Survey on Contraction Hierarchies Algorithm

Prashant Singh
Department of Computer Science and Engineering
National Institute of Technology Manipur
Imphal, India
prashant.singh1995@outlook.com

Abhigya Pranshu
Department of Computer Science and Engineering
National Institute of Technology Manipur
Imphal, India
pranshuabhigya002@gmail.com

***Abstract*—Efficient route planning algorithms has always been an area of great interest. One of the most popular algorithms in this area primarily based on node contraction is contraction hierarchy. We provide a survey on research done on this algorithm which includes papers on this algorithm. We also provide analysis of scenarios in which these variants of contraction hierarchies will be more suitable such as faster query time , space overhead, preprocessing overhead and other similar requirements.**

***Keywords*—Contraction Hierarchies, Time Dependent Contraction Hierarchies, Customized Contraction Hierarchies, Distributed Contraction Hierarchies**

I. Introduction

Path Finding algorithms have great applications in navigation, traffic simulations and game applications. [13]. Dijkstra algorithm [6] solved the problem of finding the shortest path but the query time was very large. After this many goal directed algorithms such as A* [9] and its variants like A* with landmarks [10] have produced better query times but have huge space overhead like Dijkstra.
Hierarchical based algorithms usually solves routing problems providing a hierarchy based structure such as Highway road routing.

Highway node routing(HNR)[12] is a multilevel routing scheme where we compute shortest path distances of subset of "important" nodes known as highway nodes. It initially computes the hierarchy of highway-node sets. Starting with the nodes at the lowest hierarchical level it keeps on computing the shortcuts till it reaches the highest level node.

Geisberger et al. [7] proposed a new algorithm known as contraction hierarchy. This algorithm is an extreme case of HNR where each node has its own level of hierarchy.

II. Survey on Contraction Hierarchies

A. Overview

Many algorithms add new edges in graph known as "shortcuts" to speed up query time even in large road networks. Though optimal shortcut selection has been proven to be NP-hard by [4, Bauer et al. 2012]. Geisberger et al. [2008,2012] proposed a fast speed up technique for large road networks using this shortcut addition and node contraction.

B. Contraction Hierarchies

[7]Contraction Hierarchies are mainly based on the concept of node ordering and node contraction. Initially nodes are classified according to their importance. This greatly determines the efficiency of this algorithm. For contraction of a node v only nodes at a higher hierarchical level than v need to be known. Thus nodes are contracted iteratively starting with the the node at the lowest hierarchical level. Here the selection of the next node is decided by using a queue based on priority of nodes known as priority queue. Using various priority terms it estimates contraction of which node will lead to a better result. These priority terms are updated upon the contraction of each node.

The author has proposed various methods to decide the node ordering like lazy update,edge difference,cost associated with node's contraction etc. Out of all these methods edge difference is the most useful parameter. In this method node ordering is computed by calculating the difference between edges incident on the node and shortcuts that need to be added due to its contraction.

[7]Contracting a node means removing it from the graph. conceptually. During node contractions the shortcut edge (u,w) for contraction of a node v is added only if $< u, v, w >$ is the shortest path while traversing from u to w. No shortcuts need to be added if there is an alternate shortest path from u to w without using the path via v. As only few shortcut edges are added as compared to the edges removed on contraction of the nodes ,algorithm requires negative space overhead which was a major drawback with earlier algorithms like Dijkstra and other goal directed algorithms like A* search.

The paper [1] suggests in order to query for a path using contraction hierarchy the author has proposed a slight extension to the concept of Dijkstra's algorithm known as bidirectional Dijkstra. This uses the fact that for every pair of nodes reachable from one another there exists a shortest up-down path such that from one the path goes only up in the hierarchy and from the other goes only higher in the hierarchy. The algorithm terminates only when the paths meet at a node.

C. Other variants of CHs

In real life scenarios, in contrast to time independent approach of CHs in [7], the edge weights of road network may change according to time, depending on some factors such as traffic congestion. One such paper on time dependent edges

is [2] which proposes time dependent CH. They implement a generalization of CH proposed by [7] to extend support of time dependent edges. In this paper, authors assume a FIFO property and look into earliest arrival problem. The author uses bidirectional Dijkstra's algorithm for query and expresses challenges faced in backward search, as arrival time is unknown at time of search. But they are able to implement the search because of acyclic graph generated by CH that reduces the search space.

Another paper on time dependent CH is [3] which suggests approximation to time dependent contraction hierarchies in [2]. The authors discuss the problem of space reduction and efficient reduction of travel time profile in [2].

The authors have suggested Min-Max TCHs, which is an extreme case of approximated TCH. It consumes even less memory than Approximated TCHs. The authors describe a corridor which is a subgraph of whole road networks containing start node and a reachable node.

The paper suggests a time travel function which represents time needed to reach a node from start node with a edge between the nodes at a particular departure time. The function follows FIFO property.

The earliest arrival queries are computed with both exact Time Dependent Contraction Hierarchies (TCHs) and the suggested approximated TCHs. The authors have claimed that by approximating piecewise linear functions used to defined the weights especially in subset of useful roads known as corridor, they can address the initial issues without undermining the accuracy of results

The latest paper in our survey [5], discusses restriction of Contraction Hierarchies as being edge weight dependent. The authors have proposed the gaming application of CHs, but expressed the need of the approach being edge independent in the pre-processing phase of CH.

The authors have reiterated that contraction hierachies irrespective of sub optimal node ordering can produce good results.

The author suggest that their implementation of CH known as Customizable contraction hierarchies based on principle of divide and conquer is practical and feasible. The author claim to provide better query time when less well defined metrics are taken into account such as distance.

The authors have claimed that their basic variant also produces faster customization run time. The authors unlike the previous implementation of CH have also suggested second method that is an elimination tree based query which in constrast to Dijkstra's algorithm and does not require priority queue as [7]

There have been efforts to reduce the preprocessing time taken in time dependent CH as in [2]. The paper [14] implements a parallelized version of Contraction Hierarchies by using multiple cores in the CPU. The author has stated a significant reduction in preprocessing time by paralleling node ordering in multiple cores.

In this day and age, distributed computing is a becoming a de facto. The advantage of distributed systems are many being servers can be added or reduced with requirements. In the paper [11] authors have suggested distributed time dependent variant of contraction hierarchies instead of parallel multiple core operation approach at one CPU only as in [14].

The authors suggest partitioning graph into n partitions where n is the number of processes and each partition is sent to each process. The processes contract the node independently of each other.The node contraction and ordering algorithm are iterative in nature. The authors claim that the overall result would not be affected by the order in which nodes are contracted. Parallelism is achieved using shared memory. The authors have claimed a huge reduction in preprocessing phase of CH and have suggested a distributed query algorithm but is limited to interconnected speed between systems.

III. Summary

TABLE I
Comparisons of variants of Contraction Hierarchies Algorithm

Year	Title	Contribution
2008	Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks. [7]	Introduced Contraction Hierarchy (CH) algorithm for shortest path routing in road networks
2009	Time-dependent contraction hierarchies. [2]	Provided Generalization of Contraction Hierachies to support time dependent edges
2009	Parallel time-dependent contraction hierarchies. [14]	Implemented parallel multi core approach in one system and reduced preprocessing time in TCH.
2010	Time-dependent contraction hierarchies and approximation. [3]	Provided approximations on TCH to reduce space and travel time profile without significantly affecting results
2010	Distributed time-dependent contraction hierarchies. [11]	Implemented TCH on Distributed Systems and provided significant improvement in preprocessing time.
2012	Exact routing in large road networks using contraction hierarchies. [8]	Conducted comparison with other routing algorithms and implemented exact route planning on mobile.
2013	Contraction hierarchies on grid graphs. [13]	Implemented Contraction Hierarchies on grid graphs and provided improvements for query time in CH for computer games map.
2016	Customizable contraction hierarchies. [5]	Provided Elimination tree based query and implemented customization of CH on principle of divide and conquer

References

[1] G Veit Batz et al. "Minimum time-dependent travel times with contraction hierarchies". In: *Journal of Experimental Algorithmics (JEA)* 18 (2013), pp. 1–4.

[2] G Veit Batz et al. "Time-dependent contraction hierarchies". In: *Proceedings of the Meeting on Algorithm Engineering & Expermiments*. Society for Industrial and Applied Mathematics. 2009, pp. 97–105.

[3] Gernot Veit Batz et al. "Time-dependent contraction hierarchies and approximation". In: *International Symposium on Experimental Algorithms*. Springer. 2010, pp. 166–177.

[4] Reinhard Bauer et al. "The shortcut problem-complexity and algorithms". In: *J. Graph Algorithms Appl.* 16.2 (2012), pp. 447–481.

[5] Julian Dibbelt, Ben Strasser, and Dorothea Wagner. "Customizable contraction hierarchies". In: *Journal of Experimental Algorithmics (JEA)* 21.1 (2016), pp. 1–5.

[6] Edsger W Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[7] Robert Geisberger et al. "Contraction hierarchies: Faster and simpler hierarchical routing in road networks". In: *International Workshop on Experimental and Efficient Algorithms*. Springer. 2008, pp. 319–333.

[8] Robert Geisberger et al. "Exact routing in large road networks using contraction hierarchies". In: *Transportation Science* 46.3 (2012), pp. 388–404.

[9] Andrew V Goldberg and Chris Harrelson. "Computing the shortest path: A search meets graph theory". In: *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2005, pp. 156–165.

[10] Andrew V Goldberg, Renato F Werneck, and Haim Kaplan. *Better landmarks within reach*. US Patent App. 11/593,857. 2006.

[11] Tim Kieritz et al. "Distributed time-dependent contraction hierarchies". In: *International Symposium on Experimental Algorithms*. Springer. 2010, pp. 83–93.

[12] Dominik Schultes. "Route Planning in Road Networks." In: *Ausgezeichnete Informatikdissertationen*. 2008, pp. 271–280.

[13] Sabine Storandt. "Contraction hierarchies on grid graphs". In: *Annual Conference on Artificial Intelligence*. Springer. 2013, pp. 236–247.

[14] Christian Vetter. "Parallel time-dependent contraction hierarchies". In: *Student Research Project* (2009).