

osmprob

A probabilistic routing engine for R

#DESCRIPTION
Package
Type: Calculate probabilistic routes
Title: Calculate probabilistic routes
Version: 0.0.1
Authors: Andreas Petutschnig
Mark Padgham
Description: Calculates probabilistic routes on a OSM street graph and displays the resulting probabilities in a Shiny/Leaflet web app

Introduction

Osmprob is an R package that allows users to calculate traversal probabilities and the estimated travel distance between two points on a graph. It is therefore a probabilistic counterpart to deterministic least-cost algorithms like Dijkstra's algorithm. Besides the routing process, it takes care of downloading and preprocessing the necessary OpenStreetMap (OSM) data from the area of interest using the `osmdata` package written by Padgham (2007). The routing is performed based on the methods introduced in Saerens et al. (2009). The routing results give an indication about movement on the road graph and can either be displayed with the built-in web mapping functionality or used for further analysis. Possible use cases for probabilistic routing include many fields related to transportation and city planning. This package was part of a master thesis at the department of geoinformatics at the University of Salzburg in Austria supervised by Mark Padgham.

Data preprocessing

In order to avoid unsolvable routing problems and to speed up the routing process, parts of the road graph not needed to maintain the original topology are removed from the raw OSM data. Figure 1 illustrates the necessary steps for this to be done. First, the largest component of the graph is identified and the other ones are removed. This must be done because routing on a graph is only possible if the start and end vertices can be accessed through intermediate edges. Then, all vertices that have exactly two neighbouring vertices are removed. Removing these intermediate vertices results in a smaller graph and thus better performance. Finally, the edges in between the remaining vertices are replaced by the respective direct connections so the graph can be used for routing again. To ensure high performance, the computationally expensive parts of the preprocessing routine were implemented in C++ using the Rcpp package by Eddelbuettel and François (2011).

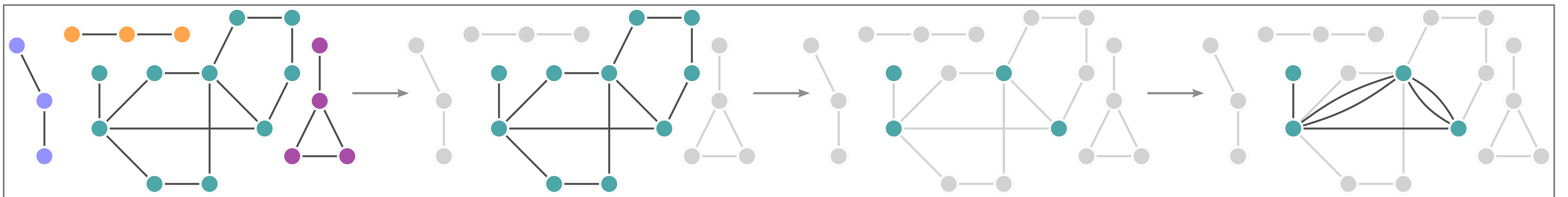


Figure 1: Preprocessing steps from multiple raw graphs to a single minimal graph.

Routing results

The results of the routing engine are the probabilities with which every edge is traversed on the way between two points. The screenshot in Figure 2 illustrates this by indicating the traversal probabilities through line thickness on a road graph. For comparison, the least cost path in between is overlayed in red. Looking at the lines close to the ideal path shows how probabilistic routing compares here. Edges reasonable close to the ideal path are still very likely to be used, however the larger a detour an edge would mean, the less likely it is to be traversed. The calculated probabilities in combination with the edge lengths can be used to estimate how far the average travel distance between the two points is. The degree of deviation from the shortest path is defined by a factor that can be passed to the routing function. By tweaking it, different ways of movement behaviour can be modeled.

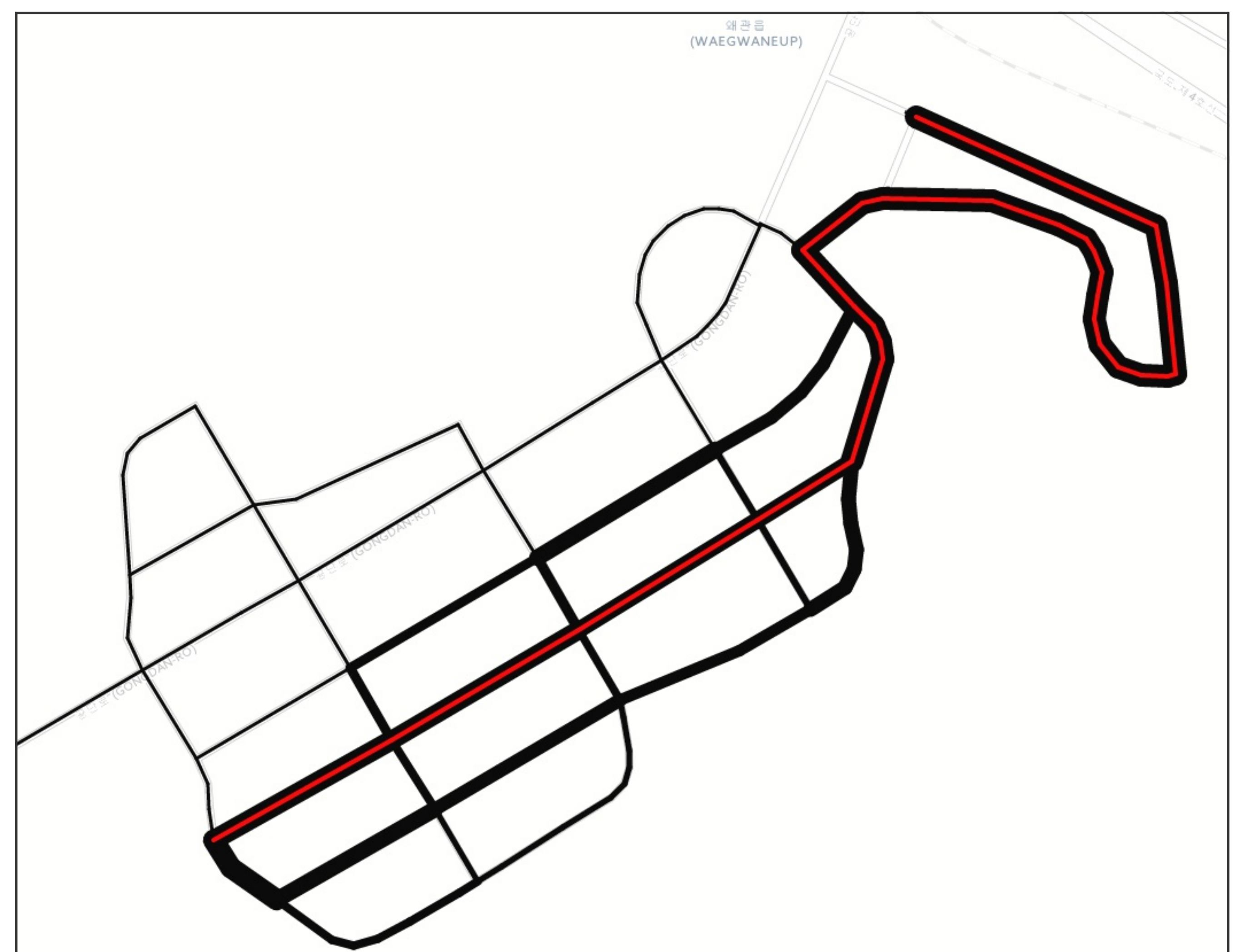


Figure 2: Road network with traversal probabilities indicated by line thickness

Discussion and outlook

Because estimations of realistic travel distances are needed for many applications but hard to acquire, `osmprob` can be useful for lots of use cases in city planning, transportation and similar fields. As the package not only takes care of the routing, but also the data acquisition and preprocessing, it can be easily facilitated by inexperienced users. Test runs using this package were successful so far, however, there are still some open questions regarding parametrisation of the routing function as well as some cartographic and technical details about the visualisation. As soon as these issues are fixed, the package will be finalised and submitted to CRAN.

Fork me.

If you want to try out `osmprob`, report issues or help improving it, visit the project site on osm-router.github.io/osmprob or find the source code on github.com/osm-router/osmprob.

References:
Eddelbuettel, Dirk, and Romain François. 2011. "Rcpp: Seamless R and C++ Integration." *Journal of Statistical Software* 40 (8): 1–18. <http://www.jstatsoft.org/v40/i08/>.
Padgham, Mark. 2017. *osmdata: Import Openstreetmap Data as Simple Features or Spatial Objects*. <https://cran.r-project.org/package=osmdata>.
Saerens, Marco, Youssef Achbany, François Fouss, and Luh Yen. 2009. "Randomized Shortest-Path Problems: Two Related Models." *Neural Computation* 21 (8). MIT Press: 2363–2404.

