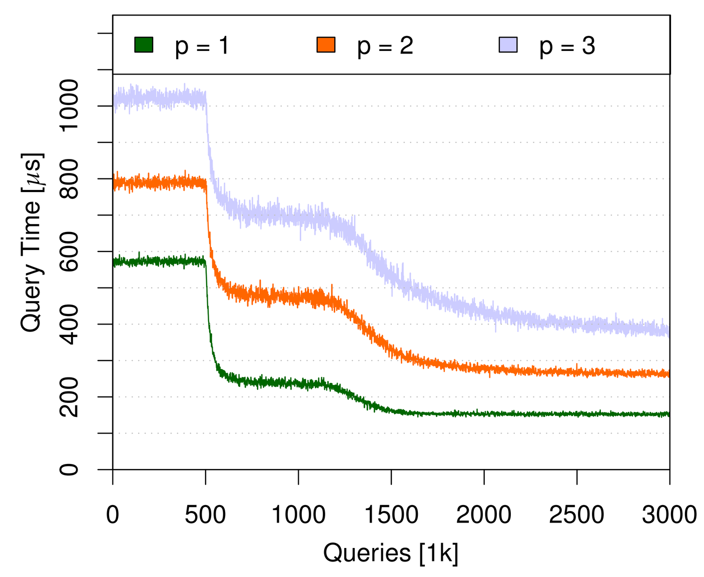
query	p=1 time [ms]	p=1 success rate [%]	p=2 time [ms]	p=2 success rate [%]
X-BDV	11.5 s	94.51	12.3 s	80.60
X-CHV	3.488	88.59	4.382	64.75
X-CHASEV	2.756	88.59	3.258	64.75
single-level	0.254	90.05	0.438	70.22
multi-level	0.188	90.06	0.386	70.40

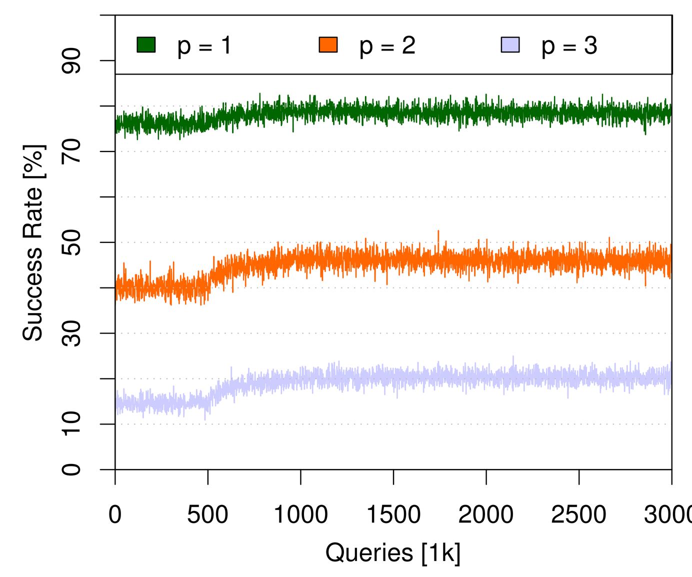


preprocessing	time [h]	size [kiB]	average size p=1	average size p=2
single-level	2.38	1 741.77	6.74	10.26
+ multi-level	1.96	7 166.44	12.20	15.09

Online Algorithm

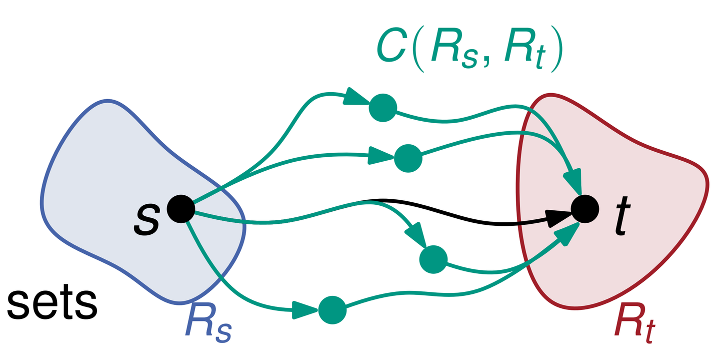
Can be added on top of a legacy system. Via node candidate sets start empty. If our algorithm does not yield an alternative, the baseline algorithm is applied as **fallback** and the found via node added to the appropriate set. Fallbacks are stopped after sets become saturated.





Alternative Graphs

- summarize multiple alternatives
- provide a sparse set of options
- computable from via node candidate sets



Bibliography

[1]Luxen, Schieferdecker. 2012. *Candidate Sets for Alternative Routes in Road Networks.* (SEA'12)

[2]Abraham, Delling, Goldberg, Werneck. 2013. *Alternative Routes in Road Networks.* (JEA #18)