Flexible Route Planning with Contraction Hierarchies

Robert Geisberger, Moritz Kobitzsch and Peter Sanders – {geisberger,kobitzsch,sanders}@kit.edu

Institute for Theoretical Computer Science, Algorithmics II
Motivation

- route planning becomes ubiquitous
- route planning services e.g. Google Maps
- wide spread generates wide variety of interests
- multiple optimization criterias possible (e.g. (time|cost))
route planning becomes ubiquitous
route planning services e.g. Google Maps
wide spread generates wide variety of interests
multiple optimization criterias possible (e.g. (time|cost) )
Motivation

- route planning becomes ubiquitous
- route planning services e.g. Google Maps
- wide spread generates wide variety of interests
- multiple optimization criteria possible (e.g. (time|cost))
Motivation

- route planning becomes ubiquitous
- route planning services e.g. Google Maps
- wide spread generates wide variety of interests
- multiple optimization criteria possible (e.g. (time|cost) )
Just imagine...

Google maps

austin airport to hyatt regency

Get Directions  My Maps

A  austin airport
B  208 Barton Springs Road, Austin, TX 78704 (Hy:

Add Destination  Show options
By car
Get Directions

Also available:  Public Transit  Walking

fast

cheap

Austin-Bergstrom International Airport
Austin, TX 78719

1. Head east on Presidential Blvd  0.6 mi
2. Slight right to stay on Presidential Blvd  0.5 mi
3. Turn left at E State Hwy 71 Service Rd  0.2 mi
4. Take the ramp on the left onto Bastrop Hwy/ TX-71 W  1.1 mi
5. Take the ramp onto US-183 N  1.9 mi
find all non dominated routes
A dominates B if A is better or equal to B in every weight term
may result in an exponential number of routes
heuristics necessary to gain practical algorithms
Pareto Optimality

- find all non dominated routes
- $A$ dominates $B$ if $A$ is better or equal to $B$ in every weight term
- may result in an exponential number of routes
- heuristics necessary to gain practical algorithms
linear combination of weight terms

(t|c) ⇒ t + p · c

preprocessing considers all possible parameter values

query for fixed parameter ⇒ single criteria techniques

parameter range allows to select maximal impact of weight terms
Parameterized Optimality

- **linear combination** of weight terms
- \((t|c) \Rightarrow t + p \cdot c\)
- **preprocessing** considers **all possible** parameter values
- **query for fixed** parameter \(\Rightarrow\) single criteria techniques
- parameter range allows to select **maximal impact** of weight terms

```
\begin{align*}
p = 0 & \quad (3|6) \\
(3|1) \rightarrow 3 & \\
(1|2) \rightarrow 1 & \\
(2|1) \rightarrow 2 & \\
(3|0) \rightarrow 3 & \\
(1|2) \rightarrow 1 & \\
(1|1) \rightarrow 1 & \\
\end{align*}
```

Robert Geisberger, Moritz Kobitzsch and Peter Sanders:
Flexible Route Planning with Contraction Hierarchies

Faculty for Computer Science
Institute for Theoretical Computer Science, Algorithmics II
Parameterized Optimality

- linear combination of weight terms
- \((t|c) \Rightarrow t + p \cdot c\)
- preprocessing considers all possible parameter values
- query for fixed parameter \(\Rightarrow\) single criteria techniques
- parameter range allows to select maximal impact of weight terms

\[ p = 1 \]

\[ (4|4) \rightarrow (3|1) \rightarrow 4 \]

\[ (1|2) \rightarrow 3 \]

\[ (2|1) \rightarrow 3 \]

\[ (3|0) \rightarrow 3 \]

\[ (1|2) \rightarrow 3 \]

\[ (1|1) \rightarrow 2 \]
Parameterized Optimality

- linear combination of weight terms
- \((t|c) \Rightarrow t + p \cdot c\)
- preprocessing considers all possible parameter values
- query for fixed parameter \(\Rightarrow\) single criteria techniques
- parameter range allows to select maximal impact of weight terms
the distance function $d(a, f)$ for two nodes $a$, $f$ is concave

- monotone due to positive weight terms

![Graph with nodes and weights](image)

$\text{distance function } d(a, f, p)$

$p$
order nodes by importance \{v_1, \ldots, v_n\}
contract in this order
preserve original distances by adding shortcuts
necessity of shortcuts decided by witness paths
query only relaxes edges to more important nodes
Contraction Hierarchy Routing

- order nodes by importance \( \{v_1, \ldots, v_n\} \)
- contract in this order
- preserve original distances by adding shortcuts
- necessity of shortcuts decided by witness paths
- query only relaxes edges to more important nodes
Flexible Contraction Hierarchies

- shortcuts in flexible scenario if \( \exists p : (a, b, c) \) is the only shortest path
- necessity of shortcuts only on continuous intervals \( \Rightarrow \) store necessity interval in edges
- necessity interval on average deductible from two single criteria Dijkstra queries
- query uses only necessary edges
shortcuts in flexible scenario if \( \exists p : (a, b, c) \) is the only shortest path

- necessity of shortcuts only on continuous intervals \( \Rightarrow \) store necessity interval in edges

- necessity interval on average deductable from two single criteria Dijkstra queries

- query uses only necessary edges
Flexible Contraction Hierarchies

- Single node order not practical for wide parameter range.
- A lot of shortcuts for all parameter values.
- Split parameter interval with heuristics.
- Repeat as necessary.
- Buckets to support fast scanning of necessary edges.
Core-ALT for flex-CH

- Core based approach to
  - Speed up preprocessing (uncontracted core)
  - Speed up query (contracted core)
- Uses ALT algorithm on $|k|$ topmost nodes
- Adaptable to flexible scenario with linear interpolation

\[ d(a, f, p) \]
find all possible paths
profile query with maximal $3 \cdot \#\text{paths} - 2$ queries
approximation: recursion only if improvement larger more than $(1 + \varepsilon)$ possible
- find all possible paths
- profile query with maximal $3 \cdot \#\text{paths} - 2$ queries
- approximation: recursion only if improvement larger more than $(1 + \epsilon)$ possible

$d(a, f, p)$
Profile Queries

- find all possible paths
- profile query with maximal $3 \cdot \#paths - 2$ queries
- approximation: recursion only if improvement larger more than $(1 + \varepsilon)$ possible
find all possible paths

profile query with maximal $3 \cdot \#\text{paths} - 2$ queries

approximation: recursion only if improvement larger more than $(1 + \varepsilon)$ possible
# Experimental results

Table: Preprocessing and query performance for 64 landmarks on the German road network, average over 10,000 queries. (4,692,751 nodes and 10,806,191 directed edges)

<table>
<thead>
<tr>
<th>core</th>
<th>preproc</th>
<th>space</th>
<th>query</th>
<th>speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[hh:mm]</td>
<td>[B/node]</td>
<td>[ms]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>60</td>
<td>2,037.52</td>
</tr>
<tr>
<td>0</td>
<td>1:54</td>
<td>159</td>
<td>2.90</td>
<td>698</td>
</tr>
<tr>
<td>uncontracted 5,000</td>
<td>1:08</td>
<td>167</td>
<td>2.58</td>
<td>789</td>
</tr>
<tr>
<td>contracted 10,000</td>
<td>2:07</td>
<td>183</td>
<td>0.63</td>
<td>3,234</td>
</tr>
</tbody>
</table>
Experimental results
Comparison to SHARC

<table>
<thead>
<tr>
<th>algorithm</th>
<th>preproc</th>
<th>$\varepsilon$</th>
<th># paths</th>
<th>time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexCH</td>
<td>5 : 15</td>
<td>0.00</td>
<td>12.5</td>
<td>21.9</td>
</tr>
<tr>
<td>flexCH</td>
<td>5 : 15</td>
<td>0.01</td>
<td>5.7</td>
<td>6.2</td>
</tr>
<tr>
<td>Pareto-SHARC</td>
<td>7 : 12</td>
<td>(no guarantee)</td>
<td>5.3</td>
<td>35.4</td>
</tr>
</tbody>
</table>
Conclusion

- **exact flexible** routing for server scenarios
- **comparable** even to single criteria speedups
- circumvents problems of Pareto-optimality
- scales well through parameter splitting
Thank you for your attention.
Questions
Future Work

- mobile implementation
- apply parallelization techniques
- more than two weight terms (?)