Efficiently updating Customizable Contraction Hierarchies

1 Intro

This is a seminar on "Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees" [4]. The paper is about finding the shortest path in road networks. By *shortest* it is meant, the path that requires the less time to get from source s to a target t. The route network is modeled as a directed graph $G(V, E)$ where each street crossing represents a vertex $v \epsilon V$ and each road between crossings represents an edge $e \epsilon E$. The most basic and solid method to find shortest paths between vertices in a graph is Dijkstra's algorithm [2]. This algorithm is proofed to always return the correct shortest path but it is not fast enough for just in time route planning on large road networks as we know it from services like Google Maps.
There are many different approaches that try to speed up shortest path queries by precomputing any different kind of index structure before doing the shortest path query. The index structure discussed in "Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees" [4] is CH (Contraction Hierarchies)[3] with some extension. This extension is CCH (Customizable Contraction Hierarchies) [1]. Although the authors of "Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees" never mention the term CCH their approaches builds the same index structure. This is a pity, as for this part they kind of reinvent the wheel.
The difference lies in updating the CCH index structure. For updating the CCH, the authors of "Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees" [4] will use yet another index structure called *SS-Graph* that helps to exactly identify the shortcuts that have to be updated, after some edge weight has changed.
This works purpose is to show that the update procedure using the *SS-Graph* described in "Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees" [4] can be taken as an extension to CCH.
In todays implementation, the whole index structure is recomputed periodically. This is an valid approach as it is fast enough to stay accurate for route planning in road networks.
In "Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees" [4] the authors would like to find a way that make it possible to handle streaming updates. Therefore it is necessary to handle single weight decreases and increases. To do so another index structure the *SS-Graph* is introduced. This index structure is a helper structure to identify the shortcuts that have to be updated in the CCH.
The disadvantage of this *SS-Graph* is that the overall space consumption rises. This can be a deal breaker for large networks. Finally they introduce a way to create the necessary *SS-Graph* on the fly. Which is only a part of the whole structure. Sadly the exact way, how, is missing in the paper "Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees" [4].

References

[1] Julian Dibbelt, Ben Strasser, and Dorothea Wagner. Erratum: Customizable contraction hierarchies. In *Experimental Algorithms*, pages E1–E1. Springer International Publishing, 2014.

[2] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, dec 1959.

[3] Robert Geisberger, Peter Sanders, Dominik Schultes, and Christian Vetter. Exact routing in large road networks using contraction hierarchies. *Transportation Science*, 46(3):388–404, aug 2012.

[4] Dian Ouyang, Long Yuan, Lu Qin, Lijun Chang, Ying Zhang, and Xuemin Lin. Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees. *Proceedings of the VLDB Endowment*, 13(5):602–615, jan 2020.