Using Contraction Hierarchies to Find Dissimilar Paths in Transportation Networks

S. Demeyer, M. Houbraken, T. Walcarius,
P. Audenaert, D. Colle, M. Pickavet, P. Demeester
Ghent University, Department of Information Technology (INTEC-IBCN)
sofie.demeyer@intec.ugent.be

Current route planning systems find the shortest path between an origin and a destination in only a fraction of a second. Users, however, are not only interested in the shortest path, but want a set of good alternative routes. This way, information that is not available in the routing application, such as toll and scenery information, still can be taken into account. These alternative routes are only of interest if they are dissimilar. This means that they should have a minimal amount of overlap. Moreover, the paths should not contain cycles and should be (T-)locally optimal. A path is called (T-)locally optimal if and only if all subpaths (of length T) are shortest paths themselves. In this research, we present a novel method to find this set of dissimilar paths between a single origin and a single destination in a transportation network. The main advantage of this algorithm is that it finds a set of dissimilar paths in only a fraction of a second, which is faster than the current known solutions.

The problem of finding dissimilar paths was already addressed by Vanhove [1]. The concept of plateaus is introduced to find locally optimal and dissimilar paths with a bidirectional algorithm. A plateau is defined as a subpath that is present in both the forward and the backward search tree. It is demonstrated that plateaus of length $> T$ are T-locally optimal. The network is searched bidirectionally and subsequently all plateaus are identified, resulting in a set of paths. From this set, all paths that are not locally optimal or that contain cycles are pruned, resulting in a set of dissimilar path that meet the prerequisites (i.e. they are locally optimal and contain no cycles). Experiments in the Belgian road network show that the execution times of this algorithm are in the seconds range.

Multiple techniques have been presented to speed up shortest path calculations in transportation networks. One of the most promising techniques is called contraction hierarchies [2]. In a preprocessing step, the network is contracted, i.e. shortcut links are added between non-neighboring nodes. Moreover, nodes are ordered in the way they are contracted. To find the shortest path between an origin and a destination in a contraction hierarchy, a bidirectional search is executed in which nodes are only visited if they have higher contraction ids. This way, only a fraction of the links are investigated, speeding up the calculations tremendously. Moreover, it can be proven that this algorithm is guaranteed to still produce the optimal solution.

In this research, we will combine the algorithm to find dissimilar paths with the theory of contraction hierarchies. Both the original dissimilar shortest path algorithm and the contraction hierarchy algorithm search the network bidirectionally. Moreover, when a contraction hierarchy is built, nodes are bypassed. This means that small detours are eliminated early on, resulting in locally optimal paths. After a set of paths has been constructed, this set needs to be filtered so that only the useful paths remain.

The algorithm works as follows. The contraction hierarchy (network) is searched bidirectionally. Each time a node has been settled in both search processes a path is constructed and added to the solution set. By making use of contraction hierarchies, this set of paths can be constructed very fast. However, not all paths of the solution sets are feasible solutions, which means that postprocessing steps are needed. The algorithm should return a set of dissimilar paths. These paths should however still be valuable solutions, which means that only those paths that do not differ too much from the optimal one should be taken into account. We opted to eliminate all paths that are longer than α times the cost of the optimal path, with $\alpha > 1$. Furthermore, paths that contain cycles should be eliminated as they are of no use in a transportation environment. In the last postprocessing step, all remaining paths are compared with one another. For each pair of paths (P_i and P_j) a dissimilarity value $D(P_i, P_j)$ is calculated as follows [1]:

$$D(P_i, P_j) = 1 - \frac{\frac{w_s(P_i,P_j)}{w(P_i)} + \frac{w_s(P_i,P_j)}{w(P_j)}}{2}$$

with $w_s(P_i, P_j)$ the weight of the overlapping parts of P_i and P_j, and $w(P_i)$ the weight of path P_i. The set of paths for which the sum of the dissimilarity values is maximal forms the result of the algorithm.

For the experiments, we made use of a Belgian road network with 349 810 nodes and 847 309 unidirectional links. From this a contraction hierarchy was built, resulting in 904 330 additional links. The algorithm was implemented in Java. Experiments have shown that the resulting paths are dependent on how the hierarchy was constructed, meaning that they might differ from the 'optimal' ones. However, still valuable paths are returned. Concerning the calculation time, it should be noted that the first step of the algorithm can be executed in only a couple of milliseconds. It is demonstrated that the shortest path algorithm can be aborted early on, i.e. after only tens of paths have been exported. As this preliminary list of paths is relatively small, the postprocessing steps do not consume much calculation time. This means that a set of dissimilar paths can be constructed in only a couple of milliseconds.

References

[1] Vanhove S, Alternative Routing Algorithms for Road Networks (Dissertation), Ghent University, Department of Applied Mathematics and Informatics (2012)

[2] Geisberger R et al., Exact Routing in Large Road Networks Using Contraction Hierarchies, Transportation Science, 46, 388-404 (2012)