ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Finding the Shortest Path Computation in online

P.Charitha
M.Tech Student,
Department of CSE,
Sree Rama institute of Technology and Science,
Kuppenakuntla, Penuballi, Khammam,TS India.

S.Suresh
Assistant Professor,
Department of CSE,
Sree Rama institute of Technology and Science,
Kuppenakuntla, Penuballi, Khammam,TS India.

ABSTRACT:

The online shortest path problem aims at computing the shortest path based on live traffic circumstances. 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 online shortest path computation. Unfortunately, the conventional client-server architecture scales poorly with the number of clients. A promising approach is to let the server collect live traffic information and then broadcast them over radio or wireless network. This approach has excellent scalability with the number of clients. Thus, we develop a new framework called live traffic index (LTI) which enables drivers to quickly and effectively collect the live traffic 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. Our experimental study shows that LTI is robust to various parameters and it offers 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) for online shortest path problem.

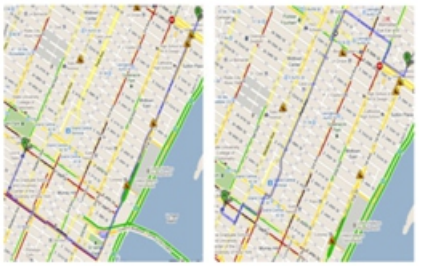
Index Terms:
Shortest path, air index, broadcasting.

INTRODUCTION:

SHORTEST path computation is an important function in modern car navigation systems and has been extensively studied in [1], [2], [3], [4], [5], [6], [7], [8]. 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. Unfortunately, road traffic circumstances change over time. Without live traffic circumstances, the route returned by the navigation system is no longer guaranteed an accurate result. We demonstrate this by an example in Fig. 1. Suppose that we are driving from Lord & Taylor (label A) to Mt Vernon Hotel Museum (label B) in Manhattan,NY. Those old navigation systems would suggest a route based on the pre-stored distance information as shown in Fig. 1a.

Note that this route passes through four road maintenance operations (indicated by maintenance icons) and one traffic congested road (indicated by a red line). In fact, if we take traffic circumstances into account, then we prefer the route in Fig. 1b rather than the route in Fig. 1a. Nowadays, several online services provide live traffic data (by analyzing collected data from road sensors, traffic cameras, and crowdsourcing techniques), such as Google- Map [9], Navteq [10], INRIX Traffic Information Provider [11], and TomTom NV [12], etc. These systems can calculate the snapshot shortest path .



(a) Shortest route using pre-stored weights (b) Shortest route using live traffic (by LTI)

Fig. 1. Two alternative shortest paths in *Manhattan, NY.*

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

queries based on current live traffic data; however, they do not report routes to drivers continuously due to high operating costs. Answering the shortest paths on the live traffic data can be viewed as a continuous monitoring problem in spatial databases, which is termed online shortest paths computation (OSP) in this work.To the best of our knowledge, this problem has not received much attention and the costs of answering such continuous queries vary hugely in different system architectures.

Existing System:

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 , etc. These systems can calculate the snapshot shortest path queries based on current live raffic data; however, they do not report routes to drivers continuously due to high operating costs. Answering the shortest paths on the live traffic data can be viewed as a continuous monitoring problem in spatial databases, which is termed online shortest paths computation (OSP) in this work. To the best of our knowledge, this problem has not received much attention and the costs of answering such continuous queries vary hugely in different system architectures. Typical client-server architecture can be used to answer shortest path queries on live traffic data.

DISADVANTAGES OF EXISTING SYSTEM:

1. Scalability limitations in terms of network bandwidth and server loading.
2. Online Shortest Paths computation is not much attention.

Proposed System:

Motivated by the lack of off-the-shelf solution for OSP, in this paper we present a new solution based on the index transmission model by introducing live traffic index (LTI) as the core technique. 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) for OSP.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.

ADVANTAGES OF PROPOSED SYSTEM:

1. The server periodically updates the travel times on these paths based on the latest traffic, and reports the current best path to the corresponding user.
2. Efficiently maintains the index for live traffic circumstances.
3. 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.
4. LTI efficiently maintains the index for live traffic circumstances by incorporating Dynamic Shortest Path Tree (DSPT) into hierarchial index techniques. In addition, a bounded version of DSPT is proposed to further reduce the broadcast overhead.
5. 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.

PRELIMINARY:

Performance Factors:

The main performance factors involved in OSP are: (i) tunein cost (at client side), (ii) broadcast size (at server side), and (iii) maintenance time (at server side), and (iv) query response time (at client side). In this work, 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) [17], [23]. In addition, shortening the duration of active mode enables the clients to receive more services simultaneously by selective tuning [24]. These services mayinclude providing live weather information, delivering latest promotions in surrounding area, and monitoring availability of parking slots at destination. If we minimize the tune-in cost of one service, then we reserve more resources for other services. 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. As the freshness is one of our main design criteria, we must provide reasonable costs for these two factors. 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.,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

few milliseconds on large road maps) which is negligible compared to access latency for current wireless network speed. The computation also consumes power but their effect is outweighed by communication. It remains, however, anevaluated factor for OSP.

LTI OVERVIEW AND OBJECTIVES:

LTI Overview:

A road network monitoring system typically consists of a service provider, a large number of mobile clients (e.g., vehicles), and a traffic provider (e.g., GoogleMap, NAVTEQ, INRIX, etc.). Fig. 3 shows an architectural overview of this system in the context of our live traffic index framework. The traffic provider collects the live traffic circumstances from the traffic monitors via techniques like road sensors and traffic video analysis. The service provider periodically receives live traffic updates from the traffic provider and broadcasts the live traffic index on radio or wireless network (e.g., 3G, LTE, Mobile WiMAX, etc.). When a mobile client wishes to compute and monitor a shortest path, it listens to the live traffic index and reads the relevant portion of the index for computing the shortest path.

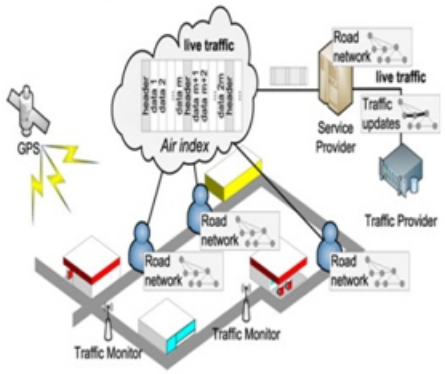


Fig. 3. LTI system overview.

LTI Objectives:

To optimize the performance of the LTI components, our solution should support the following features. (1) Efficient maintenance strategy. Without efficient maintenance strategy, long maintenance time is needed at server side so that the traffic information is no longer live. This can reduce the maintenance time spent at component a.(2) Light index overhead. The index size must be controlled in a reasonable ratio to the entire road map data. This reduces not only the length of a broadcast cycle, but also makes clients listen fewer packets in the broadcast channel. This can save the communication cost at components b and c.

(3) Efficient computation on a portion of entire index. This property enables clients to compute shortest path on a portion of the entire index. The computation at component d gets improved since it is executed on a smaller graph. This property also reduces the amount of data received andenergy consumed at component c. Inspired by these properties, LTI has relatively short tune-in cost (at client side), fast query response time (at client side), small broadcast size (at server side), and light index maintenance time (at server side) for OSP. As discussed in Section 2.2, the hierarchical index structures enable clients to compute the shortest path on a portion of entire index. However, without pairing up with the first and second features, the communication and computation costs are still infeasible for OSP. To achieve these two features, in Sections 4 and 6, we will discuss how to optimize the hierarchical structure and efficiently maintain the index according to live traffic circumstances.

LTI TRANSMISSION:

Broadcasting Scheme:

The broadcasting model uses radio or wireless network (e.g., 3G, LTE, Mobile WiMAX) as the transmission medium. When the server broadcasts a data set (i.e., a "programme"), all clients can listen to the data set concurrently. Thus, this transmission model scales well independent of the number of clients. A broadcasting scheme is a protocol to be followed by the server and the clients. The (1,m) interleaving scheme [23] is one of the best broadcasting schemes. Table 1 shows an example broadcasting cycle with m ¼ 3 packets and the entire data set contains six data items. First, the server partitions the data set into m equi-sized data segments. Each packet contains a header and a data segment, where a header describes the broadcasting schedule of all packets. In this example, the variables i and n in each header represent the last broadcasted itemand the total number of items. The server periodically broadcasts a sequence of packets (called as a broadcast cycle).We use a concrete example to demonstrate how a client receives her data from the broadcast channel. Suppose that a client wishes to query for the data object o5. First, the client tunes in the broadcast channel and waits until the next header is broadcasted. For instance, the client is listening to the header of the first packet, and finds out that the third packet contains o5. In order to preserve energy, the client sleeps until the broadcasting time of that packet. Then, it wake-ups and reads the requested data item from the packet.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Algorithm 1 Stochastic Partitioning Algorithm

PQ: a priority queue; I: index structure;
Algorithm $partition(G$:the graph, γ:the number of partitions)
1: $(\lambda, \mathcal{V}) := eigen(G)$ and $n :=$ root of I
2: insert $(n, G, \mathcal{V}, \lambda)$ into PQ in decreasing order to λ
3: **while** $|PQ| < \gamma$ **do**
4: $(n, G, \mathcal{V}, \lambda) := PQ.pop()$
5: **for** k:=2 to $\gamma - |PQ| + 1$ **do**
6: decompose G into $SG_1 ... SG_k$ s.t. Eq. 4 is minimized
7: form a temporal index I' that attaches $SG_1 ... SG_k$
8: **if** avg($\mathcal{S}(I')$) is better than $best_S$ **then**
9: update $best_S$ and $best_{SG} := \{SG_1, ..., SG_k\}$
10: attach $best_{SG}$ as n's children
11: **for** i:=1 to $|best_{SG}|$ **do**
12: insert $(n_i, SG_i, \mathcal{V}_i, \lambda_i)$ into PQ
13: **return** I

LTI MAINTENANCE:

In order to keep the freshness of the broadcasted index, the cost of index maintenance is necessarily minimized. In this section, we study an incremental update approach that can efficiently maintain the live traffic index according to the updates. As a remark, the entire update process is done at the service provider and there is no extra data structure being broadcasted to the clients. There is a bottom-up framework [21] available to maintain the hierarchical index structure according to the updates. Their idea is to re-compute the affected subgraphs starting from lowest level (i.e., leaf subgraphs).

PUTTING ALL TOGETHER:

We are now ready to present our complete LTI framework, which integrates all techniques been discussed. A client can invoke Algorithm 2 in order to find the shortest path from a source s to a destination t. First, the client generates a search graph Gq based on s (i.e., current location) and d. When the client tunes-in the broadcast channel it keeps listening until it discovers a header segment After reading the header segment, it decides the necessary segments (to be read) for computing the shortest path. We then discuss about the tasks to be performed by the service provider, as shown in Algorithm 3. The first step is devoted to construct the live traffic index; they are offline tasks to be executed once only.

Algorithm 2 Client Algorithm

Algorithm $Client(I$:LTI, s:source, t:destination)
1: generate G^q from I based on s and d
2: listen to the channel for a header segment
3: read the header segment ▷ Section 5.3
4: decide the necessary segments to be read ▷ Section 5.3
5: wait for those segments, read them to update the weight of G^q
6: compute the shortest path (from s to t) on G^q ▷ Section 4.1

Algorithm 3 Service Algorithm

Algorithm $Service(G$:graph)
1: construct I and $\{SG_i\}$ based on G ▷ Section 4.2
2: **for** each broadcast cycle **do**
3: collect traffic updates from the traffic provider
4: update the subgraphs $\{SG_i\}$ ▷ Section 6
5: broadcast the subgraphs $\{SG_i\}$ ▷ Section 5.2

CONCLUSIONS:

In this paper we studied online shortest path computation; the shortest path result is computed/updated based on the live traffic circumstances. 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. We first identify an important feature of the hierarchical index structure which enables us to compute shortest path on a small portion of index. This important feature is thoroughly used in our solution, LTI. Our experiments confirm that 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 networks. This is a very interesting topic since the decision of a shortest path depends not only on current traffic data but also based on the predicted traffic circumstances.

REFERENCES:

[1] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes, "InTransit to Constant Time Shortest-Path Queries in RoadNetworks," Proc. Workshop Algorithm Eng. and Experiments(ALENEX), 2007.
[2] P. Sanders and D. Schultes, "Engineering Highway Hierarchies,"Proc. 14th Conf. Ann. European Symp. (ESA), pp. 804-816, 2006.
[3] G. Dantzig, Linear Programming and Extensions, series Rand CorporationResearch Study Princeton Univ. Press, 1963.
[4] 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 Algorithmics and Combinatorics (ALENEX/ANALC), pp. 100-111, 2004.
[5] B. Jiang, "I/O-Efficiency of Shortest Path Algorithms: An Analysis," Proc. Eight Int'l Conf. Data Eng. (ICDE), pp. 12-19, 1992.
[6] P. Sanders and D. Schultes, "Highway Hierarchies Hasten ExactShortest Path Queries," Proc. 13th Ann. European Conf. Algorithms (ESA), pp. 568-579, 2005.
[7] D. Schultes and P. Sanders, "Dynamic Highway-Node Routing,"Proc. Sixth Int'l Conf. Experimental Algorithms (WEA), pp. 66-79,2007.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[8] F. Zhan and C. Noon, "Shortest Path Algorithms: An EvaluationUsing Real Road Networks," Transportation Science, vol. 32, no. 1,pp. 65-73, 1998.
[9] "Google Maps," http://maps.google.com, 2014.
[10] "NAVTEQ Maps and Traffic," http://www.navteq.com, 2014.
[11] "INRIX Inc. Traffic Information Provider," http://www.inrix.com, 2014.
[12] "TomTom NV," http://www.tomtom.com, 2014.
[13] "Cisco Visual Networking Index: Global Mobile Data Traffic ForecastUpdate, 2010-2015," 2011.
[14] D. Stewart, "Economics of Wireless Means Data Prices Bound toRise," The Global and Mail, 2011.
[15] W.-S. Ku, R. Zimmermann, and H. Wang, "Location-Based SpatialQuery Processing in Wireless Broadcast Environments," IEEETrans. Mobile Computing, vol. 7, no. 6, pp. 778-791, June 2008.
[16] N. Malviya, S. Madden, and A. Bhattacharya, "A ContinuousQuery System for Dynamic Route Planning," Proc. IEEE 27th Int'lConf Data Eng. (ICDE), pp. 792-803, 2011.
[17] G. Kellaris and K. Mouratidis, "Shortest Path Computation on AirIndexes," Proc. VLDB Endowment, vol. 3, no. 1, pp. 741-757, 2010.
[18] Y. Jing, C. Chen, W. Sun, B. Zheng, L. Liu, and C. Tu, "Energy-Efficient Shortest Path Query Processing on Air," Proc. 19th ACMSIGSPATIAL Int'l Conf. Advances in Geographic Information Systems(GIS), pp. 393-396, 2011.
[19] R. Goldman, N. Shivakumar, S. Venkatasubramanian, and H.Garcia-Molina, "Proximity Search in Databases," Proc. Int'l Conf. Very Large Databases (VLDB), pp. 26-37, 1998.
[20] N. Jing, Y.-W. Huang, and E.A. Rundensteiner, "HierarchicalEncoded Path Views for Path Query Processing: An OptimalModel and Its Performance Evaluation," IEEE Trans. Knowledgeand Data Eng., vol. 10, no. 3, pp. 409-432, May 1998.
[21] S. Jung and S. Pramanik, "An Efficient Path Computation Modelfor Hierarchically Structured Topographical Road Maps," IEEETrans. Knowledge and Data Eng., vol. 14, no. 5, pp. 1029-1046, Sept.2002.
[22] E.P.F. Chan and Y. Yang, "Shortest Path Tree Computation inDynamic Graphs," IEEE Trans. Computers, vol. 58, no. 4, pp. 541-557, Apr. 2009.
[23] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Data on Air:Organization and Access," IEEE Trans. Knowledge and Data Eng.,vol. 9, no. 3, pp. 353-372, May/June 1997.
[24] J.X. Yu and K.-L. Tan, "An Analysis of Selective Tuning Schemesfor Nonuniform Broadcast," Data and Knowledge Eng., vol. 22,no. 3, pp. 319-344, 1997.
[25] A.V. Goldberg and R.F.F. Werneck, "Computing Point-to-PointShortest Paths from External Memory," Proc. SIAM Workshop AlgorithmsEng. and Experimentation and the Workshop Analytic Algorithmicsand Combinatorics (ALENEX/ANALCO), pp. 26-40, 2005.
[26] M. Hilger, E. K€ohler, R. M€ohring, and H. Schilling, "Fast Point-to-Point Shortest Path Computations with Arc-Flags," The ShortestPath Problem: Ninth DIMACS Implementation Challenge, vol. 74,pp. 41-72, American Math. Soc., 2009.
[27] A.V. Goldberg and C. Harrelson, "Computing the Shortest Path:Search Meets Graph Theory," Proc. 16th Ann. ACM-SIAM Symp.Discrete Algorithms (SODA), pp. 156-165, 2005.
[28] D. Delling and D. Wagner, "Landmark-Based Routing in DynamicGraphs," Proc. Sixth Int'l Workshop Experimental Algorithms (WEA),pp. 52-65, 2007.

Author's:

P.Charitha is a student of Sree Rama Institute of Technology & Science, Kuppenakuntla, Penuballi, Khammam, TS,India.Presently she is Pursuing her M.Tech (CSE) from this college Her area of interests includes Information Security, Cloud Computing, Data Communication & Networks.

S. Suresh well known author and excellent teacher. He is working as a H.O.D for the Department of M.Tech., Computer Science & Engineering, Sree Rama Institute of Technology & Science, Kuppenakuntla, Penuballi, Khammam. He has vast teaching experience in various engineering colleges. To his credit couple of publications both National& International conferences / journals. His area ofInterest includes Data Warehouse and Data Mining,information security, Data Communications &Networks, Software Engineering and otheradvances in Computer Applications. He has guided many projects for Engineering Students