ISSN: 2321-7782 (Online)
Volume 4, Issue 4, April 2016
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Find Online Shortest Path on the Basis of Live Traffic Circumstances

Nirav J Padhiyar[1]
India

Azaruddin C Shaikh[2]
India

Sandip Janjire[3]
India

Prof. Ashutosh Chowdary[3]
India

Abstract: On the live traffic circumstances online shortest path calculate shortest distance. In modern car navigation system it is very useful for sensible decision to the driver. The online shortest path computation can offer affordable costs at both client and server sides there are no efficient system solution. Unfortunately, the conventional client-server architecture balance badly with the number of clients. The server collects live traffic information and then broadcast them over radio or wireless network by using promising approach. This approach has outstanding scalability with the number of clients. Thus, system build up a new framework called live traffic index (LTI) which enables drivers too rapidly and efficiently gather the live traffic information on the spreading channel. An inspiring result is that the driver can calculate/update their shortest path result by getting only a small portion of the index. Our experimental study shows that LTI is robust to different parameters and it offers comparatively short tune-in cost (at client side), fast query answer time (at client side), small transmit size (at server side), and light maintenance time (at server side) for online shortest path problem.

Keywords: Shortest path, air index, broadcasting.

I. INTRODUCTION

In modern car navigation systems and has been expansively studied the shortest path computation is an important function. This function helps a driver to figure out the most excellent route from his present location to target location. 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. These systems can calculate the snapshot shortest path 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. Typical client-server architecture can be used to answer shortest path queries on live traffic data. In this case, the navigation system typically sends the shortest path query to the service provider and waits the result back from the provider. However, given the rapid growth of mobile devices and services, this model is facing scalability limitations in terms of network bandwidth and server loading.

II. LITERATURE REVIEW

1) An Efficient Path Computation Model for Hierarchically Structured Topographical Road Maps

AUTHORS: S. Jung and S. Pramanik

In this paper, we have developed a HiTi (Hierarchical MulTi) graph model for structuring large topographical road maps to speed up the minimum cost route computation. The HiTi graph model provides a novel approach to abstracting and structuring a topographical road map in a hierarchical fashion. We propose a new shortest path algorithm named SPAH, which utilizes HiTi graph model of a topographical road map for its computation. We give the proof for the optimality of SPAH. Our performance analysis of SPAH on grid graphs showed that it significantly reduces the search space over existing methods. We also present an in-depth experimental analysis of HiTi graph method by comparing it with other similar works on grid graphs. Within the HiTi graph framework, we also propose a parallel shortest path algorithm named ISPAH. Experimental results show that inter query shortest path problem provides more opportunity for scalable parallelism than the intra query shortest path problem.

2) Shortest Path Tree Computation in Dynamic Graphs

AUTHORS: E.P.F. Chan and Y. Yang

Let G = (V, E, omega) be a simple digraph, in which all edge weights are nonnegative real numbers. Let G' be obtained from G by an application of a set of edge weight updates to G. Let sisinV and let T_s and T_s' be Shortest Path Trees (SPTs) rooted at s in G and G', respectively. The Dynamic Shortest Path (DSP) problem is to compute T_s' from T_s. Existing work on this problem focuses on either a single edge weight change or multiple edge weight changes in which some of them are incorrect or are not optimized. We correct and extend a few state-of-the-art dynamic SPT algorithms to handle multiple edge weight updates. We prove that these algorithms are correct. Dynamic algorithms may not outperform static algorithms all the time. To evaluate the proposed dynamic algorithms, we compare them with the well-known static Dijkstra algorithm. Extensive experiments are conducted with both real-life and artificial data sets. The experimental results suggest the most appropriate algorithms to be used under different circumstances.

3) Data on Air: Organization and Access

AUTHORS: T. Imielinski, S. Viswanathan, and B.R. Badrinath,

Organizing massive amount of data on wireless communication networks in order to provide fast and low power access to users equipped with palmtops, is a new challenge to the data management and telecommunication communities. Solutions must take under consideration the physical restrictions of low network bandwidth and limited battery life of palmtops. This paper proposes algorithms for multiplexing clustering and non clustering indexes along with data on wireless networks. The power consumption and the latency for obtaining the required data are considered as the two basic performance criteria for all algorithms. First, this paper describes two algorithms namely, (1, m) indexing and Distributed Indexing, for multiplexing data and its clustering index. Second, an algorithm called Non clustered Indexing is described for allocating static data and its corresponding non clustered index. Then, the Non clustered indexing algorithm is generalized to the case of multiple indexes. Finally, the proposed algorithms are analytically demonstrated to lead to significant improvement of battery life while retaining a low latency.

4) Query Processing Using Distance Oracles for Spatial Networks

AUTHORS: J. Sankaranarayanan and H. Samet,

The popularity of location-based services and the need to do real-time processing on them has led to an interest in performing queries on transportation networks, such as finding shortest paths and finding nearest neighbors. The challenge here is that the efficient execution of spatial operations usually involves the computation of distance along a spatial network instead of "as the crow flies," which is not simple. Techniques are described that enable the determination of the network distance between any pair of points (i.e., vertices) with as little as O(n) space rather than having to store the n^2 distances between all pairs. This is done by being willing to expend a bit more time to achieve this goal such as O(log n) instead of O(1), as well as by accepting an error varepsilon in the accuracy of the distance that is provided. The strategy that is adopted reduces the space requirements and is based on the ability to identify groups of source and destination vertices for which the distance is approximately the same within some varepsilon. The reductions are achieved by introducing a construct termed a distance oracle that yields an estimate of the network distance (termed the varepsilon-approximate distance) between any two vertices in the spatial network. The distance oracle is obtained by showing how to adapt the well-separated pair technique from computational geometry to spatial networks. Initially, an varepsilon-approximate distance oracle of size O ({nover varepsilon^2}) is used that is capable of retrieving the approximate network distance in O(log n) time using a B-tree. The retrieval time can be theoretically reduced further to O(1) time by proposing another varepsilon-approximate distance oracle of size O({n log nover varepsilon^2}) that uses a hash table. Experimental results indicate that the proposed technique is scalable and can be applied to sufficiently large road networks. For example, a 10-percent-approximate oracle (vareps- - ilon = 0.1) on a large network yielded an average error of 0.9 percent with 90 percent of the answers having an error of 2 percent or less and an average retrieval time of 68 mu {rm seconds}. The fact that the network distance can be approximated by one value is used to show how a number of spatial queries can be formulated using appropriate SQL constructs and a few built-in primitives. The result is that these operations can be executed on almost any modern database with no modifications, while taking advantage of the existing query optimizers and query processing strategies.

5) Spectral Clustering Based on The Graph – Laplacian

AUTHORS: T. B€uhler and M. Hein,

We present a generalized version of spectral clustering using the graph p-Laplacian, a nonlinear generalization of the standard graph Laplacian. We show that the second eigenvector of the graph p-Laplacian interpolates between a relaxation of the normalized and the Cheeger cut. Moreover, we prove that in the limit as p ! 1 the cut found by thresholding the second eigenvector of the graph p-Laplacian converges to the optimal Cheeger cut. Furthermore, we provide an e_cient numerical scheme to compute the second eigenvector of the graph p-Laplacian. The experiments show that the clustering found by p-spectral clustering is at least as good as normal spectral clustering, but often leads to signi_cantly better results.

III. Survey of 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.

Advantage 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.

IV. Mathematical Model

Let S is the Whole System Consist of

S= {I, P, O}

I = Input.

I = {U, Q, A, S,}

U = User

U = {u1,u2….un}

Q = Query Entered by user

Q = {q1, q2, q3…qn}

A = Algorithms

S = Source

P = Process:

Adaptation of Existing Approaches

Raw Transmission Model:

- Under the raw transmission model, the traffic data (i.e.,edge weights) are broadcasted by a set of packets for each broadcast cycle.
- Each header stores the latest time stamp of the packets, so that clients can decide which packets have been updated, and only fetch those updated packets in the current broadcast cycle.

Following methods either directly calculate the shortest path or efficiently maintain certain data structure for the shortest path computation.

- Uninformed search:

Traverses graph nodes in ascending order of their distances from the source s, and eventually discovers the shortest path to the destination t.

- Goal directed approaches:

Search towards the target by filtering out the edges that cannot possibly belong to the shortest path. The filtering procedure requires some pre-computed information.

- Dynamic shortest path tree (DSPT) :

It Maintain a tree structure locally for efficient shortest path retrieval.

Index Transmission Model

The index transmission model enables servers to broadcast an index instead of raw traffic data. We review the state-of the-art indices for shortest path computation and discuss their applicability on the index transmission model.

Road map hierarchical approaches: try to exploit the hierarchical structure to the road map network in a pre-processing step, which can be used to accelerate all subsequent queries.

LTI Objectives

To optimize the performance of the LTI components, our solution should support the following features.

(1) Efficient maintenance strategy.

(2) Light index overhead.

(3) Efficient computation on a portion of entire index.

LTI TRANSMISSION

We present how to transmit LTI on the air index.

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.

LTI on Air

To broadcast a hierarchical index using the (1,m) interleaving scheme, we first partition the index into two components: the index structure and the weight of edges.

LTI MAINTENANCE

In order to keep the freshness of the broadcasted index, the cost of index maintenance is necessarily minimized.

OUTPUT: Finally the user is able to search shortest path.

V. System Architecture

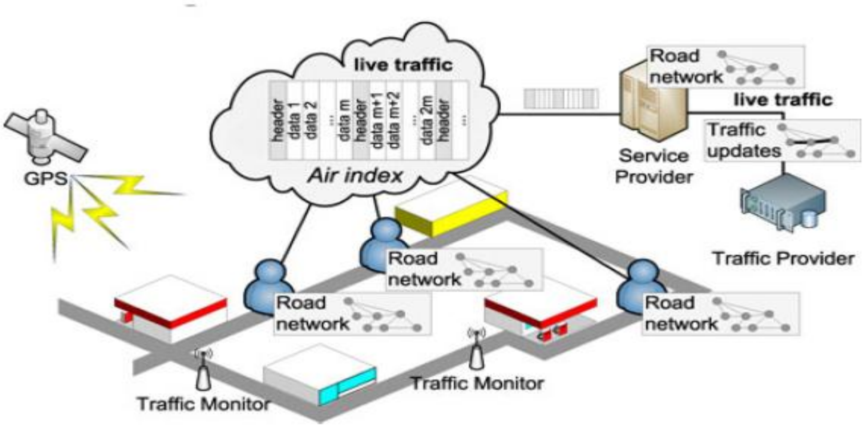

Fig1. System architecture

VI. Conclusion

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, "In Transit to Constant Time Shortest-Path Queries in Road Networks," 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 Corporation Research 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 Exact Shortest 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.

8. F. Zhan and C. Noon, "Shortest Path Algorithms: An Evaluation Using Real Road Networks," Transportation Science, vol. 32, no. 1, pp. 65-73, 1998.

9. "Google Maps," http://maps.google.com, 2014.

10. "NAVTEQMapsandTraffic,"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 Forecast Update, 2010-2015," 2011.

14. D. Stewart, "Economics of Wireless Means Data Prices Bound to Rise," The Global and Mail, 2011.

15. W.-S. Ku, R. Zimmermann, and H. Wang, "Location-Based Spatial Query Processing in Wireless Broadcast Environments," IEEE Trans. Mobile Computing, vol. 7, no. 6, pp. 778-791, June 2008.