Road Network Hierarchy Generation by Distributed Agents for a Routing Application

Joris Maervoet [a,b] Lode Blomme [c] Katja Verbeeck [a]
Greet Vanden Berghe [a]

[a] *CODeS, Vakgroep IT, KaHo Sint-Lieven, G. Desmetstraat 1, 9000 Gent*
[b] *CODeS, K.U.Leuven Campus Kortrijk, E. Sabbelaan 53, 8500 Kortrijk*
[c] *RouteYou, Kerkstraat 108, 9050 Gent*

Abstract

Scalability is a major issue in finding the shortest path in a transportation network graph. Therefore, many contemporary routing web applications make use of hierarchical heuristics. This means that the road class (highway, national road, provincial road ...) of edges is used to crop the search space drastically. However, for wayfinding in network graphs for leisure and tourism, the existing road classes cannot serve as an efficient hierarchical guidance. This demonstration shows the automatic generation of hierarchical levels for this type of networks and its effect on point-to-point routing performance. The generation is realised in a bottom-up manner by distributed agents, representing geographical cells.

1 Introduction

The company. The company RouteYou manages a web 2.0 environment which enables users to create, share and use tourist routes in an interactive way. Besides, it offers a routing platform for various application developers and digital content providers. Most of these processes require a performant calculation of the path with the lowest cost from start node to end node in a road topology.

Hierarchical routing. In transportation network graphs, the edge weights represent travel time. In order to find the (fastest) path with the lowest cost, it is common practice to organise the road network in road classes (levels) for improving performance [1]. These road classes are easy to obtain and geographic content providers have a long tradition in providing them. Hierarchical routing involves that long distance subpaths result from the application of classical shortest path algorithms on subnetworks of higher importance (level). It is effective when a road class is a good indicator for the actual speed over the subnetwork, and when higher level subnetworks are sparser but nevertheless fully connected. Hierarchical routing results in a slight decrease in solution quality (i.e. nearly shortest paths), whereas computational time and memory footprint decrease drastically, which is vital in a web application.

Routing objectives for leisure and tourism. When considering wayfinding for subdomains of leisure and tourism, such as nordic walking or hiking, the company uses a graph in which an edge's weight is calculated as the edge distance, multiplied by a resistance. This resistance represents a domain-specific unlikeliness and is always ≥ 1. It originates from geographical environmental conditions and from the company's social network. In order to support hierarchical routing for these network graphs, a network hierarchy different from the content provider's routing class hierarchy is required.
This demonstration presents

- a technique for the automatic construction of a hierarchy for network graphs for leisure and tourism,
- an evaluation of the generated hierarchy, based on performance criteria of a hierarchical routing application used by the company. This application uses the multi-level heuristic node promotion algorithm.

2 Approach

Heuristic node promotion. Jagadeesh et al. [2] compared and evaluated several techniques for hierarchical two-level routing in transportation road networks. They attained best results (computation time vs. result quality) with the heuristic node promotion algorithm, of which an adapted multi-level version has been implemented for RouteYou. This algorithm assumes the availability of cells for each level L i.e. the polygons enclosed between planar graph edges of level $\geq L$, and, of transition points for each of these cells of level L with the rest of the network of level $\geq L$. The recursive algorithm first determines its level of execution E as the highest level for which start and end node are not located in contiguous or same cells. The edges of level $\geq E$ are loaded. Next, when $E > 0$, a set of virtual edges, from the start node to its surrounding cell's transition points and from the end node's cell transition points to the end node, is created and promoted to the network. The weights of the virtual edges are estimated using straight line distance and average resistance. Next, classical routing is applied from start to end node. Each virtual link in the resulting path, is solved by a recursive call to this algorithm.
This algorithm performs well when the following requirements of the hierarchy are met:

- The higher the level, the higher pct. of low resistance weight links it contains. (Solution quality.)
- Each subnetwork of edges of class $\geq L$ is fully connected. (Robustness.)
- After 2-valid reduction, a cell of level L contains a maximum of 1000 links of level $L - 1$. (Computation time and memory footprint.)
- The ratio of the amount of higher level links to the amount of level 0 links should be kept low. (Scalability; computation time and memory footprint.)

Hierarchy generation. The network hierarchy is generated by a distributed multi-agent application. In the first iteration, all edges of level 1 are labelled. It starts from the planar graph of the road network of the area of interest. Initially, an agent is assigned to each of the atomar planes (cells) enclosed between the edges of the planar graph. Next, the agents can negotiate with neighbouring agents in order to merge their cells. Each agent cell can be evaluated by a goal function, and a merge can take place when this function improves for both agents. The goal calculation is based on: the average resistance of the cell border edge weights and the internal edge weights, and, the number of edges enclosed. The iteration ends when none of the agents is able to improve its goal. At this stage, all edges that are located on cell borders, are promoted to level 1.
At the start of each of the next iterations, the goal function is relaxed w.r.t. the number of edges enclosed and tightened w.r.t. the average cell border resistance. Next, the negotiation and merge iteration is resumed, until no further improvement occurs. Finally, the cell border edge levels are increased by 1.

3 Demonstration and specifications

The demonstration starts by a film showing the hierarchical network generation for motorcycle routes by distributed agents. It consists of snapshots of the spatial cell distribution during several steps of the generation process. It shows the influence of the goal function on the collective behaviour of the system.
Next, visitors are offered to plan long-distance point-to-point routes over this network, using an interactive web page. They are able to compare the planned route and its costs, the trace of routing levels, calculation time and memory footprint for both a flat and a hierarchical Dijkstra implementation.
The system consists of a hierarchical level generation component, a data preprocessing component, and a hierarchical routing component, interacting with a spatial database. It has been developed by Joris Maervoet, using Aptana (IDE), PHP, PostGIS SQL, Google Mashup and Quantum GIS.

Acknowledgements. This demonstration is part of the results of the industrial PhD project 'Structural heuristics for performant personalised routes', organised by the company RouteYou and partially funded by the IWT (090726).

References

[1] L. Fu, D. Sun, and L. R. Rilett. Heuristic shortest path algorithms for transportation applications: state of the art. *Computers and Operations Research*, 33(11):3324–3343, 2006.

[2] G. R. Jagadeesh, T. Srikanthan, and K. H. Quek. Heuristic techniques for accelerating hierarchical routing on road networks. *IEEE Transactions on Intelligent Transportation Systems*, 3(4):301–309, 2002.