THIRD OPEN SOURCE GEOSPATIAL RESEARCH & EDUCATION SYMPOSIUM (OGRS)
Espoo, Finland, 10-13 June 2014

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

Juha Järvi[1], Maria Salonen[2], Perttu Saarsalmi[2], Henrikki Tenkanen[2], and Tuuli Toivonen[2,3]

[1]BusFaster Ltd. juha@busfaster.com
[2]Department of Geosciences and Geography, P.O. Box 64, 00014 University of Helsinki
[3]Department of Biosciences, P.O. Box 65, 00014 University of Helsinki

---

Abstract: This paper presents a public transport routing system suitable for complicated reachability analyses. Its main goals are to be flexible, easy to extend and allow creating public services where large numbers of people can calculate routes and examine reachability of different locations, with no server software or installation needed.

Keywords: Routing, public transport

---

1 Introduction

Accessibility analysis is considered an appropriate way to assess interactions between transportation and land use (Bertolini et al. 2005, Silva & Pinho 2010). Different distance measures typically form an integral part of accessibility indicators. As one such measure, travel time is considered to correspond well to people's perceptions of friction of distance (Frank et al. 2008, Mavoa et al. 2012). Traditionally travel time has been calculated using privately owned car as the subject. Concern over the environmental and social sustainability of land use and transportation solutions has in recent years highlighted the need to incorporate different modes of transport in accessibility analyses. Particularly, there is a need to properly analyse accessibility using public transportation (Salonen & Toivonen 2013). Until recently, however, adequate data and tools for detailed public transport analysis have been lacking.

During the last years Finland, as many other countries in Europe, has witnessed a fast development in data availability. Helsinki region has been one of the active promoters of open administration. As part of this process several important spatial and non-spatial data resources have been opened for public use. One of the most used resources has been the public transportation Route Planner service (Reittiopas), provided by Helsinki Regional Transport Authority (HSL). The route and schedule database underlying Route Planner is accessible via an application programming interface (API) and through a dump file in Kalkati.net XML format. This type of data has proven to be useful not only for application developers but also for researcher and urban planners interested in assessing the spatial patterns of public transport accessibility.

The MetropAccess-project (University of Helsinki, Department of Geosciences and Geography) aims at developing comparable tools for analysing spatial patterns of accessibility and travel time using different travel modes. In this paper, we present one of our key analysis tools, Reititin, which is developed in collaboration with BusFaster Ltd. The tool calculates optimal public transportation travel chains for extensive sets of origin-destination pairs using the Kalkati.net XML dump provided by HSL and OpenStreetMap data.

2 Technology

The tool is written in the JavaScript language to work on all common operating systems and with the goal to embed it on a website, allowing any interested person to make route calculations en masse. Large analyses with thousands of routing calculations using different combinations of starting locations, times, enabled transport modes and other options can take hours to complete. It would be impractical to allow free unrestricted access for everyone to execute such queries on our servers, while we'd also like to allow casual users to make analyses for example to select a hotel with good connections to specific locations. Users are unlikely to install software on their own computer just to see if the results would be useful to them. If the calculations are instead performed in a web browser, the entire service can be provided through any low-cost or free online file storage service. Then there is practically no limit to the number of concurrent users and each user is able to run analyses as complicated as their own computer is able to handle. Results can be shared online together with selected analysis options encoded in a link to the service, allowing the analysis to be modified and repeated by others.

The web browser environment is constrained in terms of memory and calculation speed, and the entire street network and public transport schedule must be transferred to the user for batch routing. For this, data files must be kept small to reduce the downloading time.

2.1 Algorithms

Schedules are compressed by identifying redundancies between different days of the week, delta coding and using the LZ77 algorithm (Ziv & Lempel 1977). This compression shrinks both memory usage and data file size. The road map is simplified and topology stored together with delta coded geometry by referring to existing road points when a road intersects another previously stored road. This processing can reduce Kalkati or Google Transit Feed Specification format to under 2% and OpenStreetMap pedestrian routing maps to under 20% of original compressed size without affecting routing results. This process lasts a few minutes for the Helsinki metropolitan region.

In the routing, optimal single source shortest paths are calculated using Dijkstra's algorithm (Dijkstra 1959) modified to track time of the day in addition to the cost being optimised. The road network for pedestrian routing and public transit route network are connected at public transit stops, which are conceptually a part of the road network. Points along public transport routes passing by the stop are all considered separate locations.

Cost of moving from a routing graph node representing the sidewalk near a bus stop to another node representing a bus at the stop depends on the waiting time, which can be computed only when the pedestrian's earliest possible arrival time at the stop is known. Therefore many graph edge costs are computed only when the best route to their start point is found. Instead of storing locations in the priority queue used by Dijkstra's algorithm, we store objects encapsulating several types of information.

Events such as arriving at an intersection or stop are represented as JavaScript objects representing a visit at a location and containing the event's cost, time, location in the routing graph, previous location and a visit method. Routing is based on traversing the priority queue and calling visit methods of events found. Costs and arrival directions for all locations are kept in a separate structure without modifying the routing graph allowing multiple parallel route searches. Different modes of transport are represented by subclasses of visitor objects making them easy to modify independently of each other.

In object oriented programming terms, there are classes for visitors for transit vehicles, transit stops and road network nodes. Adding a new type of transportation is relatively simple. For example a bicycle sharing system could be represented by adding a new type of road node visitor that does not allow entering bus stops, has faster connections between nodes and is connected to the regular road network only at bike lending points.

Routes are optimised for cost represented in increments of 100 milliseconds spent sitting inside. The cost of walking is then typically set higher than actual elapsed time due to the effort spent and exposure to the weather. Transfers between public transport lines are penalised by 3 minutes in addition to actual waiting time because they add uncertainty and risk to the user if a vehicle is delayed or arrives too early, or the user makes a mistake while trying to find the correct location on time.

The priority queue in Dijkstra's algorithm is simply an array with a slot for each possible time step, which can contain one or more events stored in a linked list. If maximum route cost is 5 hours, array length in 100 millisecond increments is 180 000 items regardless of road network size. Complexity of the algorithm is O(V+E+T) where V is the number of road intersections or public transport stops, E the number of connections between them and T the maximum route cost. The discrete time steps speed up calculations for large cities while causing an average error of 50 milliseconds at every intersection, which is negligible compared to other random factors such as traffic lights.

2.2 Output

To produce Pareto optimal routes in large batches, the calculation is repeated with several initial departure times. The latest possible departure time still giving the earliest possible arrival is then found for each origin-destination pair. This way producing an OD matrix takes a very small constant lookup time per element and routing time scales linearly with road network intersections, number of origins and different departure times tested. Calculating routes from a single origin and departure time to any number of locations in the Helsinki metropolitan area takes about 3 seconds in a web browser.

As an output, Reititin produces detailed travel chain descriptions between the origin and destination coordinates. The descriptions include information on travel times and distances for each mode of public transport and necessary walking stretches. Also all stops and lines are listed. The tool enables creation of new public transportation routes and routing along them. This feature can be used to analyse alternative transport scenarios for future, and testing the effect of potential new public transport routes. It is also possible to conduct temporal analyses by comparing travel time patterns for current and older routes and schedules.

3 Conclusion

The outputs of routing analyses can be and have been used for various purposes. The basic outputs – travel time matrices - enable detailed analyses of service accessibility (Jäppinen et al. 2013, Salonen et al. 2012, Salonen & Toivonen, 2013) and its temporal variation (Saarsalmi 2014). Furthermore, the detailed travel chain descriptions make it possible to assess theoretical CO_2 emissions for each route (Lahtinen et al. 2013).

In order to make the most use of the positive trend of open data policies, it is essential that tools that are created to handle the data sources are also open access. Reititin provides an example of such development and it will hopefully provide a useful tool for urban planners of the region.

References

Bertolini, L., le Clercq, F. & Kapoen, L. (2005), 'Sustainable accessibility: a conceptual framework to integrate transport and land use plan-making. Two test applications in the Netherlands and a reflection on the way forward', *Transport Policy* **12** (3), 207–220.

Dijkstra, E. (1959), 'A note on two problems in connexion with graphs', *Numerische Mathematik* **1**, 269–271.

Frank, L., Bradley, M., Kavage, S., Chapman, J. & Lawton, T.K. (2008), 'Urban form, travel time, and cost relationships with tour complexity and mode choice', *Transportation* **35** (1), 37–54.

Jäppinen, S., Toivonen, T. & Salonen, M. (2013), 'Modelling the potential effect of shared bicycles on public transport travel times in Greater Helsinki: An open data approach', *Applied Geography* **43**, 13-24.

Lahtinen, J., Salonen, M. & Toivonen, T. (2013), 'Facility allocation strategies and the sustainability of service delivery: Modelling library patronage patterns and their related CO2-emissions', *Applied Geography* **44**, 43-52.

Mavoa, S., Witten, K., McCreanor, T. & O'Sullivan, D. (2012), 'GIS based destination accessibility via public transit and walking in Auckland, New Zealand', *Journal of Transport Geography* **20** (1), 15–22.

Saarsalmi, P. (2014), 'Päivittäistavarakaupan spatio-temporaalinen saavutettavuus pääkaupunkiseudulla', Pro gradu -tutkielma, Helsingin yliopisto, Geotieteiden ja maantieteen laitos.

Salonen, M., Toivonen, T. & Vaattovaara, M. (2012), 'Arkiliikkumisen vaihtoehdoista monikeskuksistuvassa metropolissa: Kaksi näkökulmaa palvelujen saavutettavuuteen pääkaupunkiseudulla', *Yhdyskuntasuunnittelu* 2, 50, 8-27.

Salonen, M. & Toivonen, T. (2013), 'Modelling travel time in urban networks: comparable measures for private car and public transport', *Journal of Transport Geography* **31**, 143–153.

Silva, C. & Pinho, P. (2010), 'The structural accessibility layer (SAL): revealing how urban structure constrains travel choice', *Environment and Planning* A 42 (11), 2735–2752.

Ziv, J. & Lempel, A. (1977), 'A universal algorithm for sequential data compression', *IEEE Transactions on Information Theory* **23**, 337–343.