Algorithm Engineering Group
Prof. Dr. Tobias Friedrich
Bachelor Project 2016/17

HPI Hasso Plattner Institut
IT Systems Engineering
Universität Potsdam

TOMTOM

Efficient Shortest Paths on Portable Devices

Motivation The road network graph of Europe contains more than 100 million nodes and half a billion edges. Performing route searches on such a graph in a setup with limited memory, like a Personal Navigation Device or a mobile phone, is a non-trivial task. For this purpose, modern maps are partitioned and stored in blocks (map tiles) according to a geometrical grid. Therefore, the major factor during the computation is the number of blocks of graph data loaded into memory. An A* search algorithm with strong lower bounds for shortest path distances is important to optimize the number of cache loads.

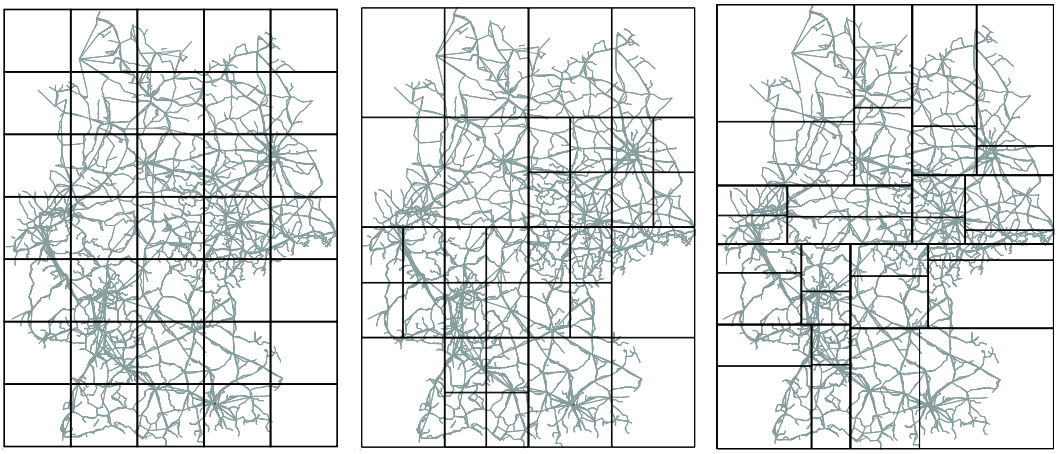

Figure 1: Germany with three different partitions. (Pictures from Köhler et al. [9th DIMACS Implementation Challenge])

Precomputed lower bounds for shortest paths between two tiles can be used to modify the order in which nodes are settled and further speed up Dijkstra-like searches. The right use of these properties will help to speed up route search on embedded devices and allow to build applications with a lower memory and energy footprint compared to the ones available today.

Theory A portable device can be described in the Aggarwal-Vitter external memory model with a cache of size M, which stores blocks of size B. The runtime is measured in the number of blocks loaded into the cache during a computation. To analyze the number of block load operations for one Dijkstra search, we consider a sequence of settled nodes $\langle v_1, v_2, \ldots, v_\ell \rangle$, which correspond to a sequence of blocks from which data is accessed $\langle b_1, b_2, \ldots, b_\ell \rangle$. Loading a block more than once is costly and should be avoided. Repeated entries in $\langle b_1, b_2, \ldots, b_\ell \rangle$ should therefore have distance at most $\lfloor M/B \rfloor$.

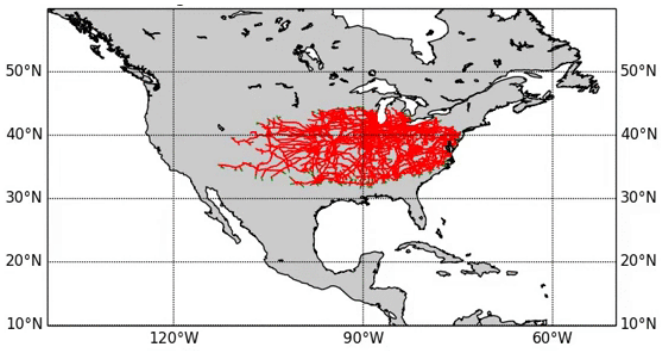

Figure 2: Ongoing A* search from Washington D.C. to Los Angeles.

For the purpose of reducing the worst-case complexity, properties of the underlying graph class like the edge weights, the partitioning, and the lower bound function should be studied in the theoretical part.

Practice In the experimental part of the project, modifications to Dijkstra's algorithm will be implemented and evaluated. The first scenario aims at finding implementations that minimize the size of a cache which is needed to find all shortest s-t-paths without loading the same block twice during one search. In the second scenario, a fixed cache size is given, and the task is to minimize the number of block loads for all searches. The experiments will also cover different caching strategies.

Different modifications of Dijkstra's algorithm are planned for both scenarios. Such modifications could, for example, consider settling nodes from the priority queue before they have minimal cost even if that means that they have to be settled again later. Furthermore, changes to the A^* costs should be investigated. For instance, these changes could consider the lower bounds of shortest paths between partitions or the fact that a new partition has to be accessed in order to settle a node.

For this part, a routable map of Europe and a graph that partitions the nodes based on a geometrical grid will be provided by TomTom Development Berlin. This also includes the length and travel time for each edge to allow for different cost models like fastest or shortest. A set of origin and destination pairs of real customer requests serves as the test instance.

Contact

Tobias Friedrich, HPI
Martin Krejca, HPI
Gregor Lagodzinski, HPI
Ralf Rothenberger, HPI

Jan-Ole Sasse, TomTom
Sven Grothklags, TomTom
Henning Hasemann, TomTom
Alexander Kröller, TomTom