Reititin
an open source tool for analysing accessibility by public transport in Greater Helsinki

Juha Järvi, Maria Salonen, Perttu Saarsalmi, Henrikki Tenkanen, and Tuuli Toivonen

BusFaster Oy and University of Helsinki
MetropAccess project
Background

- Large public transport projects planned and ongoing in Greater Helsinki area.
Motivation

• Quantitative analysis of reachability by public transport now and in the future.
• Travel times by public transport are more complicated than walking or cycling.
• Good route calculation services available but making a million queries is impolite.
• Goal was a flexible reachability analysis tool for researchers and the general public.
Public transport routing pitfalls

• Transfers are sensitive to exact schedules.
• Earliest arrival vs fastest or most practical route. Easiest to try with different departure times.
• No fancy way to calculate routes combining transport lines with transfers on foot or bicycle. Simple 1960s style algorithms.
Technology

• We want to allow others to later easily try complicated analyses, but without overloading our computers.
• Algorithms selected to also run inside a web browser.
• Challenging constraints for memory usage, calculation time, data size and licensing.
Methods

- Map, route and schedule compression.
- OpenStreetMap allows redistributing raw map data, globally available.
- Only store information relevant to routing.
- Simple Dijkstra-based shortest path algorithm.
Compression

- Public transport schedule GTFS and Kalkati format data is very repetitive.
- OpenStreetMap contains data irrelevant to routing and compressed representation can be improved.
- We reduced schedule size by over 98% and map size by over 80% compared to original DEFLATE compressed data.
Compression methods 1/2

- Reorder data, group by physical proximity or logical similarity.
- Delta encoding, only store differences between nearby coordinates or times.
- Use variable number of bytes for numbers.
Compression methods 2/2

- Replace duplicate data by a reference to previously stored copy.
- Cache data for different contexts to use shorter references. Transport stops that previously followed another, road map points in same rectangular tile...
Routing

• Optimize for cost, also tracking time.
• Both measured in tenths of a second but walking costs more than actual time spent and transfers have a fixed extra cost.
• Accurate road map and schedules, possible to disable specific public transport routes or define new custom ones.
Routing methods 1/2

- Flexible Dijkstra implementation using “visitor objects” for road, stop and vehicle.
- Routing starts with a single visitor. When visited, it creates new visitors for connected locations and calculates the time and cost for reaching them.
- Once visited, visitors are destroyed (recycled) to save memory.
Routing methods 2/2

- Public transport schedules are handled by stop and vehicle visitors. All costs are unknown until visiting neighbor locations.
- No fancy data structures. Visitors with countably finite possible costs stored in a simple list.
- Visitors are called in order of increasing cost until maximum cost is reached.
Results

• Under 3 seconds to calculate routes in a web browser between a single point and all other points in the Greater Helsinki area.
• Command line tool for millions of routes.
• Used in the MetropAccess project to analyze reachability of libraries, grocery stores at different times of the day, effects of the new subway extension...
Source: University of Helsinki MetropAccess project, Tuuli Toivonen et al.
Open data, open source

- Public transport
  - Kalkati.net XML format.
  - Google Transit Feed Specification.
- Road network
  - OpenStreetMap.
- Code
  - blogs.helsinki.fi/saavutettavuus
Thank you

Juha Järvi
BusFaster
juha@busfaster.com

Code at:
blogs.helsinki.fi/saavutettavuus