

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 6, Issue 10, October 2017

Tracing Optimal Path on the Basis of Crowd on Road

Kalyani Shebe¹, Ashwini Shebe², Rushab Warahdkar³, Viren Ghate⁴, Associate Prof. Raana Syeda⁵

UG Student, Department of Computer Science and Engineering, Jhulelal Institute of Technology, Lonara,
Nagpur, India¹

UG Student, Department of Computer Science and Engineering, Jhulelal Institute of Technology, Lonara,
Nagpur, India²

UG Student, Department of Computer Science and Engineering, Jhulelal Institute of Technology, Lonara,
Nagpur, India³

UG Student, Department of Computer Science and Engineering, Jhulelal Institute of Technology, Lonara,
Nagpur, India⁴

Associate Professor, Department of Computer Science and Engineering, Jhulelal Institute of Technology, Lonara,
Nagpur, India⁵

ABSTRACT: Nowadays, navigation system have become integral part of everyone's life used to gives the shortest path to a client. This is very important in modern car navigation systems as it helps drivers to make sensible decisions. To our best knowledge, there is no efficient system/solution that can offer affordable costs at both client and server sides for shortest path over large crowd traffic. Unfortunately, the client-server architecture scales poorly with the number of clients. The approach is to let the server collect live traffic information and then broadcast them over radio or wireless network channel. We develop a new framework called live traffic index (LTI) which enables drivers to quickly and effectively collect the live traffic circumstances information on the broadcasting channel. An impressive result is that the driver can compute/update their shortest path result by receiving only a small fraction of the index and they are also download images on their sufficient devices.

KEYWORDS: Shortest path, Global Positioning System, Broadcasting.

I. INTRODUCTION

Shortest path computation is an important function in modern car navigation systems. This function helps a driver to figure out the best route from his current position to destination. Typically, the shortest path is computed by offline data pre-stored in the navigation systems and the weight (travel time) of the road edges is estimated by the road distance or historical data. The project is a data warehouse and mining based one. The primary objective behind the development of "Shortest Path Over Large Crowd Traffic" is to provide shortest path among large traffic.

With the ever growing popularity of online map applications and their wide deployment in mobile devices and car-navigation systems, the multiple number of users search for efficient fastest paths with the corresponding travel-time. In real-world the actual travel time on a road heavily depends on the traffic congestion and therefore, it is time-dependent network. Specifically, the quickest path from one place to another varies depending on the departure-time from the source. To our best knowledge, there is no efficient solution that can offer affordable costs at both client and server sides for shortest route over crowd. Unfortunately, the conventional client-server architecture can be used to answer shortest path problem on live traffic data. In this case, the navigation system typically sends the shortest path

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 6, Issue 10, October 2017

query to the service provider and waits the result back from the traffic provider (called result transmission model) therefore, this model is facing scalability limitations in terms of network bandwidth and server loading. Based on a telecommunication expert, the world's cellular networks need to provide 100 times the capacity in 2015 when compared to the networks in 2011. Nowadays, several online services provide live traffic data (by analyzing collected data from road sensors, traffic cameras,) and crowdsourcing techniques (such as Google-Map, Navteq, INRIX Traffic Information Provider and TomTom NV on certain mobile devices). These systems can calculate the snapshot shortest path queries based on current live traffic data. Developed a client-server system for monitoring of registered shortest path queries. For each registered query, the server first precomputes different shortest paths from source to target. Then, the server periodically updates the travel times on these paths based on the live traffic, and reports the current best path to the corresponding user. In addition, the current best paths are approximate results and the system does not provide any accuracy guarantee.

An alternative solution live traffic is updated frequently as these data can be collected by using broadcast live traffic data over wireless network (e.g., 3G, LTE, Mobile WiMAX etc.). The navigation system receives the live traffic data from the broadcast channel and executes the computation locally (called raw transmission model). The traffic data are broadcasted by a sequence of packets for each broadcast cycle. To answer shortest path queries based on live traffic circumstances, the navigation system must fetch those updated packets for each broadcast cycle. The main challenge on answering live shortest paths is scalability, in terms of the number of clients and the amount of live traffic updates. A new and promising solution to the shortest path computation is to broadcast an air index over the wireless network (called index transmission model). The main advantages of this model are that the network overhead is independent of the number of clients and every client only downloads a portion of the entire road map according to the index information. For instance, the proposed index constitutes a set of pair wise minimum and maximum traveling costs between every two sub partitions of the road map. However, these methods only solve the scalability issue for the number of clients but not for the amount of live traffic updates.

LTI is expected to provide relatively short tune-in cost (at client side), fast query response time (at client side), small broadcast size (at server side), and light maintenance time (at server side). The index structure of LTI is optimized by two novel techniques, graph partitioning and stochastic-based construction, after conducting a thorough analysis on the hierarchical index techniques. To the best of our knowledge, this is the first work to give a thorough cost analysis on the hierarchical index techniques and apply stochastic process to optimize the index hierarchical structure. LTI efficiently maintains the index for live traffic circumstances by incorporating Dynamic Shortest Path Tree (DSPT) into hierarchical index techniques. LTI reduces the tune-in cost up to an order of magnitude as compared to the state-of-the-art competitors; while it still provides competitive query response time, broadcast size, and maintenance time. An impressive result is that the driver can compute/update their shortest path

The index structure of LTI is optimized by two novel techniques-1] Graph partitioning and 2] Stochastic-based construction, after conducting a thorough analysis on the hierarchical index techniques. The graph partitioning divided the word and minimize communication and also it partition in to two parts it is called graph by section which recursively applied can be turn into algorithms for complete graph. To the best of our knowledge, this is the first work to give a thorough cost analysis on the hierarchical index techniques and apply stochastic process to optimize the index hierarchical structure. LTI efficiently maintains the index for live traffic circumstances by incorporating Dynamic Shortest Path Tree (DSPT) into hierarchical index techniques. LTI reduces the tune-in cost up to an order of magnitude as compared to the state-of-the-art competitors; while it still provides competitive query response time, broadcast size, and maintenance time. Paper is organized as follows. Section II describes shortest path over large crowd traffic using live traffic index algorithm are uses.. Which modules are used in it that is given in Section III. Section IV presents the step of the algorithm used in paper. Finally, Section V presents conclusion.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 6, Issue 10, October 2017

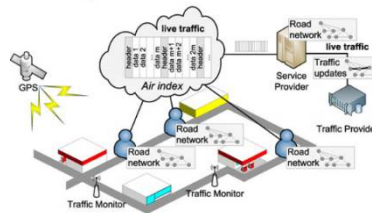


Fig: System Architecture of LTI

II. RELATED WORK

A. Spectral Clustering Based on the Graph Laplacian:

A connection between the Cheeger cut and the second eigenvector of the graph p -Laplacian, a nonlinear generalization of the graph Laplacian. A p -Laplacian which is slightly from the one used for semi supervised learning. The main motivation for the use of eigenvectors of the graph p -Laplacian was the generalize isoperimetric inequality. In which relates the second eigenvalue of the graph p -Laplacian to the optimal Cheeger cut. The isoperimetric inequality becomes tight as p , so that the second Eigen value converges to the optimal Cheeger cut value.

B. SHARC: Fast and Robust Unidirectional Routing :

Introduce SHARC-Routing, a fast and robust approach for unidirectional routing in large networks. The central idea of SHARC (Shortcuts + Arc-Flags) is the adaptation of techniques developed for Highway Hierarchies to Arc Flags. In general, SHARC-Routing iteratively constructs a contraction-based hierarchy during pre-processing and automatically sets arc-flags for edges removed during contraction. More precisely, arc-flags are set in such a way that a unidirectional query considers these removed component-edges only at the beginning and the end of a query as a result, able to route very efficiently in scenarios where other techniques fail due to their bidirectional nature. It turned out that SHARC was a promising candidate for routing in time-dependent networks.

C. Computing Point To Point Shortest Path from External Memory:

The ALT algorithm for the point-to-point shortest path problem in the context of road networks the suggest improvements to the algorithm itself and to its pre-processing stage. Also develop a memory-efficient implementation of the algorithm that runs on a Pocket PC (Personal Computer). It stores graph data in a flash memory card and uses RAM (Random Access Memory) to store information only for the part of the graph visited by the current shortest path computation. The implementation works even on very large graphs, including that of the North America road network, with almost 30 million vertices.

D. Time-Dependent SHARC-Routing :

During the last years, many speed-up techniques for Dijkstra's algorithm have been developed. As a result, computing a shortest path in a static road network is a matter of microseconds. However, only few of those techniques work in time-dependent networks. Unfortunately, such networks appear frequently in reality.

E. Shortest Path Tree Computation in Dynamic Graphs:

The Dynamic Shortest Path (DSP) problem is to compute S from D . This problem either focuses on a single edge weight change, or for multiple edge weight changes, some of them are incorrect or are not optimized. The correct and extend a few state-of-the-art dynamic SPT algorithms to handle multiple edge weight updates. Hence prove that these algorithms are correct dynamic algorithms may not out perform static algorithms all the time to evaluate the proposed dynamic algorithms, compare them with the well-known static Dijkstra's algorithm.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 6, Issue 10, October 2017

III. PROPOSED METHODOLOGY

We analyse the hierarchical index structures and study how to optimize the index. we present a stochastic based index construction that minimizes not only the size overhead but also reduces the search space of shortest path queries. To the best of our knowledge, this is the first work to analyse the hierarchical index structures and exploit the stochastic process to optimize the index. 4.1 Analysis of Hierarchical Index Structures Hierarchical index structures (e.g., HiTi) enable fast shortest path computation on a portion of entire index which significantly reduces the tune-in cost on the index transmission model.

It may comprise of following task:

- a. Live traffic index is expected to provide relatively short tune-in cost at client side.
- b. Fast query response time at client side.
- c. Small broadcast size at server side.
- d. Light maintenance time at server side.

Tune-in Cost (Client Side)

We prioritize the tune-in cost as the main optimized factor since it affects the duration of client receivers into active mode and power consumption is essentially determined by the tuning cost (i.e., number of packets received). These services may include providing live weather information, delivering latest promotions in surrounding area, and monitoring avail-ability of parking slots at destination.

Broadcast Size and Maintenance Time (Server Side)

The index maintenance time and broadcast size relate to the freshness of the live traffic information. The maintenance time is the time required to update the index according to live traffic information. The broadcast size is relevant to the latency of receiving the latest index information.

Query Response Time (Client Side)

The last factor is the response time at client side. Given a proper index structure, the response time of shortest path computation can be very fast (i.e., few milliseconds on large road maps) which is negligible compared to access latency for current wireless network speed. The computational so consumes power but their effect is Outweighed by communication.

IV. EXPERIMENTAL RESULTS

In this section, we empirically evaluate the performance of some representative algorithms using the broadcasting architecture ,we ignore the client-server architecture due to massive live traffic in near future in section 1. From our discussion in Section 2, bi-directional search, ALT on dynamic graph (DALT), and dynamic shortest paths tree, are applicable to raw transmission model. On the other hand, contraction hierarchies, Hierarchical Multi-graph model, and our proposed live traffic index are applicable to index transmission model. We omit some methods (such as , SHARC,) due to their prohibitive maintenance time and broadcast size. In the following, we first describe the road map data used in experiments and describe the simulation of clients' movements and live traffic circumstances on a road map. Then, we study the performance of the above methodwith respect to various factors.

MAP DATA: We test with four different road maps, including New York City (NYC) (264k nodes, 733k edges), San Francisco bay area road map (SF) (174k nodes, 443k edges), San Joaquin road map (SJ) (18k nodes, 48k edges), and Oldenburg road map (OB) (6k nodes, 14k edges). All of them are available.

Simulation of Clients And Traffic Updates: We run the networkbased generator to generate the weight of edges. It initializes 100,000 cars (i.e., clients) and generates 1,000 new cars in each iteration default values. The weight of an edge is set to the average driving time on it. The initial weights of edges are assigned by the above network-based generator. In each iteration, we randomly select a set of edges to the update ratio and specific weight update

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 6, Issue 10, October 2017

settings. In our work, each weight update can be either a *light* traffic change, a *heavy* traffic change, or a *road maintenance*. The proportion of these update types are β , $1-\beta/2$, and $1-\beta/2$, respectively, where b is a ratio parameter. For each light traffic change, the edge weight is set to 20% of the current weight. For each heavy traffic change, the weight is set to a large value by multiplying a weight factor v (which is set to 5 by default). For each road maintenance, the weight is set to 1.

Implementation and Evaluation Platforms: All tested methods except CH [30] were implemented in Java. Experiments on the service provider were conducted on an Intel Xeon E5620 2.40 GHz CPU machine with 18 GBytes memory, running Ubuntu 10.1 and experiments on the client were performed on an Intel Core2Duo 2.66 GHz CPU machine with 4 GBytes memory, running Windows 7. For each method, we measure its performance in terms of tune-in size, query response time, broadcast size, and index maintenance time for all tested methods, and report its average performance over 2,000 shortest path queries. The response time is the query computation time at client and the maintenance time is the index maintenance time at service provider. In order to measure the exact transmission behavior, we use the number of packets received (broadcasted) by client (service provider) to represent the tune-in (broadcast) performance. Each packet size is of 128 bytes and the packet format. Each edge weight occupies 4 bytes. We randomly generate 1,000 queries at each Monte Carlo estimation and we only partition the graph into 2 to 16 subgraphs at every partitioning for boosting up the construction time. As a remark, each subgraph/partition (in the HiTi and LTI methods) may span over multiple packets.

V. CONCLUSION

We have implemented shortest path over large crowd traffic. In this paper we studied online shortest path computation; the shortest path result is computed/updated based on the live traffic circumstances. Our algorithm successfully detects the shortest possible path, which consists for large road traffic. We carefully analyze the existing work and discuss their inapplicability to the problem (due to their prohibitive maintenance time and large transmission overhead). To address the problem, we suggest a promising architecture that broadcasts the index on the air. LTI is a Pareto optimal solution in terms of four performance factors for online shortest path computation. In the future, we will extend our solution on time dependent.

REFERENCES

1. L. W.u, X. Xiao, D. Deng, G. Cong, A.D. Zhu, and S. Zhou, "Shortest Path and Distance Queries on Road Networks: An Experimental Evaluation," Proc. VLDB Endowment, vol. 5, no. 5, pp. 406-417, 2011.
2. Leong Hou U, Hong Jun Zhao, Man Lung Yiu, Yuhong Li, and Zhiguo Gong, "Towards Online Shortest Path Computation", IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. April 2014.
3. F. Zhan and C. Noon, "Shortest Path Algorithms: An Evaluation Using Real Road Networks," Transportation Science, vol. 32, no. 1, pp. 65-73, 1998.
4. M. Rice and V. J. Tsotras. Graph indexing of road networks for shortest path queries with label restrictions. PVLDB, 4:69-80.
5. Tejas Kadu, Suresh Kumar, Omkar Aher, Prasanna Ingrole, Prof. Ms. Shivganga Mujgond, "An Efficient Online Shortest Path Computation System For Real Road Networks", International Engineering Research Journal (IERJ) Volume 2 Issue 3 Page 1120-1123, 2016, ISSN 2395-62.
6. Ch. Venkateswarlu, B. V. Sri Ram, V. Sindhu "Online Shortest Path Computation Based on Live Traffic Index", Vol. 4, Issue 6, June 2016.
7. Madha Ramakrishna, Deepthi Janaagama, N. Anjaneyulu "Towards Online Shortest Path Computation" ISSN 2348-2370 Vol.09, Issue.02, February-2017, Pages:0245-0250.
8. Alison Brown, Jacob Griesbach and Bruce Bockius, "GPS tracking location based service using wrist watch GeoZigBee Sensors", Proceedings of ION NTM, 2007, Som Diego, pp 1-10.
9. S. Jung and S. Pramanik, "An Efficient Path Computation Model for Hierarchically Structured Topographical Road Maps," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 5, pp. 1029-1046.
10. F. Wei, "TED: Efficient Shortest Path Query Answering on Graphs," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 99-110, 2010.
11. R.J. Gutman, "Reach-Based Routing: A New Approach to Shortest Path Algorithms Optimized for Road Networks," Proc. Sixth Workshop Algorithm Eng. and Experiments and the First Workshop Analytic Algorithms and Combinatorics (ALENEX/ANALC), pp. 100-111, 2004.
12. Alison Brown, Jacob Griesbach and Bruce Bockius, "GPS tracking location based service using wrist watch GeoZigBee Sensors", Proceedings of ION NTM, 2007, Som Diego, pp 1-10.
13. B. Jiang, "I/O-Efficiency of Shortest Path Algorithms: An Analysis," Proc. Eight Int'l Conf. Data Eng. (ICDE), pp. 12-19, 1992.