Public Transit Routing with Unrestricted Walking

Dorothea Wagner and Tobias Zündorf

ATMOS · September 7th, 2017
Overview

Introduction:
- Problem Statement
- State of the Art

Our Contribution:
- Our Profile Algorithm
- Direct Walking

Evaluation:
- Instances & Experimental Setup
- Runtime Performance
- Travel Time Comparison
Problem – Public Transit & Walking

Public Transit Routing:

Given:
- Stops (where vehicles can be entered or alighted)
- The timetable (routes, trips, connections)
- Footpaths (possible transfers between stops)

Goal:
- Optimal journey between two stops
- With respect to travel time, number of transfers, ...
Problem – Public Transit & Walking

Public Transit Routing:

Given:
- Stops (where vehicles can be entered or alighted)
- The timetable (routes, trips, connections)
- Footpaths (possible transfers between stops)

Goal:
- Optimal journey between two stops
- With respect to travel time, number of transfers, ...
Problem – Public Transit & Walking

Public Transit Routing:

Given:
- Stops (where vehicles can be entered or alighted)
- The timetable (routes, trips, connections)
- Footpaths (possible transfers between stops)

Goal:
- Optimal journey between two stops
- With respect to travel time, number of transfers, ...

Question:
- How many footpaths are needed?
- How are travel times affected by different footpaths?
Problem – Public Transit & Walking

Public Transit Routing:
Given:
- Stops (where vehicles can be entered or alighted)
- The timetable (routes, trips, connections)
- Footpaths (possible transfers between stops)

Goal:
- Optimal journey between two stops
- With respect to travel time, number of transfers, ...

Question:
- How many footpaths are needed?
- How are travel times affected by different footpaths?
Problem – Public Transit & Walking

Public Transit Routing:
Given:
- Stops (where vehicles can be entered or alighted)
- The timetable (routes, trips, connections)
- Footpaths (possible transfers between stops)

Goal:
- Optimal journey between two stops
- With respect to travel time, number of transfers, ...

Question:
- How many footpaths are needed?
- How are travel times affected by different footpaths?
Related Work

State of the Art:
- Many different footpath models
- Often require restrictions

Common Restrictions:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Footpaths</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAPTOR [Delling et al. '12/’14]</td>
<td>Transitivity closed</td>
</tr>
<tr>
<td>CSA [Dibbelt et al. ’13/’14]</td>
<td>Transitivity closed</td>
</tr>
<tr>
<td>Trip-Based Routing [Witt ’15]</td>
<td>Transitivity closed</td>
</tr>
<tr>
<td>Transfer Patterns [Bast et al. ’10/’16]</td>
<td>Max. 400 meters</td>
</tr>
<tr>
<td>Frequency-Based [Bast, Storandt ’14]</td>
<td>Max. 15 minutes</td>
</tr>
<tr>
<td>Public Transit Labeling [Delling et al. ’15]</td>
<td>As specified by the timetable</td>
</tr>
</tbody>
</table>
Reasons for Restricted Walking

Common Arguments:
- Passengers do not want to walk far
- Restricted walking is sufficient for optimal solutions
Reasons for Restricted Walking

Common Arguments:
- Passengers do not want to walk far
- Restricted walking is sufficient for optimal solutions

But:
- Some passengers might want to walk far
- Decision is made without knowing how much walking is required
- It has not been proven how much walking is required
How to Evaluate the Importance of Walking

Important Aspect:
- Departure time (day vs. night)
- Query distance (short range vs. long range)
- Source and target location (rural vs. urban)
How to Evaluate the Importance of Walking

Important Aspect:
- Departure time (day vs. night)
- Query distance (short range vs. long range)
- Source and target location (rural vs. urban)

Thus, we need:
- Profile algorithms (analyze travel times for a whole day)
- Realistic queries for several source to target distances
- Instances with different amounts of footpaths
Profiles & Profile Algorithms

**Definition:**
- Function mapping **departure time** to either **travel time** or **arrival time**
Profiles & Profile Algorithms

Definition:
- Function mapping departure time to either travel time or arrival time

Profile Algorithms for Unrestricted Walking:
- Most algorithms are not suitable for unrestricted walking
- Multimodal Multicriteria RAPTOR (MCR) [Delling et al. ’13]
  - Can handle arbitrary walking
  - Only earliest arrival queries
Profiles – RAPTOR & MCR

MCR Profile Algorithm?:
- MCR is based on RAPTOR
- RAPTOR can be used to compute profiles (rRAPTOR)
- Why does this not work for MCR?
Profiles – RAPTOR & MCR

MCR Profile Algorithm?:
- MCR is based on RAPTOR
- RAPTOR can be used to compute profiles (rRAPTOR)
- Why does this not work for MCR?

rRAPTOR Profile Query: (rough outline)
- Profile entries are limited by number of trips departing from source
- Collect all departure times at the source
- Run RAPTOR once for every departure time
Profiles – RAPTOR & MCR

MCR Profile Algorithm?:
- MCR is based on RAPTOR
- RAPTOR can be used to compute profiles (rRAPTOR)
- Why does this not work for MCR?

rRAPTOR Profile Query: (rough outline)
- Profile entries are limited by number of trips departing from source
- Collect all departure times at the source
- Run RAPTOR once for every departure time

Problem with direct walking / MCR:
- Passenger can walk to arbitrary stop before boarding the first trip
- Every departure time needs to be considered
  ⇒ To many calls of RAPTOR
Our Algorithm

Goal:
- Use earliest arrival algorithm as black box (MCR)
- Earliest arrival algorithm computes single profile entry
- Number of algorithm calls \( \approx \) number of profile entries
Our Algorithm

Goal:
- Use earliest arrival algorithm as black box (MCR)
- Earliest arrival algorithm computes single profile entry
- Number of algorithm calls \( \approx \) number of profile entries

Overview:
- Establish profile one entry at a time
- Use two black box calls per entry (forward + backward)
  - Start with earliest possible departure
  - Forward query \( \Rightarrow \) earliest possible arrival
  - Backward query \( \Rightarrow \) latest possible departure for that arrival
  - Continue with latest possible departure \( + \varepsilon \)
Our Algorithm – Example

Goal:
- Compute a profile for the interval [0:00, 24:00]
Our Algorithm – Example

Goal:
- Compute a profile for the interval [0:00, 24:00]

**Step 1:** Run a forward query from the earliest departure time (0:00)
Our Algorithm – Example

Goal:
- Compute a profile for the interval [0:00, 24:00]

Step 2: Run a backward query from the resulting arrival time (8:00)
Our Algorithm – Example

Goal:
- Compute a profile for the interval [0:00, 24:00]

Step 3: Both results define one point of the profile

![Graph showing a profile for the interval [0:00, 24:00]](image_url)
Our Algorithm – Example

Goal:
- Compute a profile for the interval [0:00, 24:00]

Step 4: The profile is correct for the interval [0:00, 4:00]
Goal:
- Compute a profile for the interval \((4:00, 24:00]\)

**Step 1:** Run a forward query from the earliest departure time \((4:00 + \varepsilon)\)
Our Algorithm – Example

Goal:
- Compute a profile for the interval (4:00, 24:00]

Step 2: Run a backward query from the resulting arrival time (11:00)
Our Algorithm – Example

Goal:
- Compute a profile for the interval (4:00, 24:00]

**Step 3:** Both results define one point of the profile
Our Algorithm – Example

Goal:
- Compute a profile for the interval (4:00, 24:00]

Step 4: The profile is correct for the interval [0:00, 7:00]
Our Algorithm – Example

Goal:
- Compute a profile for the interval (7:00, 24:00]

**Step 1:** Run a forward query from the earliest departure time \((7:00 + \varepsilon)\)
Our Algorithm – Example

Goal:
- Compute a profile for the interval \((7:00, 24:00]\)

End: The algorithm terminates when the profile is complete
Our Algorithm – Direct Walking

Problem:
- **Direct walking** prevents progress

Example:
Direct Walking takes 6 hours
Our Algorithm – Direct Walking

**Goal:**
- Compute a profile for the interval \((4:00, 24:00]\)

**Step 1:** Run a forward query from the earliest departure time \((4:00 + \varepsilon)\)

**Example:**
Direct Walking takes 6 hours
Goal:
- Compute a profile for the interval (4:00, 24:00]

**Step 2:** Run a backward query from the resulting arrival time (10:00)

Example: Direct Walking takes 6 hours
Our Algorithm – Direct Walking

Goal:
- Compute a profile for the interval (4:00, 24:00]

Step 3: Both results define one point of the profile

Example:
Direct Walking takes 6 hours
Goal:
- Compute a profile for the interval (4:00, 24:00]

**Step 4:** The profile is still only correct for the interval [0:00, 4:00]

**Example:**
Direct Walking takes 6 hours
Our Algorithm – Direct Walking

Problem:
- Direct walking prevents progress

Solution: Compute profile in network without direct walking & Add direct Walking path to the solution afterwards

Example: Direct Walking takes 6 hours
Evaluation

Instances:
- Germany (from bahn.de)
- Switzerland (GTFS feed)
- Footpath graph from OpenStreetMap (OSM)

<table>
<thead>
<tr>
<th>PT network</th>
<th>Footpaths</th>
<th>Stops</th>
<th>Vertices</th>
<th>Edges</th>
<th>Connections</th>
<th>Max. deg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germany</td>
<td>original</td>
<td>244 245</td>
<td>244 245</td>
<td>95 036</td>
<td>46 119 896</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>partial</td>
<td>244 245</td>
<td>244 245</td>
<td>26 193 136</td>
<td>46 119 896</td>
<td>2622</td>
</tr>
<tr>
<td></td>
<td>complete</td>
<td>244 245</td>
<td>6 876 758</td>
<td>21 382 408</td>
<td>46 119 896</td>
<td>21</td>
</tr>
<tr>
<td>Switzerland</td>
<td>original</td>
<td>25 427</td>
<td>25 427</td>
<td>5 604</td>
<td>4 373 268</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>partial</td>
<td>25 427</td>
<td>25 427</td>
<td>3 104 974</td>
<td>4 373 268</td>
<td>1246</td>
</tr>
<tr>
<td></td>
<td>complete</td>
<td>25 427</td>
<td>604 230</td>
<td>1 844 286</td>
<td>4 373 268</td>
<td>25</td>
</tr>
</tbody>
</table>
Evaluation – Partial Instances

Goal:
- Preserve paths between stops up to a certain walking time
- Obtain a transitively closed graph of reasonable size

![Graph showing connected components and isolated stops for Switzerland and Germany](image)
Evaluation – Partial Instances

Goal:
- Preserve paths between stops up to a certain walking time
- Obtain a transitively closed graph of reasonable size
- Reasonable size $\approx$ average vertex degree $\leq 100$
  $\Rightarrow$ Switzerland 15 min walking
  $\Rightarrow$ Germany 8 min walking
Evaluation – Queries

Problem:
- Random queries do not reflect reality
- Could result in an overestimation of the importance of walking
- Real queries are not available
Evaluation – Queries

Problem:
- Random queries do not reflect reality
- Could result in an overestimation of the importance of walking
- Real queries are not available

Assumption:
- Stop is often used as source/target ⇔ Stop is part of many trips

Our Queries:
- Choose source at random
- Probability is proportional to number of trips containing the stop
- Choose target at random from stops with certain distance rank
Evaluation – Running Time

Run time comparison:
- Switzerland public transit network
- Computation of Pareto-profiles (travel time, number of transfers)
- CSA for the transitively closed instances
- Our algorithm for the complete instance

![Run time comparison chart]

<table>
<thead>
<tr>
<th>Distance Rank ($2^X$)</th>
<th>Time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>$10^3$</td>
</tr>
<tr>
<td>8</td>
<td>$10^4$</td>
</tr>
<tr>
<td>9</td>
<td>$10^5$</td>
</tr>
<tr>
<td>10</td>
<td>$10^4$</td>
</tr>
<tr>
<td>11</td>
<td>$10^5$</td>
</tr>
<tr>
<td>12</td>
<td>$10^3$</td>
</tr>
<tr>
<td>13</td>
<td>$10^4$</td>
</tr>
<tr>
<td>14</td>
<td>$10^5$</td>
</tr>
<tr>
<td>15</td>
<td>$10^3$</td>
</tr>
<tr>
<td>16</td>
<td>$10^4$</td>
</tr>
<tr>
<td>17</td>
<td>$10^5$</td>
</tr>
<tr>
<td>18</td>
<td>$10^3$</td>
</tr>
</tbody>
</table>
Evaluation – Travel Time Results

Travel time comparison:
- Switzerland complete vs. Switzerland original (Distance rank 16)

Average travel time (complete)
Evaluation – Travel Time Results

**Travel time comparison:**
- Switzerland complete vs. Switzerland original *(Distance rank 16)*

- **Average travel time (complete)**
- **Average travel time (original)**

![Graph showing travel time comparison](image-url)
Evaluation – Travel Time Results

Travel time comparison:
- Switzerland complete vs. Switzerland original (Distance rank 16)

- Average travel time (complete)
- Average travel time (original)
- Median of travel time difference
Evaluation – Travel Time Results

Travel time comparison:

- Switzerland complete vs. Switzerland original (Distance rank 16)

- Average travel time (complete)
- Average travel time (original)
- Median of travel time difference
- Interquartile range of travel time diff.

![Travel Time Comparison Graph]

- **Departure time**
  - 0:00
  - 6:00
  - 12:00
  - 18:00
  - 24:00

- **Travel Time Comparison**
  - 100%
  - 75%
  - 50%
  - 25%
  - 0%

- **Average travel time**
  - Switzerland complete
  - Switzerland original

- **Median of travel time difference**
- **Interquartile range of travel time diff.**
Evaluation – Travel Time Results

Travel time comparison:
- Switzerland complete vs. Switzerland original (Distance rank 16)

- Average travel time (complete)
- Average travel time (original)
- Median of travel time difference
- Interquartile range of travel time diff.
- Percentage of differing travel times

![Graph showing travel time comparison](image-url)
Evaluation – Travel Time Results

Travel time comparison:
- Switzerland complete vs. Switzerland original (Distance rank 16)

- [Graph showing travel time comparison with legend:
  - Average travel time (complete)
  - Average travel time (original)
  - Median of travel time difference
  - Interquartile range of travel time diff.
  - Percentage of differing travel times
  - Percentage with difference > 1h]
Evaluation – Travel Time Results

**Travel time comparison:**

- Switzerland complete vs. Switzerland partial (Distance rank 16)

- **Average travel time (complete)**
- **Average travel time (original)**
- **Median of travel time difference**
- **Interquartile range of travel time diff.**
- **Percentage of differing travel times**
- **Percentage with difference > 1h**

![Graph showing travel time comparison](image-url)
Conclusion

Results:
- New public transit profile algorithm for unrestricted walking
- Detailed comparison of common footpath graphs
- Walking has a strong influence on the travel time
- Footpaths specified in the timetable are not sufficient

Future work:
- Initial and final walking vs. walking between trips
- More efficient algorithms for unrestricted walking
Conclusion

Results:
- New public transit profile algorithm for unrestricted walking
- Detailed comparison of common footpath graphs
- Walking has a strong influence on the travel time
- Footpaths specified in the timetable are not sufficient

Future work:
- Initial and final walking vs. walking between trips
- More efficient algorithms for unrestricted walking

Thank you for your attention!