Karlsruhe Institute of Technology

Efficient Route Planning with Temporary Driving Restrictions

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · **Tim Zeitz** | June 17th, 2020

INSTITUTE OF THEORETICAL INFORMATICS

Scenario


12m
Samstag 7:00-12:00
vom 15.12. bis 30.4.


22-6h
Lärmschutz

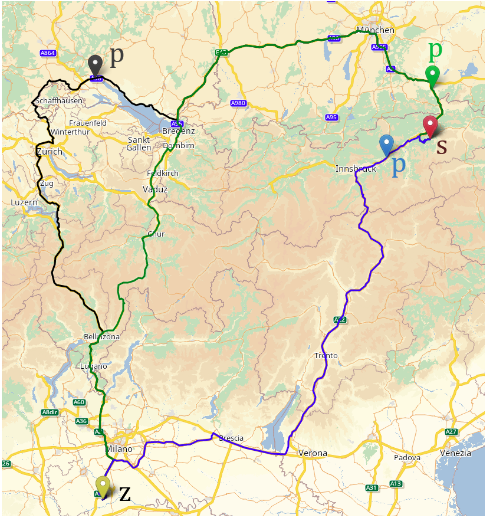

Feiertags
uneinheitlich

Scenario


HÜER
DUVENBECK


ESSO
J. CUENCA
TransWest
KOCIUK LOGISTICS
ISO 9001

Scenario



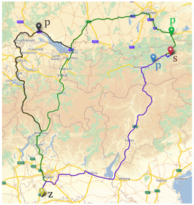
p
p
s
p
z
München
Schaffhausen
Frauenfeld
Winterthur
Sankt Gallen
Bregenz
Dornbirn
Zürich
Zug
Luzern
Feldkirch
Vaduz
Chur
Innsbruck
Bellinzona
Lugano
Milano
Brescia
Verona
Trento
Padova
Venezia

Goals

1. Achieve practical performance
   - Avoid *NP*-hard problem formulation

2. Consider quality of parking locations

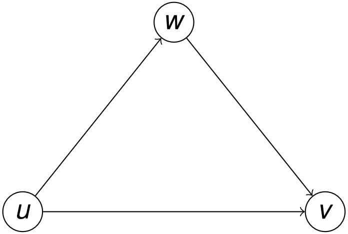
3. Consider trade-offs between earlier arrival and more comfortable routes



Model

Given:

- Graph $G = (V, E)$
- Travel times δ
- Temporary driving restrictions
- Parking locations with ratings
- Waiting costs w_i
- Driving costs d



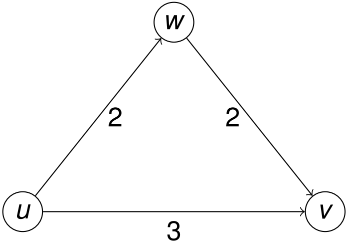
Problem

- Find Pareto-optimal routes between vertices s and z
  - regarding arrival time and abstract costs
- Waiting at a node causes costs depending on the rating
- Waiting at unrated locations is allowed, driving also has a cost

Model

Given:

- Graph $G = (V, E)$
- Travel times δ
- Temporary driving restrictions
- Parking locations with ratings
- Waiting costs w_i
- Driving costs d



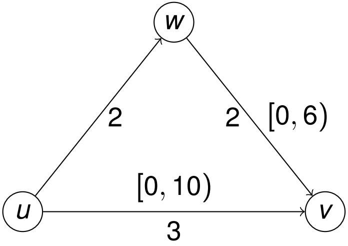
Problem

- Find Pareto-optimal routes between vertices s and z
  - regarding arrival time and abstract costs
- Waiting at a node causes costs depending on the rating
- Waiting at unrated locations is allowed, driving also has a cost

Model

Given:

- Graph $G = (V, E)$
- Travel times δ
- Temporary driving restrictions
- Parking locations with ratings
- Waiting costs w_i
- Driving costs d



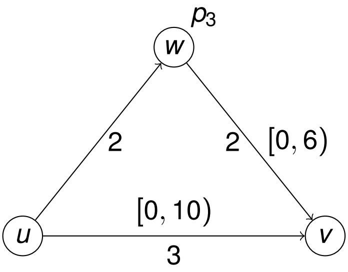
Problem

- Find Pareto-optimal routes between vertices s and z
  - regarding arrival time and abstract costs
- Waiting at a node causes costs depending on the rating
- Waiting at unrated locations is allowed, driving also has a cost

Model

Given:

- Graph $G = (V, E)$
- Travel times δ
- Temporary driving restrictions
- Parking locations with ratings
- Waiting costs w_i
- Driving costs d



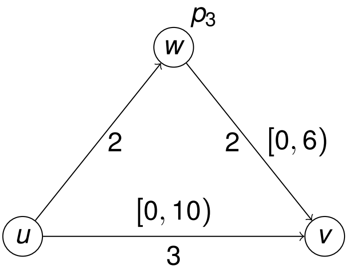
Problem

- Find Pareto-optimal routes between vertices s and z
  - regarding arrival time and abstract costs
- Waiting at a node causes costs depending on the rating
- Waiting at unrated locations is allowed, driving also has a cost

Model

Given:

- Graph $G = (V, E)$
- Travel times δ
- Temporary driving restrictions
- Parking locations with ratings
- Waiting costs w_i
- Driving costs d



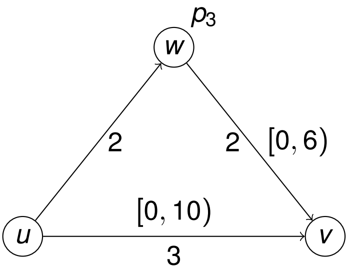
Problem

- Find Pareto-optimal routes between vertices s and z
  - regarding arrival time and abstract costs
- Waiting at a node causes costs depending on the rating
- Waiting at unrated locations is allowed, driving also has a cost

Model

Given:

- Graph $G = (V, E)$
- Travel times δ
- Temporary driving restrictions
- Parking locations with ratings
- Waiting costs w_i
- Driving costs d



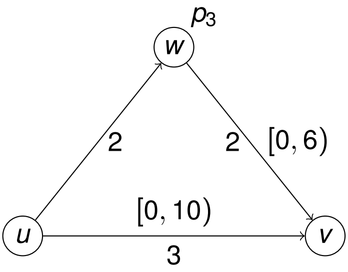
Problem

- Find Pareto-optimal routes between vertices s and z
  - regarding arrival time and abstract costs
- Waiting at a node causes costs depending on the rating
- Waiting at unrated locations is allowed, driving also has a cost

Model

Given:

- Graph $G = (V, E)$
- Travel times δ
- Temporary driving restrictions
- Parking locations with ratings
- Waiting costs w_i
- Driving costs d



Problem

- Find Pareto-optimal routes between vertices s and z
  - regarding arrival time and abstract costs
- Waiting at a node causes costs depending on the rating
- Waiting at unrated locations is allowed, driving also has a cost

Complexity

$d < w_0$

NP-hard by reduction from PARTITION

$d > w_0$

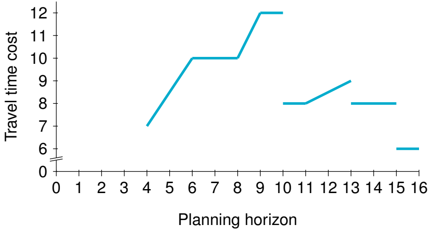
Possibly exponential number of Pareto-optimal routes

$d = w_0$

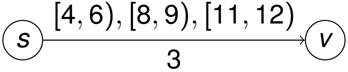
Feasible!

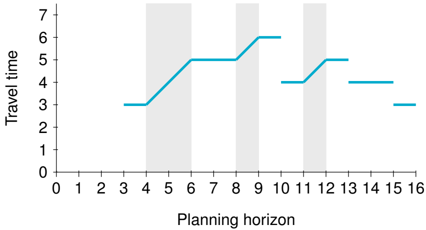
Algorithm

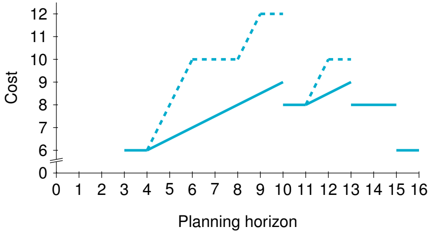
- Label-correcting Dijkstra
- Labels: Cost profiles
  - Tentative minimal costs
  - as function of arrival time
  - Piecewise linear functions
- Queue ordered by update time
- Polynomial running time
  - when $d = w_0$



Algorithm: Linking and Merging



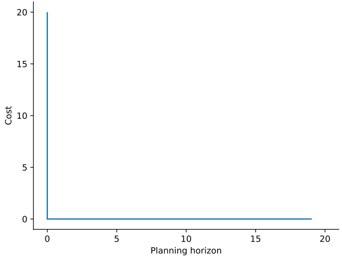


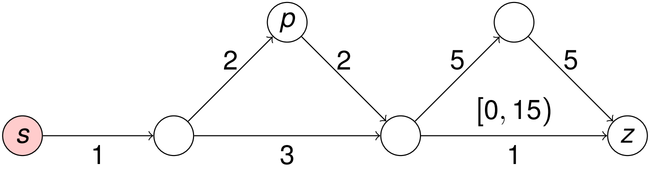


$$\mathcal{C}'_v(t) := \mathcal{C}_u(t - \mathcal{T}_e(t)) + d \cdot \delta(e) + w_0 \cdot (\mathcal{T}_e(t) - \delta(e)) \tag{1}$$

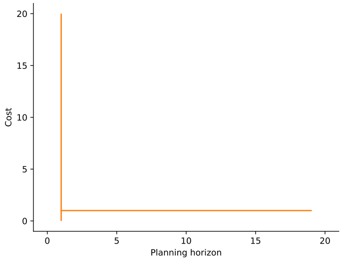
$$\mathcal{C}'_v(t) := \min\{\mathcal{C}'_v(t') + w_{\rho(v)} \cdot (t - t') \mid t^{min} \leq t' \leq t\} \tag{2}$$

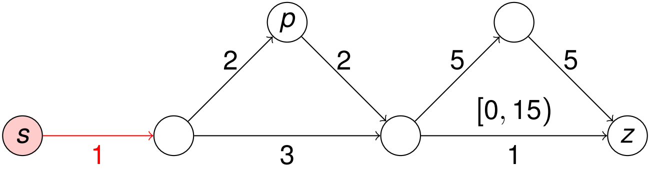
Algorithm: Example I



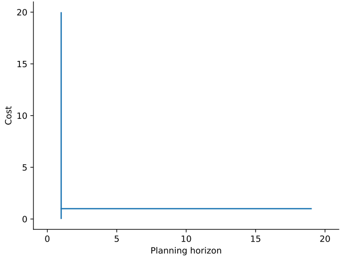


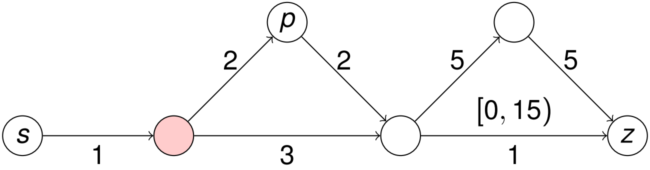
Algorithm: Example I



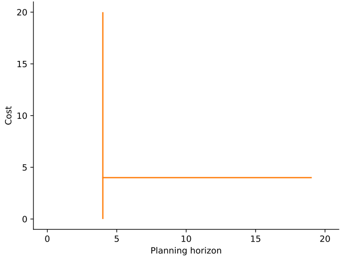


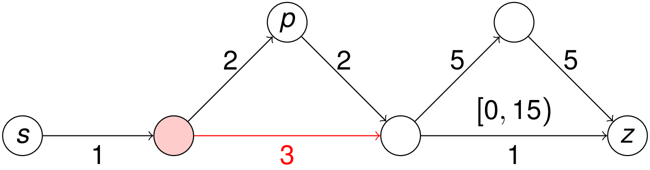
Algorithm: Example I



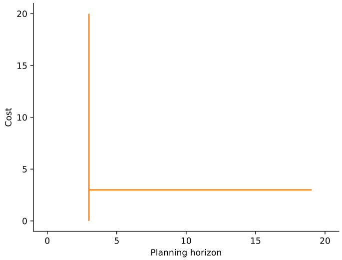


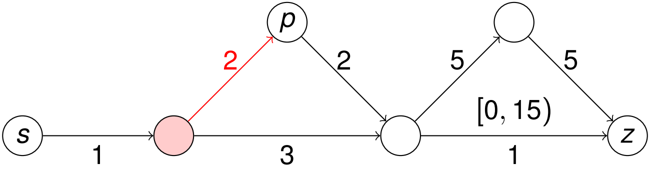
Algorithm: Example I



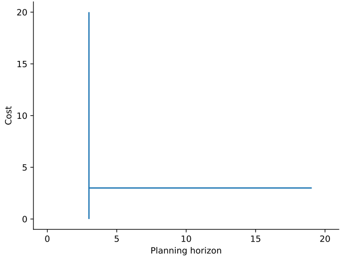


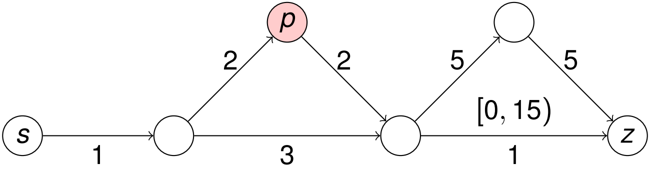
Algorithm: Example I





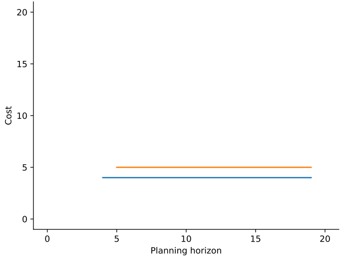
Algorithm: Example I

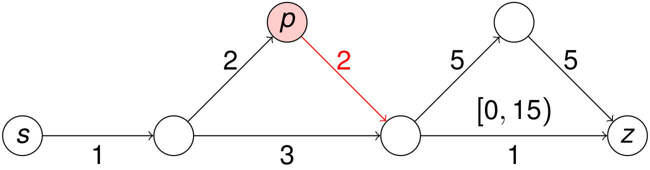




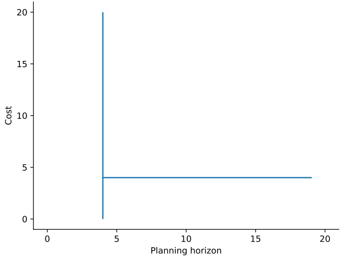
Algorithm: Example I

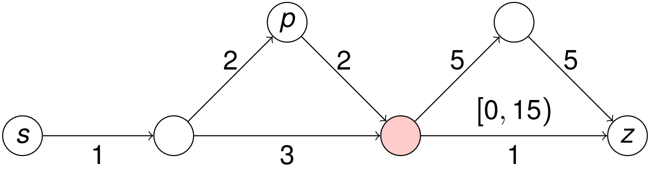
Karlsruhe Institute of Technology



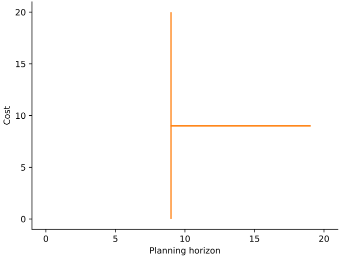


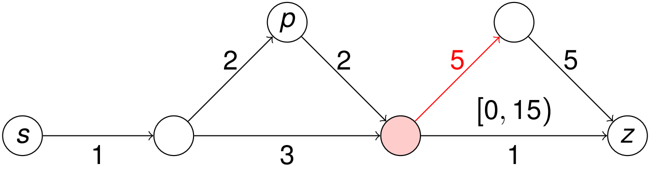
Algorithm: Example I



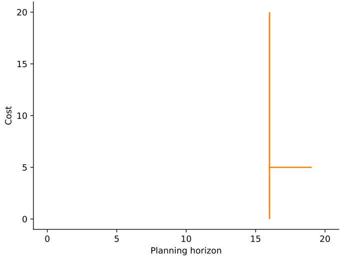


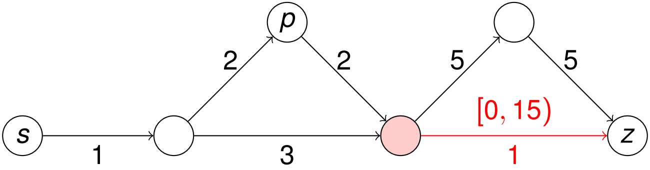
Algorithm: Example I



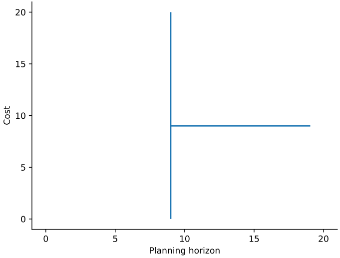


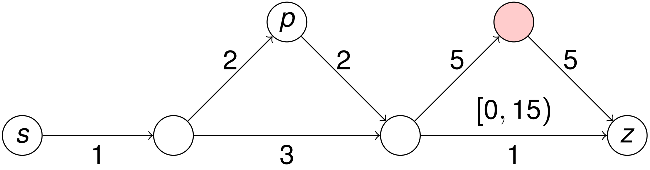
Algorithm: Example I



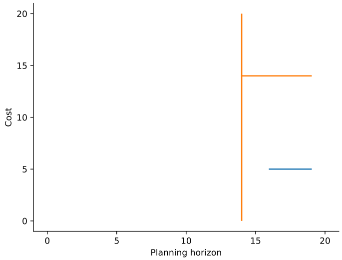


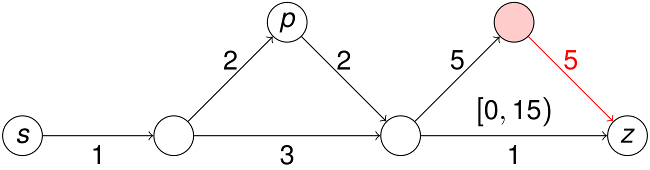
Algorithm: Example I



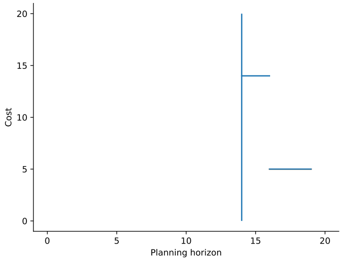


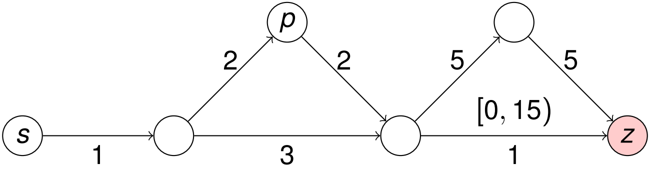
Algorithm: Example I



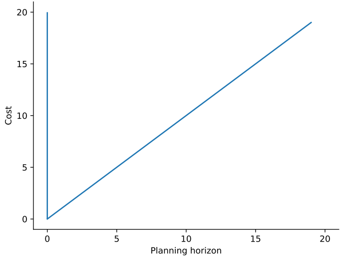


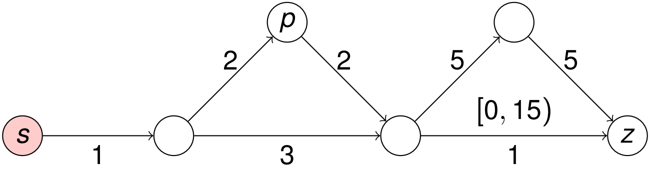
Algorithm: Example I



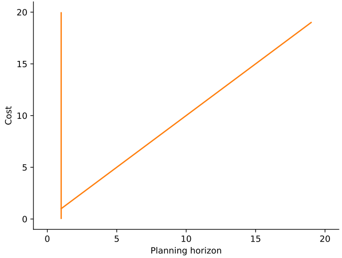


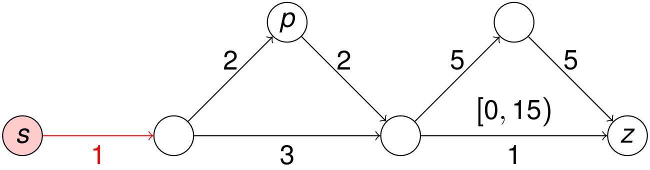
Algorithm: Example II



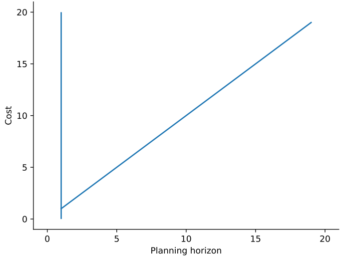


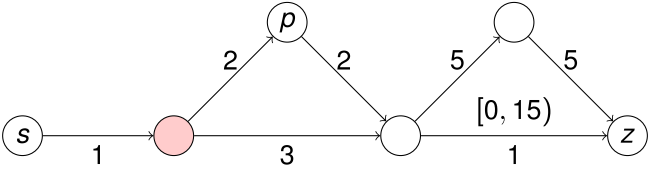
Algorithm: Example II



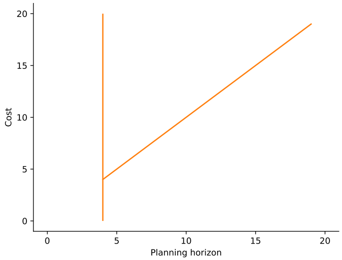


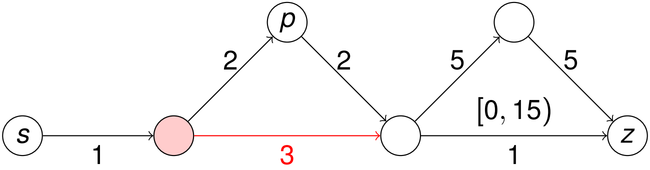
Algorithm: Example II



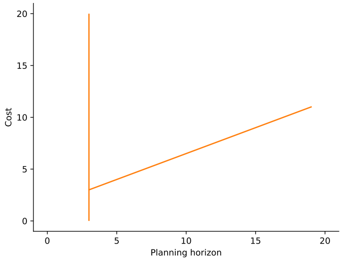


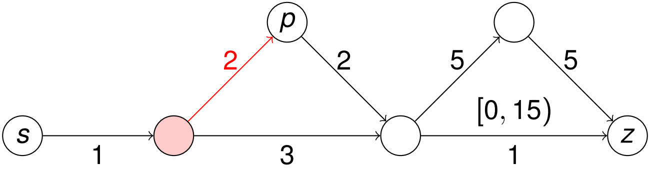
Algorithm: Example II



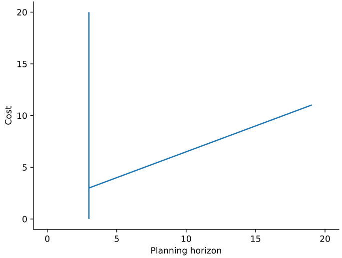


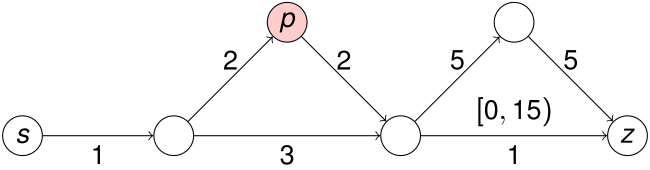
Algorithm: Example II



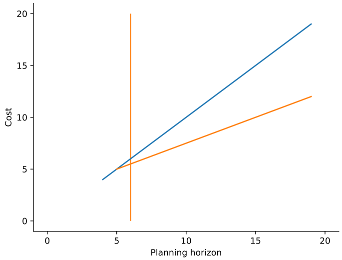


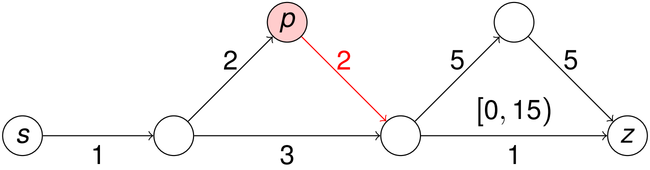
Algorithm: Example II



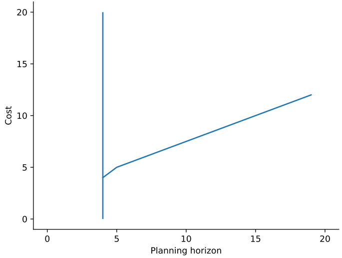


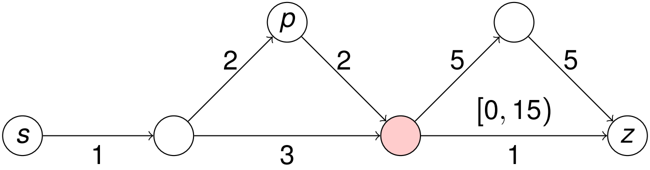
Algorithm: Example II



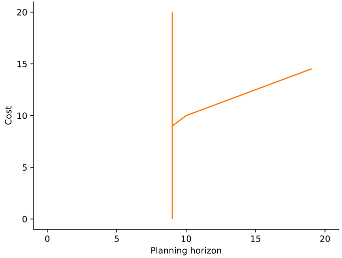


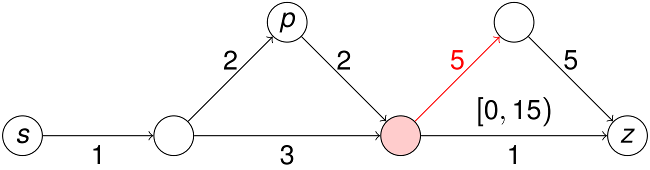
Algorithm: Example II



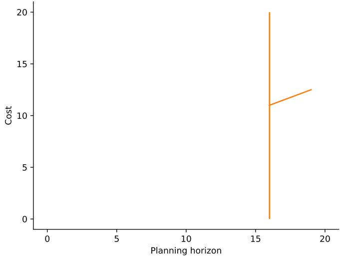


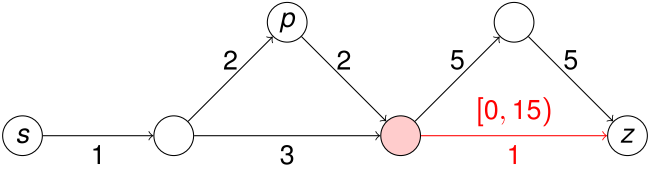
Algorithm: Example II



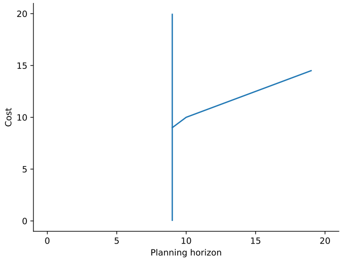


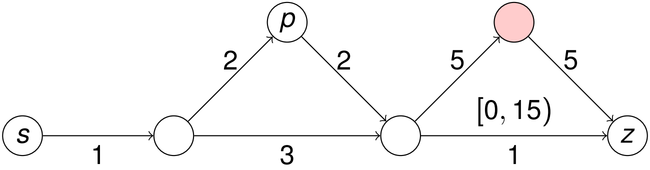
Algorithm: Example II



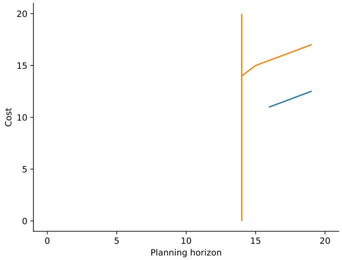


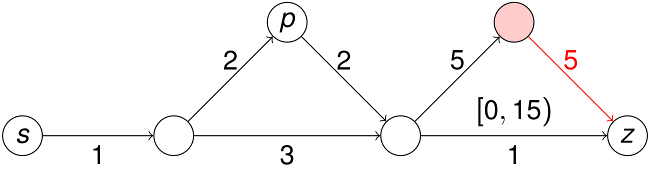
Algorithm: Example II



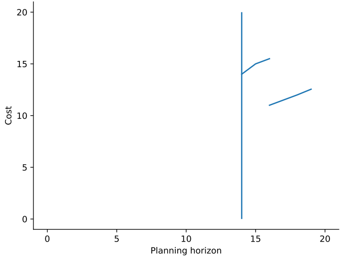


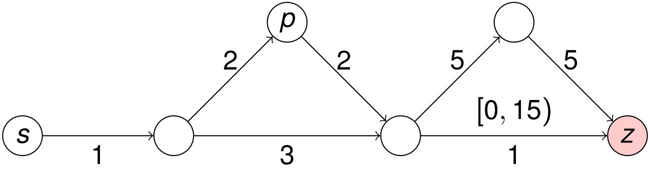
Algorithm: Example II





Algorithm: Example II





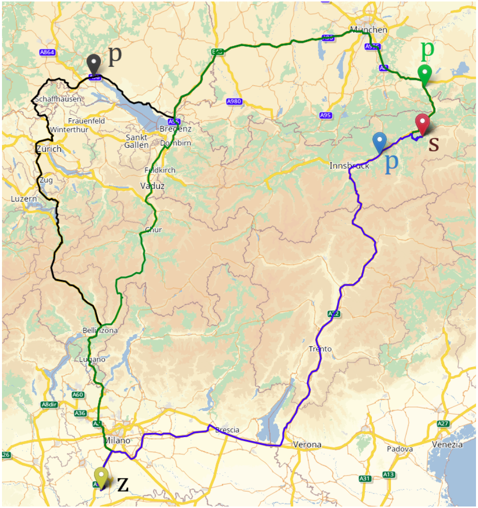
Implementation

- C++14
- Using RoutingKit
- Pruning...
  - with bounds
  - with the target profile
  - loops
- Goal directed search with A*
  - with CH-Potentials [SZ19]

Experiments

- Road network of central Europe
  - 21.9M vertices, 47.6M edges
  - Sunday and night driving bans and local road closures
  - 15317 parking location vertices
- Queries
  - Select random vertices from A and B
  - Make algorithm cope with night driving bans
- Machine
  - Intel i7-7600 CPU with 3.4 GHz
  - 32 GB DDR4 RAM

Trade-offs



16 h driving time,
2.5 min waiting time (rating 3)

13 h driving time,
4 h waiting time (rating 5)

10 h driving time,
8 h waiting time (rating 4)

Waiting

- Most of waiting scheduled at s
- At most one additional stop per route
  - Average 0.2
- Waiting at unrated locations
  - Happens
  - Always the quickest route
  - Alternative routes exist
- Results stable against different cost parameterizations
  - Availability + costs influences where most waiting happens

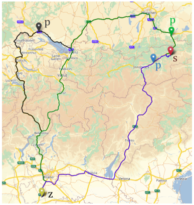
Performance

	Planning horizon	Optimal Routes [#]	Arrival time deviation [h:mm]	Running time Avg. [ms]	Running time Median [ms]
	Mon. 18:00, 1 day	2.86	2:17	529.4	266.3
	Mon. 18:00, 2 days	3.54	3:19	648.1	405.6
Random	Fri. 06:00, 1 day	1.04	0:10	10.0	0.6
Random	Fri. 06:00, 2 days	1.08	0:16	79.5	0.7
Random	Fri. 18:00, 1 day	1.13	0:08	205.8	0.6
Random	Fri. 18:00, 2 days	1.32	0:20	1 028.1	0.7

Conclusion

1. Achieve practical performance
2. Consider quality of parking locations
3. Consider trade-offs between earlier arrival and more comfortable routes

- Introduced problem formulation achieving these goals
  - Solvable in polynomial time for certain parametrizations
- Implementation
  - Reasonable routes on realistic instances
  - Average running times below 1 s
- Future work
  - Bidirectional search



Thank you!

References I

Ben Strasser and Tim Zeitz. *A* with Perfect Potentials*. 2019. arXiv: 1910.12526 [cs.DS].