Faster Multi-Modal Route Planning
with Bike Sharing Using ULTRA

SEA · June 17, 2020
Jonas Sauer, Dorothea Wagner, and Tobias Zündorf
Multi-Modal Route Planning

Goals:
- Journey planning for public transit
- Find **optimal** journeys
- Consider modes of transportation:
  - All timetable-based modes (trains, trams, buses, ...)

Institute of Theoretical Informatics
Algorithmics Group
Multi-Modal Route Planning

Goals:
- Journey planning for public transit
- Find **optimal** journeys
- Consider modes of transportation:
  - All timetable-based modes (trains, trams, buses, ...)
  - Walking (from, to, and between stops)
Multi-Modal Route Planning

Goals:

- Journey planning for public transit
- Find \textbf{optimal} journeys
- Consider modes of transportation:
  - All timetable-based modes (trains, trams, buses, ...)
  - Walking (from, to, and between stops)
  - Bike sharing (or other rental based services)
Multi-Modal Route Planning

Goals:
- Journey planning for public transit
- Find **optimal** journeys
- Consider modes of transportation:
  - All timetable-based modes (trains, trams, buses, ...)
  - Walking (from, to, and between stops)
  - Bike sharing (or other rental based services)
  - No limits on any of the transportation modes
Problem Statement

Given:
- Public transit network (timetable)
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
- Vertices (crossings, places)
Problem Statement

Given:

- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
- Vertices (crossings, places)
- Edges (roads, paths)
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
- Vertices (crossings, places)
- Edges (roads, paths)
- Bike sharing stations (per operator)
Problem Statement

Given:
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
- Vertices (crossings, places)
- Edges (roads, paths)
- Bike sharing stations (per operator)
Problem Statement

Given:

- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
- Vertices (crossings, places)
- Edges (roads, paths)
- Bike sharing stations (per operator)
- Source $s$, target $t$, and a departure time
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
- Vertices (crossings, places)
- Edges (roads, paths)
- Bike sharing stations (per operator)
- Source $s$, target $t$, and a departure time

Objective: Find all Pareto-optimal journeys w.r.t. arrival time and number of trips
Problem Statement

Given:
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)
- Source $s$, target $t$, and a departure time

Objective: Find all Pareto-optimal journeys w.r.t. arrival time and number of trips
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
- Vertices (crossings, places)
- Edges (roads, paths)
- Bike sharing stations (per operator)
- Source $s$, target $t$, and a departure time

Objective: Find all Pareto-optimal journeys w.r.t. arrival time and number of trips
Problem Statement

Given:
- Public transit network (timetable)
- Stops (bus stops, stations)
- Routes (bus lines, train lines)
- Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
- Vertices (crossings, places)
- Edges (roads, paths)
- Bike sharing stations (per operator)
- Source $s$, target $t$, and a departure time

Objective: Find all Pareto-optimal journeys w.r.t. arrival time and number of trips
Approaches

Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared
Approaches

Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared
Approaches

Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared
Approaches

Greatest Challenge:
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared
Approaches

Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared

![Diagram showing routes and times]
Approaches

Greatest Challenge:
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared
Approaches

Greatest Challenge:
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared
Approaches

Greatest Challenge:
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared

Two Possible Solutions:
Approaches

Greatest Challenge:
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared

Two Possible Solutions:
- The **Operator-Dependent** (OD) model
  - Handle operators in the algorithm explicitly
  - Similar to a third dominance criterion
Approaches

Greatest Challenge:
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared

Two Possible Solutions:
- **The Operator-Dependent (OD) model**
  - Handle operators in the algorithm explicitly
  - Similar to a third dominance criterion
- **The Operator-Expanded (OE) model**
  - Encode operators within a “normal” network
  - Use an existing algorithm with the modified network
The Operator-Dependent (OD) Model

Basic Idea:

- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm
The Operator-Dependent (OD) Model

Basic Idea:
- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

Integration into multi-Modal multi-Criteria RAPTOR (MCR): [Delling et al. 2013]
- Naive:
  - Use label-bags of MCR for bike sharing operators
The Operator-Dependent (OD) Model

Basic Idea:
- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

Integration into multi-Modal multi-Criteria RAPTOR (MCR): [Delling et al. 2013]
- Naive:
  - Use label-bags of MCR for bike sharing operators

Bag per stop, #trips
The Operator-Dependent (OD) Model

Basic Idea:
- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

Integration into multi-Modal multi-Criteria RAPTOR (MCR): [Delling et al. 2013]
- Naive:
  - Use label-bags of MCR for bike sharing operators
- Observation:
  - Bike sharing operators are few and discrete
  - Scan routes separately for each operator
The Operator-Dependent (OD) Model

**Basic Idea:**
- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

**Integration into multi-Modal multi-Criteria RAPTOR (MCR):** [Delling et al. 2013]
- **Naive:**
  - Use label-bags of MCR for bike sharing operators
- **Observation:**
  - Bike sharing operators are few and discrete
  - Scan routes separately for each operator
The Operator-Expanded (OE) Model

Basic Idea:
- Encode bike sharing within a “normal” network
The Operator-Expanded (OE) Model

Basic Idea:
- Encode bike sharing within a “normal” network

Our Approach:
The Operator-Expanded (OE) Model

Basic Idea:
- Encode bike sharing within a “normal” network

Our Approach:
The Operator-Expanded (OE) Model

Basic Idea:
- Encode bike sharing within a “normal” network

Our Approach:
- Copy network once per operator
The Operator-Expanded (OE) Model

Basic Idea:
- Encode bike sharing within a “normal” network

Our Approach:
- Copy network once per operator
- Connect networks at bike sharing stations
The Operator-Expanded (OE) Model

Basic Idea:
- Encode bike sharing within a “normal” network

Our Approach:
- Copy network once per operator
- Connect networks at bike sharing stations
The Operator-Expanded (OE) Model

Basic Idea:
- Encode bike sharing within a “normal” network

Our Approach:
- Copy network once per operator
- Connect networks at bike sharing stations

Properties:
- Any existing algorithm can run on this network
- Using the green network ⇔ Renting a green bike
  (Using the blue network ⇔ Renting a blue bike)
The Operator-Expanded (OE) Model

Basic Idea:
- Encode bike sharing within a “normal” network

Our Approach:
- Copy network once per operator
- Connect networks at bike sharing stations

Properties:
- Any existing algorithm can run on this network
- Using the green network ⇔ Renting a green bike
  (Using the blue network ⇔ Renting a blue bike)
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network

Operator Hull $\mathcal{H}$:
- Subset of the network
- For every bike sharing operator $o$
- For every vertex/edge/trip $x$ in the network
- If $x$ is used with a bike of $o$ in some optimal journey $\Rightarrow x \in \mathcal{H}(o)$
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network

Operator Hull $\mathcal{H}$:
- Subset of the network
- For every bike sharing operator $o$
- For every vertex/edge/trip $x$ in the network
- If $x$ is used with a bike of $o$ in some optimal journey $\Rightarrow x \in \mathcal{H}(o)$
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network

Operator Hull $\mathcal{H}$:
- Subset of the network
- For every bike sharing operator $o$
- For every vertex/edge/trip $x$ in the network
- If $x$ is used with a bike of $o$ in some optimal journey $\Rightarrow x \in \mathcal{H}(o)$

Preprocessing:
- Computing $\mathcal{H}(o)$ can be done with standard MCR
Observation:
- Not every rental bike is useful throughout the whole network
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network

Operator-Dependent Queries:
- Use $\mathcal{H}(o)$ to prune the search space
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network

Operator-Dependent Queries:
- Use $\mathcal{H}(o)$ to prune the search space

Operator-Expanded Queries:
- Build a reduced Network
- Do not copy the whole network
- Use $\mathcal{H}(o)$ as copy for operator $o$
Speed-up Technique: Operator Pruning (OP)

Observation:
- Not every rental bike is useful throughout the whole network

Operator-Dependent Queries:
- Use $\mathcal{H}(o)$ to prune the search space

Operator-Expanded Queries:
- Build a reduced Network
- Do not copy the whole network
- Use $\mathcal{H}(o)$ as copy for operator $o$
Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts
ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts
Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

ULTRA (UnLimited TRAnsfers) overview:
- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts

Adaptation for Bike Sharing:
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented
Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

ULTRA (UnLimited TRAnsfers) overview:
- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts

Adaptation for Bike Sharing:
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented
Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

ULTRA (UnLimited TRAnsfers) overview:
- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts

Adaptation for Bike Sharing:
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented
Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

ULTRA (UnLimited TRAnsfers) overview:
- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts

Adaptation for Bike Sharing:
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented
Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

ULTRA (UnLimited TRAnsfers) overview:
- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts

Adaptation for Bike Sharing:
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented
Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

ULTRA (UnLimited TRAnsfers) overview:
- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts

Adaptation for Bike Sharing:
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented

Solution:
- Perform the ULTRA preprocessing on the operator-expanded network
Experimental Evaluation

Instances:

- London, Switzerland, and Germany
- Timetables comprising two days from TfL, GTFS-CH, and DB
- Transfer graphs and bike sharing stations from OpenStreetMap

<table>
<thead>
<tr>
<th>Network</th>
<th>Stops</th>
<th>Routes</th>
<th>Trips</th>
<th>Vertices</th>
<th>Edges</th>
<th>Stations</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>London</td>
<td>20 595</td>
<td>2 107</td>
<td>125 k</td>
<td>183 k</td>
<td>579 k</td>
<td>823</td>
<td>4</td>
</tr>
<tr>
<td>Switzerland</td>
<td>25 426</td>
<td>13 934</td>
<td>369 k</td>
<td>604 k</td>
<td>1 847 k</td>
<td>534</td>
<td>11</td>
</tr>
<tr>
<td>Germany</td>
<td>244 055</td>
<td>231 089</td>
<td>2 387 k</td>
<td>6 872 k</td>
<td>21 372 k</td>
<td>2 682</td>
<td>22</td>
</tr>
</tbody>
</table>
Experimental Evaluation – Preprocessing

Impact of Operator-Pruning:

- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

<table>
<thead>
<tr>
<th></th>
<th>London</th>
<th>Switzerland</th>
<th>Germany</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded stops</td>
<td>102,975</td>
<td>31,216</td>
<td>301,500</td>
</tr>
<tr>
<td>ULTRA shortcuts</td>
<td>1,831,779</td>
<td>521,882</td>
<td>3,389,309</td>
</tr>
<tr>
<td>Operator hulls (parallel 16)</td>
<td>–</td>
<td>15:34</td>
<td>–</td>
</tr>
<tr>
<td>Total (CH + OP + ULTRA)</td>
<td>14:15:19</td>
<td>59:33</td>
<td>10:01:54</td>
</tr>
</tbody>
</table>
## Experimental Evaluation – Preprocessing

### Impact of Operator-Pruning:
- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

<table>
<thead>
<tr>
<th></th>
<th>London</th>
<th>Switzerland</th>
<th>Germany</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded stops</td>
<td>102,975</td>
<td>31,216</td>
<td>301,500</td>
</tr>
<tr>
<td>ULTRA shortcuts</td>
<td>1,831,779</td>
<td>521,882</td>
<td>3,389,309</td>
</tr>
<tr>
<td>Operator hulls (parallel 16)</td>
<td>–</td>
<td>15:34</td>
<td>–</td>
</tr>
<tr>
<td>Total (CH + OP + ULTRA)</td>
<td>14:15:19</td>
<td>59:33</td>
<td>10:01:54</td>
</tr>
</tbody>
</table>
Impact of Operator-Pruning:
- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

<table>
<thead>
<tr>
<th></th>
<th>London</th>
<th></th>
<th>Switzerland</th>
<th></th>
<th>Germany</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded stops</td>
<td>102 975</td>
<td>31 216</td>
<td>301 500</td>
<td>36 892</td>
<td>5 613 265</td>
<td>411 980</td>
</tr>
<tr>
<td>ULTRA shortcuts</td>
<td>1 831 779</td>
<td>521 882</td>
<td>3 389 309</td>
<td>435 514</td>
<td>?</td>
<td>7 873 379</td>
</tr>
<tr>
<td>Operator hulls (parallel 16)</td>
<td>–</td>
<td>15:34</td>
<td>–</td>
<td>4:15</td>
<td>–</td>
<td>8:45:22</td>
</tr>
<tr>
<td>Total (CH + OP + ULTRA)</td>
<td>14:15:19</td>
<td>59:33</td>
<td>10:01:54</td>
<td>28:03</td>
<td>≈21 weeks</td>
<td>40:13:48</td>
</tr>
</tbody>
</table>
### Impact of Operator-Pruning:

- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

<table>
<thead>
<tr>
<th></th>
<th>London</th>
<th>Switzerland</th>
<th>Germany</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded stops</td>
<td>102 975</td>
<td>31 216</td>
<td>301 500</td>
</tr>
<tr>
<td>ULTRA shortcuts</td>
<td>1 831 779</td>
<td>521 882</td>
<td>3 389 309</td>
</tr>
<tr>
<td>Operator hulls (parallel 16)</td>
<td>–</td>
<td>15:34</td>
<td>–</td>
</tr>
<tr>
<td>Total (CH + OP + ULTRA)</td>
<td>14:15:19</td>
<td>59:33</td>
<td>10:01:54</td>
</tr>
</tbody>
</table>
## Impact of Operator-Pruning:
- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

<table>
<thead>
<tr>
<th></th>
<th>London</th>
<th>Switzerland</th>
<th>Germany</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded stops</td>
<td>102,975</td>
<td>301,500</td>
<td>5,613,265</td>
</tr>
<tr>
<td>ULTRA shortcuts</td>
<td>1,831,779</td>
<td>3,389,309</td>
<td>?</td>
</tr>
<tr>
<td>Operator hulls (parallel 16)</td>
<td>– 15:34</td>
<td>– 4:15</td>
<td>– 8:45:22</td>
</tr>
<tr>
<td>Total (CH + OP + ULTRA)</td>
<td>14:15:19 59:33</td>
<td>10:01:54 28:03</td>
<td>21 weeks 40:13:48</td>
</tr>
</tbody>
</table>
Experimental Evaluation – Query

Average Running Times:
- Combining ULTRA, OE, and OP yields the fastest algorithm

<table>
<thead>
<tr>
<th>Network</th>
<th>Algorithm</th>
<th>Preprocessing</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time [h:m:s]</td>
<td>Rounds</td>
</tr>
<tr>
<td>Switzerland</td>
<td>MCR-OD</td>
<td>0:56</td>
<td>9.55</td>
</tr>
<tr>
<td></td>
<td>MCR-OE</td>
<td>1:02</td>
<td>9.55</td>
</tr>
<tr>
<td></td>
<td>MCR-OE-OP</td>
<td>5:40</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>ULTRA-OE-OP</td>
<td>28:03</td>
<td>8.48</td>
</tr>
<tr>
<td>Germany</td>
<td>MCR-OD</td>
<td>13:19</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>MCR-OE</td>
<td>15:21</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>MCR-OE-OP</td>
<td>9:05:48</td>
<td>10.24</td>
</tr>
<tr>
<td></td>
<td>ULTRA-OE-OP</td>
<td>40:13:48</td>
<td>10.38</td>
</tr>
</tbody>
</table>
### Average Running Times:

Combining ULTRA, OE, and OP yields the fastest algorithm.

<table>
<thead>
<tr>
<th>Network</th>
<th>Algorithm</th>
<th>Preprocessing</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time [h:m:s]</td>
<td>Rounds</td>
</tr>
<tr>
<td>Switzerland</td>
<td>MCR-OD</td>
<td>0:56</td>
<td>9.55</td>
</tr>
<tr>
<td></td>
<td>MCR-OE</td>
<td>1:02</td>
<td>9.55</td>
</tr>
<tr>
<td></td>
<td>MCR-OE-OP</td>
<td>5:40</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>ULTRA-OE-OP</td>
<td>28:03</td>
<td>8.48</td>
</tr>
<tr>
<td>Germany</td>
<td>MCR-OD</td>
<td>13:19</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>MCR-OE</td>
<td>15:21</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>MCR-OE-OP</td>
<td>9:05:48</td>
<td>10.24</td>
</tr>
<tr>
<td></td>
<td>ULTRA-OE-OP</td>
<td>40:13:48</td>
<td>10.38</td>
</tr>
</tbody>
</table>
### Experimental Evaluation – Query

#### Average Running Times:

- Combining ULTRA, OE, and OP yields the fastest algorithm

<table>
<thead>
<tr>
<th>Network</th>
<th>Algorithm</th>
<th>Preprocessing</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time [h:m:s]</td>
<td>Rounds</td>
</tr>
<tr>
<td>Switzerland</td>
<td>MCR-OD</td>
<td>0:56</td>
<td>9.55</td>
</tr>
<tr>
<td></td>
<td>MCR-OE</td>
<td>1:02</td>
<td>9.55</td>
</tr>
<tr>
<td></td>
<td>MCR-OE-OP</td>
<td>5:40</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>ULTRA-OE-OP</td>
<td>28:03</td>
<td>8.48</td>
</tr>
<tr>
<td>Germany</td>
<td>MCR-OD</td>
<td>13:19</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>MCR-OE</td>
<td>15:21</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>MCR-OE-OP</td>
<td>9:05:48</td>
<td>10.24</td>
</tr>
<tr>
<td></td>
<td>ULTRA-OE-OP</td>
<td>40:13:48</td>
<td>10.38</td>
</tr>
</tbody>
</table>
## Average Running Times:

Combining ULTRA, OE, and OP yields the fastest algorithm

<table>
<thead>
<tr>
<th>Network</th>
<th>Algorithm</th>
<th>Preprocessing</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time [h:m:s]</td>
<td>Rounds</td>
</tr>
<tr>
<td>Switzerland</td>
<td>MCR-OD</td>
<td>0:56</td>
<td>9.55</td>
</tr>
<tr>
<td></td>
<td>MCR-OE</td>
<td>1:02</td>
<td>9.55</td>
</tr>
<tr>
<td></td>
<td>MCR-OE-OP</td>
<td>5:40</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>ULTRA-OE-OP</td>
<td>28:03</td>
<td>8.48</td>
</tr>
<tr>
<td>Germany</td>
<td>MCR-OD</td>
<td>13:19</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>MCR-OE</td>
<td>15:21</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>MCR-OE-OP</td>
<td>9:05:48</td>
<td>10.24</td>
</tr>
<tr>
<td></td>
<td>ULTRA-OE-OP</td>
<td>40:13:48</td>
<td>10.38</td>
</tr>
</tbody>
</table>
Experimental Evaluation – Query

Running Times Depending on Number of Operators:

- Operator-expanded model benefits more from operator-pruning
- ULTRA reduces query time significantly

![Box plot showing running times depending on the number of available bike sharing operators. The plot compares MCR-OD-OP, MCR-OE-OP, and ULTRA-OE-OP models. The x-axis represents the number of available operators, ranging from 0 to 22. The y-axis represents query time in seconds, ranging from 0 to 5. The plot indicates that ULTRA reduces query time significantly compared to the other models.](image-url)
Conclusion

Our Contribution:
- We introduced two new approaches for modeling bike sharing:
  - Operator-Dependent
  - Operator-Expanded
- We presented a novel speed-up technique: Operator-Pruning
- Overall, we are more than 10 times faster than the base-line