Efficient Route Planning with Temporary Driving Restrictions

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz | June 17th, 2020
Scenario
Scenario
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 $\delta$
- 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 $\delta$
- 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 $\delta$
- 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 \( \delta \)
- 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 $\delta$
- 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 $\delta$
- 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 $\delta$
- 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

\[ S \rightarrow [4, 6), [8, 9), [11, 12) \rightarrow V \]

\[ C'_v(t) := C_u(t - T_e(t)) + d \cdot \delta(e) + w_0 \cdot (T_e(t) - \delta(e)) \]  

\[ C'_v(t) := \min \{ C'_v(t') + w_{\rho(v)} \cdot (t - t') \mid t_{\min} \leq t' \leq t \} \]
Algorithm: Example I

Introduction Model Algorithm Evaluation Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz –
Efficient Route Planning with Temporary Driving Restrictions
Algorithm: Example I

Introduction Model Algorithm Evaluation Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz – Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
Algorithm: Example I
Algorithm: Example I

Introduction
Model
Algorithm
Evaluation
Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz —
Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
Algorithm: Example I

Introduction
Model
Algorithm
Evaluation
Conclusion

Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
Algorithm: Example I

Introduction Model Algorithm Evaluation Conclusion
Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz –
Efficient Route Planning with Temporary Driving Restrictions
June 17th, 2020 10/18
Algorithm: Example I

Introduction

Model

Algorithm

Evaluation

Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz

Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020

10/18
Algorithm: Example I

Introduction

Model

Algorithm

Evaluation

Conclusion
Algorithm: Example I

Introduction Model Algorithm Evaluation Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz – Efficient Route Planning with Temporary Driving Restrictions
Algorithm: Example I
Algorithm: Example II
Algorithm: Example II

- Introduction
- Model
- Algorithm
- Evaluation
- Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz – Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020

11/18
Algorithm: Example II

![Graphical representation of a planning horizon and cost relationship with nodes and edges labeled with numerical values.]

Introduction
Model
Algorithm
Evaluation
Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz
Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
11/18
Algorithm: Example II

![Algorithm Diagram]

Introduction  Model  Algorithm  Evaluation  Conclusion
Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz — Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
Algorithm: Example II

Planning horizon

Cost

Introduction Model Algorithm Evaluation Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz – Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
Algorithm: Example II

Introduction

Model

Algorithm

Evaluation

Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz –
Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
Algorithm: Example II

Introduction

Model

Algorithm

Evaluation

Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz —
Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
Algorithm: Example II
Algorithm: Example II

Introduction Model Algorithm Evaluation Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz –
Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020    11/18
Algorithm: Example II

Introduction  Model  Algorithm  Evaluation  Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz — Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020
Algorithm: Example II
Algorithm: Example II
Algorithm: Example II

Introduction
Model
Algorithm
Evaluation
Conclusion

Alexander Kleff · Frank Schulz · Jakob Wagenblatt · Tim Zeitz –
Efficient Route Planning with Temporary Driving Restrictions

June 17th, 2020

11/18
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

<table>
<thead>
<tr>
<th>Planning horizon</th>
<th>Optimal Routes [#]</th>
<th>Arrival time deviation [h:mm]</th>
<th>Running time</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon. 18:00, 1 day</td>
<td>2.86</td>
<td>2:17</td>
<td>529.4</td>
<td>266.3</td>
<td></td>
</tr>
<tr>
<td>Mon. 18:00, 2 days</td>
<td>3.54</td>
<td>3:19</td>
<td>648.1</td>
<td>405.6</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fri. 06:00, 1 day</td>
<td>1.04</td>
<td>0:10</td>
<td>10.0</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>Fri. 06:00, 2 days</td>
<td>1.08</td>
<td>0:16</td>
<td>79.5</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>Fri. 18:00, 1 day</td>
<td>1.13</td>
<td>0:08</td>
<td>205.8</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>Fri. 18:00, 2 days</td>
<td>1.32</td>
<td>0:20</td>
<td>1028.1</td>
<td>0.7</td>
<td></td>
</tr>
</tbody>
</table>
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!