

Prediction of Traffic on the Network at Client End Using Signal Processing

V.Vijayadeepa¹, K.Baby²Head of the Department, Dept. of CS, Muthayammal College of Arts & Science, Rasipuram, Namakkal, India¹Research Scholar, Dept. of CS, Muthayammal College of Arts & Science, Rasipuram, Namakkal, India²

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.

KEYWORDS: LTI, Shortest path, air index, broadcasting.

I. INTRODUCTION

Shortest path computation is an important function in modern car navigation systems and has been extensively studied. 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). Nowadays, several online services provide live traffic data (by analyzing collected data from road sensors, traffic cameras, and crowd sourcing 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 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 (called result transmission model). However, given the rapid growth of mobile devices and services, this model is facing scalability limitations in terms of network bandwidth and server loading. According to the Cisco Visual Networking Index forecast, global mobile traffic in 2010 was 237 petabytes per month and it grew by 2.6-fold in 2010, nearly tripling for the third year in a row. Based on a telecommunication expert

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

, the world's cellular networks need to provide 100 times the capacity in 2015 when compared to the networks in 2011. Furthermore, live traffic are updated frequently as these data can be collected by using crowd sourcing techniques (e.g., anonymous traffic data from Google map users on certain mobile devices). As such, huge communication cost will be spent on sending result paths on the this model. Obviously, the client-server architecture will soon become impractical in dealing with massive live traffic in near future. Ku et al. raise the same concern in their work which processes spatial queries in wireless broadcast environments based on Euclidean distance metric. Malviya et al. developed a client-server system for continuous monitoring of registered shortest path queries. For each registered query (s, t), the server first pre-computes K different candidate paths from s to t. Then, the server periodically updates the travel times on these K paths based on the latest traffic, and reports the current best path to the corresponding user. Since this system adopts the client-server architecture, it cannot scale well with a large number of users, as discussed above. In addition, the reported paths are approximate results and the system does not provide any accuracy guarantee. 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 in constitutes a set of pairwise minimum and maximum traveling costs between every two subpartitions 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. As reported in , the re-computation time of the index takes 2 hours for the San Francisco (CA) road map. It is prohibitively expensive to update the index for OSP, in order to keep up with live traffic circumstances. The rest of the paper is organized as follows. We first introduce four main performance factors for evaluating OSP and overview the state-of-the-art shortest path computation methods in Section . The system overview and objectives of our live traffic index are introduced in Section The LTI construction, LTI transmission, and LTI maintenance are subsequently discussed. We summarize our complete framework in Section 7 and evaluate LTI thoroughly in Section Finally, our work is concluded .

II. RELATED WORK

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. Having downloaded the raw traffic data from the broadcast channel, the following methods either directly calculate the shortest path or efficiently maintain certain data structure for the shortest path computation. Uninformed search (e.g., Dijkstra's algorithm) traverses graph nodes in ascending order of their distances from the source s, and eventually discovers the shortest path to the destination t. Bi-directional search (BD) reduces the search space by executing Dijkstra's algorithm simultaneously forwards from s and backwards from t. As to be discussed shortly, bi-directional search can also be applied on some advanced index structures. However, the response time is relatively high and the clients may receive large amount of irrelevant updates due to the transmission model. 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. ALT and arc flags (AF) are two representative algorithms in this category. ALT makes use of search, landmarks, and triangle inequality . A few landmark nodes are selected and the distances between each landmark and every node are pre-computed. These pre-computed distances can be exploited to derive distance bounds for A_ search on the graph. Delling and Wagner proposes a lazy update paradigm for ALT (DALT) so that it can tolerate certain extents of edge weights changes on a dynamic graph. The distance bounds derived from the pre-computed information remain correct if no edge weight becomes lower than the initial weight used at the ALT construction. This lazy update paradigm significantly reduces the index maintenance cost.

III. EXISTING SYSTEM

Nowadays, several online services provide live traffic data (by analyzing collected data from road sensors, traffic cameras, and crowd sourcing 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 traffic data;

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

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

IV. PROPOSED SYSTEM

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. Having downloaded the raw traffic data from the broadcast channel, the following methods either directly calculate the shortest path or efficiently maintain certain data structure for the shortest path computation. Uninformed search (e.g., Dijkstra's algorithm) traverses graph nodes in ascending order of their distances from the source s , and eventually discovers the shortest path to the destination t . Bi-directional search (BD) reduces the search space by executing Dijkstra's algorithm simultaneously forwards from s and backwards from t . As to be discussed shortly, bi-directional search can also be applied on some advanced index structures. However, the response time is relatively high and the clients may receive large amount of irrelevant updates due to the transmission model. 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. ALT and arc flags (AF) are two representative algorithms in this category. ALT makes use of A_* search, landmarks, and triangle inequality. A few landmark nodes are selected and the distances between each landmark and every node are pre-computed. These pre-computed distances can be exploited to derive distance bounds for A_* search on the graph. Delling and Wagner proposes a lazy update paradigm for ALT (DALT) so that it can tolerate certain extents of edge weights changes on a dynamic graph. The distance bounds derived from the pre-computed information remain correct if no edge weight becomes lower than the initial weight used at the ALT construction. This lazy update paradigm significantly reduces the index maintenance cost.

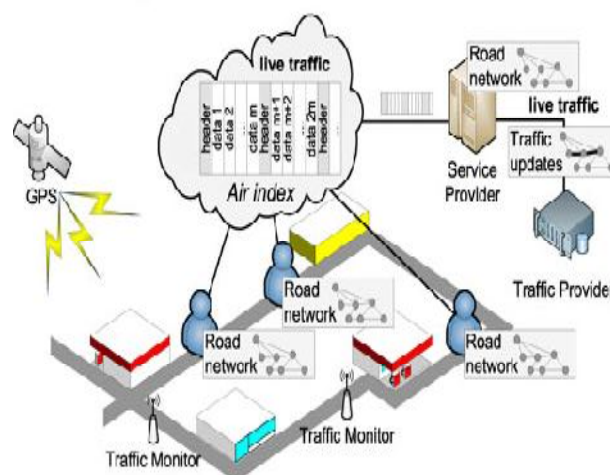


Fig1. LTI System Overview.

ADVANTAGES OF PROPOSED SYSTEM

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

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.

V. ALGORITHMS

Stochastic Based Index Construction:

The graph partitioning framework only returns a binary tree index I_{bi} that is constructed based on Cheeger cut sequences. However, I_{bi} only fulfills the first two objectives (i.e., minimizing the overhead $jDSG_j$ subject to g). Intuitively, the size of search graphs G_q is highly relevant to the index hierarchical structure. As a motivating example, the number of relevant entries of q_{db} ; dP is reduced from 9 to 8 if we remove one index node from the index tree. The new index and the relevant entries are illustrated. Given the size of leaf entries g , minimizing the size of search graph can be viewed as a problem of finding the best hierarchical index structure for potential queries. Finding the optimal hierarchical structure is challenging since (1) the performance of an index cannot be easily estimated (which should be estimated by a query workload Q or a universal query set U) and (2) the index statistics (e.g., shortcuts) are changed on different index hierarchical structures (which is necessarily recalculated based on the structure). This problem is similar to those combinatorial optimization problems (e.g., hierarchical clustering) that groups data into a set of hierarchical partitions such that the objective function is optimized. Typically, these combinatorial problems are solved by approximate solutions under reasonable response time. Thus, we propose a top-down approach that greedily decides the structure based on a stochastic estimation.

Client Algorithm

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 G_q 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. These issues are addressed. The client then wait for those segments, read them, and update the weight of G_q . Subsequently, G_q is used to compute the shortest path in the client machine locally. Note that Algorithm is kept running in order to provide online shortest path until the client reaches to the destination.

Service Algorithm

The first step is devoted to construct the live traffic index; they are offline tasks to be executed once only. The service provider builds the live traffic index by partitioning the graph G into a set of subgraphs $fSGig$ such that they are ready for broadcasting. We develop an effective graph partitioning algorithm for minimizing the total size of subgraphs and study a combinatorial optimization for reducing the search space of shortest path queries. In each broadcasting cycle, the server first collects live traffic updates from the traffic provider, updates the subgraphs $fSGig$, and eventually broadcasts them.

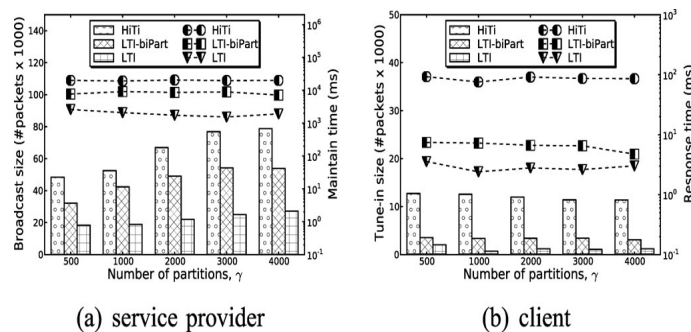


Fig 2 Varying Number Of Partitions,

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

VI. IMPLEMENTATION

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. In this work, we focus on handling traffic updates but not graph structure updates. For real road networks, it is infrequent to have graph structure updates (i.e., construction of a new road) when compared to edge weight updates (i.e., live traffic circumstances). Thus, we assume that the graph structures are distributed to every client in advance (e.g., by monthly updates or at system boot-up) via typical transmission protocol (i.e., HTTP and FTP). In Fig. 4, we illustrate the components and system flow in our LTI framework. The components shaded by gray color are the core of LTI. In order to provide live traffic information, the server maintains (component a) and broadcasts (component b) the index according to the up-to-date traffic circumstances. In order to compute the online shortest path, a client listens to the live traffic index, reads the relevant portions of the index (component c), and computes the shortest path (component d).

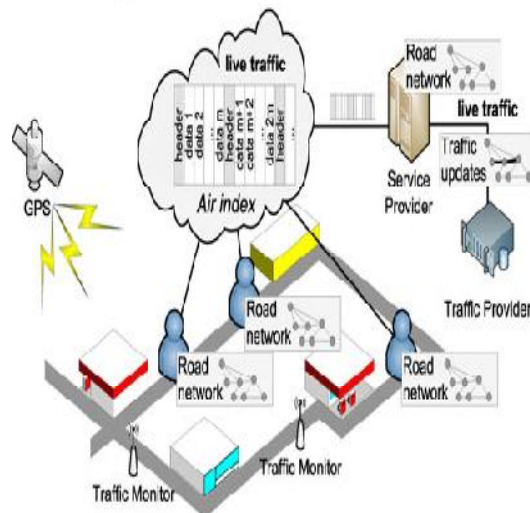


Fig 3 LTI Overview

VII. RESULT ANALYSIS

We demonstrate how the methods perform at service provider. Fig the broadcast size and maintenance time of the methods by varying d, s, b, and v. For all testings, LTI is superior to CH in terms of maintenance time but produces around 40 percent more packets than CH.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

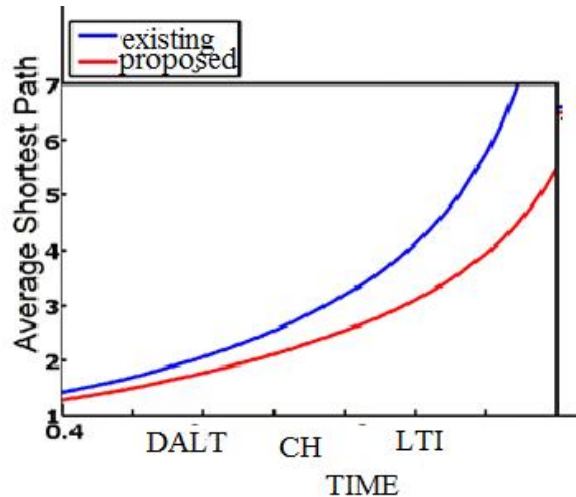


Fig 4 Result graph

A more promising result is that the maintenance time of LTI is no longer sensitive to the update ratio when $d > 20\%$. This is because most of BSPTs are necessarily updated when the update ratio is around 20 percent. The subsequent updates ($> 20\%$) are more likely some incremental work in updating the BSPTs (i.e., traversing few more edges by the Dijkstra like algorithm) so that it becomes less sensitive to d . To express the comparison in absolute terms, we show the time it takes to broadcast over a 1.92 Mbps (WCDMA) and a 14 Mbps (HSDPA) channel in Table 5, which are typical transmission rates in 3G networks and 3.5G networks. LTI takes 11.6 and 3.32 s to complete maintenance and broadcast cycle at WCDMA and HSDPA, respectively; while CH takes 12.25 and 6.35 s to complete the same cycle, respectively. In addition, DALT and CH require the clients to tune-in the broadcast channel for 5 and 0.7 s over WCDMA and HSDPA, respectively, which significantly affects the number of simultaneous services in the wireless broadcast environments. Although DALT does not bother any maintenance cost at service provider, the tune-in cost and response time of DALT makes it infeasible on the live traffic circumstance.

VIII. CONCLUSION AND FUTURE WORK

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.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

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] "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 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.
- [16] N. Malviya, S. Madden, and A. Bhattacharya, "A Continuous Query System for Dynamic Route Planning," Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), pp. 792-803, 2011.
- [17] G. Kellaris and K. Mouratidis, "Shortest Path Computation on Air Indexes," 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 ACM SIGSPATIAL 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, "Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation," IEEE Trans. Knowledge and Data Eng., vol. 10, no. 3, pp. 409-432, May 1998.
- [21] 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, Sept. 2002.
- [22] E.P.F. Chan and Y. Yang, "Shortest Path Tree Computation in Dynamic 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 Schemes for 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-Point Shortest Paths from External Memory," Proc. SIAM Workshop Algorithms Eng. and Experimentation and the Workshop Analytic Algorithmics and Combinatorics (ALENEX/ANALCO), pp. 26-40, 2005.
- [26] M. Hilger, E. Köhler, R. Möhring, and H. Schilling, "Fast Point-to-Point Shortest Path Computations with Arc-Flags," The Shortest Path 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 Dynamic Graphs," Proc. Sixth Int'l Workshop Experimental Algorithms (WEA), pp. 52-65, 2007.
- [29] G. D'Angelo, D. Frigioni, and C. Vitale, "Dynamic Arc-Flags in Road Networks," Proc. 10th Int'l Symp. Experimental Algorithms (SEA), pp. 88-99, 2011.
- [30] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks," Proc. Seventh Int'l Workshop Experimental Algorithms (WEA), pp. 319-333, 2008.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

BIOGRAPHY



Mrs.V.Vijayadeepa M.Sc.,M.C.A.,M.Phil.,[Ph.D].,received her B.Sc from University of Madras and M.Sc and MCA from Periyar University.She has completed her M.Phil at Bharathidasan University. She is having 10 Years of experience in collegiate teaching and she is the Associate Professor and Head of Department of Computer Science and applications in Muthayammal College of Arts and Science, Rasipuram Affiliated by Periyar University. Her main research interests include personalized web, web information retrieval, data mining, data warehousing and information systems.



K.BABY. She received B.sc Computer Science degree from Thiruvalluvar govt arts college, periyar university , Rasipuram ,and Tamilnadu, India. She received Master of Computer Science degree from muthayammal college of arts and science,rasipuram,periyar university,salem ,Tamilnadu, India. She is Pursuing M.Phil (full time) degree from Muthayammal College of Arts & Science, in Periyar University Salem,Tamilnadu, India. She interested research area is Datamining.