

Geo-calendar based Predictive Skyline Queries using Fuzzy Inference Engine

K.B. Priya Iyer

Research Scholar, Sathyabama University
Associate Professor, MOP Vaishnav College for
Women(Autonomous)

V. Shanthi

Professor, Department of M.C.A
St. Joseph College of Engineering

ABSTRACT

Embedding Information technology with Transportation system creates a new era in building Intelligent Transportation System (ITS). An effective Intelligent Transportation System reduces traffic congestion, environmental pollution, fuel consumption and driver in-convenience etc. The widespread adoption of GPS-enabled mobile devices has opened new possibilities of developing an ITS. With advancements in Wireless technologies, GIS and sensors, spatial route search are important class of queries under location based services. In this paper, we introduce a new spatial query called Predictive Skyline query (PSQ) on time dependent road networks. The PSQ is a spatial route search query for future journey schedule which are popular *apps under* Intelligent Transportation Systems. The algorithm consists of five phases namely PSQuery Initiator (QI), Fuzzy Travel Time Forecaster (FTF), PSQuery Executor and Optimizer (QEO), Traffic Fuzzy Controller (TFC), PSQuery Recorder (QR). The FTF predicts the travel time based on historical data using fuzzy inference rules. The TFC predicts the overall traffic congestion in the user specified spatial region on the scheduled journey date. The experimental evaluation reflects the accuracy of travel time prediction in real road networks and efficiency of algorithm in processing PSQ queries.

Keywords

Predictive Skyline queries, Traffic Congestion, Fuzzy Controller, Fuzzy rules, Traffic Prediction, Spatial Databases, Skyline Queries, GIS, Location based services, GPS.

1. INTRODUCTION

The widespread adoption of modern technologies (GPS and GIS) has brought the world of communications in the hands of users. The emerging trends in these technologies made mobile users geo-savvy providing location based services no matter where they are. In modern geographic information systems, skyline queries represent an important class in spatial route search. For example, a skyline query can be tourists who wish to search for a low priced hotel near-by a hospital and a beach.

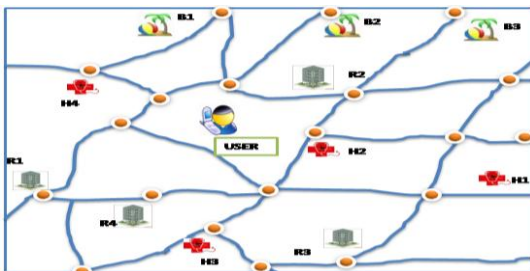


Figure 1: Road Network

(in above Figure3: R1,R2,R3 – restaurants, H1,H2,H3 – hospitals, T1,T2,T3 – temples, B1,B2,B3 – beach).

All the previous studies gives the route planning trips based on network distances instead of travel time. Thus Predicted Skyline Query is finding the set of data objects by considering multiple query points to user specified location on a future journey schedule basing on travel time. Travel time information plays an important role in a variety of real-time transportation applications. These real-time applications include Advanced Traveler Information Systems (ATIS), Route Guidance Systems (RGS), and Advanced Traffic Management Systems (ATMS), performance monitoring, Congestion management, Travel demand modeling and forecasting, Traffic simulation etc., which are part of the Intelligent Transportation Systems (ITS). However travel time is affected by a number of factors such as road conditions, climatic conditions, driver attitude and country's cultural habits. Understanding traffic factors that affects the travel time is essential for improving prediction accuracy in related travel time studies. The PSQ performs skyline search for future journey schedule where travel time prediction is done by fuzzy inference engine by considering factors such as significance of day, journey time. We also predict the traffic congestion on user specified spatial region by using a Traffic Fuzzy Controller.

To sum up we make the following contributions:

1. Finds data objects for Predictive Skyline queries for future journey schedule by taking travel time as metric.
2. Travel time prediction on specified route is computed by Fuzzy Travel Time Forecaster (FTF) using fuzzy inference rules and historical data.
3. Designing efficient algorithm for finding skyline objects thru five phases namely PSQuery Initiator (QI), Fuzzy Travel Time Forecaster (FTF), PSQuery Executor and Optimizer (QEO), Traffic Fuzzy Controller (TFC), PSQuery Recorder (QR).
4. The Traffic Fuzzy Controller gives the status of overall traffic congestion on the route for user specified future journey schedule.
5. Developed cache technique (QR) which reduces the computation cost of future predictive skyline queries.

The reminder of this paper is organized as follows. In section 2, we review the related work on both skyline queries and developing intelligent transportation System. In section 3, we formally defined Predictive Skyline query in Road Networks. In section 4, we introduce Fuzzy based approach for predictive skyline data objects in road networks. Section 5 presents the results of our experimental evaluation of our proposed approaches with a variety of Spatial Network with large number of data and query objects. Finally section 6 concludes the paper with future research.

2. RELATED WORK

2.1 Skyline query

The *skyline* operator was first introduced into the database community by Borzsonyi et al. [2]. Borzsonyi et al. [2] propose the Block-Nested-Loops algorithm (BNL) and the Extended Divided-and-Conquer algorithm (DC). Both algorithms processes the entire object set for retrieving the skyline data. Kossmann et al. [6] propose an online nearest neighbor skyline query processing method which can progressively report the skyline points in an order according to user's preference. The objects in the dominated subspace are pruned, and the objects in each non-determined subspace form a to-do list. To remedy this problem, Papadias et al. [12] propose an R-tree based algorithm, Brand and Bound Skyline (BBS), which retrieves skyline points by browsing the R-tree based on the best-first strategy. BBS only visits the intermediate nodes not dominated by any determined skyline points. This method has more efficient memory consumption than the method in [6].

There are many variants of the traditional skyline query. Pei et al. [14] and Yuan et al. [20] proposed methods to compute skylines in all possible subspaces. Tao et al. [18] gave an efficient algorithm to calculate skylines in a specific subspace. In the context of uncertain databases, Pei et al. [13] proposed the probabilistic skyline over uncertain data, which returns a number of objects that are expected to be skylines with probability higher than a threshold. The most relevant problems to our work are the dynamic skyline[12], spatial skyline [15], multi-source skyline on road networks [3], multi-preference path planning approach[7] and continuous probabilistic skyline queries over uncertain data streams[4]. Specifically, Papadias et al. [12] applies BBS algorithm to retrieve skyline points, where dynamic attributes of data objects are computed by a set of dimension functions. However, only Euclidean distance was considered for dimension functions. Similarly, the method proposed for multi-source skyline on road networks [3] also utilizes geometric information of data objects during the pruning, which is thus limited to road network application.

2.2 Travel time Studies

There has been a significant amount of work identifying the importance of accurate travel time predictions in a transportation system. From the travelers' perspective, accurate travel time predictions reduce the uncertainty in decision making about departure time and route choice, which in turn reduce travelers' stress and anxiety. From the operators' point of view, travel time prediction models may be used to determine the reliability of a transportation system. Consequently, travel time prediction methods are central to Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS) [19][1]. Bus travel times are the result of nonlinear and complex interactions of many different constituent factors influencing either demand (e.g. passenger's demand or traffic flow) or capacity (e.g. accidents, weather condition, route characteristics) [9]. Mean-variance estimation method was also used in [10] to predict bus travel time variability. Travel Time Prediction Using floating Car Data Applied to Logistics Planning by taking into account that routes of long-range trips are not completely given in advance but are rather unknown and subject to change[16]. short-term prediction of highway travel times, which represent an accurate estimation of the

expected travel time for a driver commencing on a particular route based on the fusion of different types of data that come from different sources (inductive loop detectors and toll tickets) and from different calculation algorithms[17]. Car following model using a fuzzy inference system (FIS) to simulate and predict the future behavior of a Driver-Vehicle-Unit (DVU) based on a new idea for estimating the instantaneous reaction of DVU, as an input of fuzzy model[5].

In summary, previous studies on skyline variants are limited to either Euclidean space or metric space for a specific application. In contrast, our work focuses on the travel time as a metric for skyline search which play a vital role in road networks. Travel time prediction is based on previous traffic data considering factors such as time slice of the day, day significance etc using fuzzy inference rules. It also helps future queries computation time by caching technique.

3. SYSTEM MODEL

In this section, we describe the road network and system model; define the predictive skyline query search in spatial networks. We assume a spatial network [California Road Network], containing set of static data objects as well as query objects searching their skyline objects. We assume all road maps and daily traffic data are maintained by cloud server.

2.3 Road Network

We model the underlying road network as a weighted undirected graph $G = (V, E)$ where E is an Edge set of road segments in the road network, V is the Vertex set of intersection points of the road segment and each edge is given travel time of its corresponding road segment as weights.

In this model (Figure 1), we consider our system with a mobile environment in which mobile user is able to communicate with the service provider through wireless communication infrastructure e.g.: Wi-Fi.

2.4 Definition

A Predictive Skyline query is a skyline query on Time dependent road networks where data objects are returned in the relative to the user location and travel time is predicted from historical data. The PSQ is useful in following situations:

Example1: if a user wants to find a serviced apartment close to a hospital and a restaurant.

Query Q1: $\Pi_{\text{poi_category}=\text{"Apartment"}} (g_{\text{min(traveltime)}} \sigma_{\text{poi_category}=\text{"Hospital"}(\text{poi_master})) \cap g_{\text{min(traveltime)}} \sigma_{\text{poi_category}=\text{"Restaurant"}(\text{poi_master}))$

From the Fig: 1, the algorithm returns serviced apartment (A3) object which is close to hospital (H4) and restaurant (R1) based on the travel time.

Example2: if a tourist wants to find a low price restaurant nearer to a temple and close to a beach. From the Fig: 1, the algorithm returns restaurant (R1) object which is nearer to temple (T3) and close to beach (B1) based on the travel time.

Query Q2: $\Pi_{(\text{poi_category}=\text{"Restaurant"} \cup g_{\text{min(price)}})} (g_{\text{min(traveltime)}} \sigma_{\text{poi_category}=\text{"Temple"}(\text{poi_master})) \cap g_{\text{min(traveltime)}} \sigma_{\text{poi_category}=\text{"Beach"}(\text{poi_master}))$
in both the queries,

$\sigma_p(r) = \{t \mid t \in r \text{ and } p(t)\}$, p is a formula in propositional calculus consisting of terms connected by : \wedge (and), \vee (or), \neg (not). Each term is one of: $\langle \text{attribute} \rangle$ op $\langle \text{attribute} \rangle$ or $\langle \text{constant} \rangle$ where op is one of: $=$, \neq , $>$, \geq , $<$, \leq . $\prod A_1, A_2, \dots, A_k (r)$, where A_1, A_2 are attribute names and r is a relation name.

g is an aggregate operation in relational algebra: $G_1, G_2, \dots, G_n g F_1(A_1), F_2(A_2), \dots, F_n(A_n)(E)$. here E is any relational-algebra expression, G_1, G_2, \dots, G_n is a list of attributes on which to group (can be empty), Each F_i is an aggregate function, Each A_i is an attribute name.

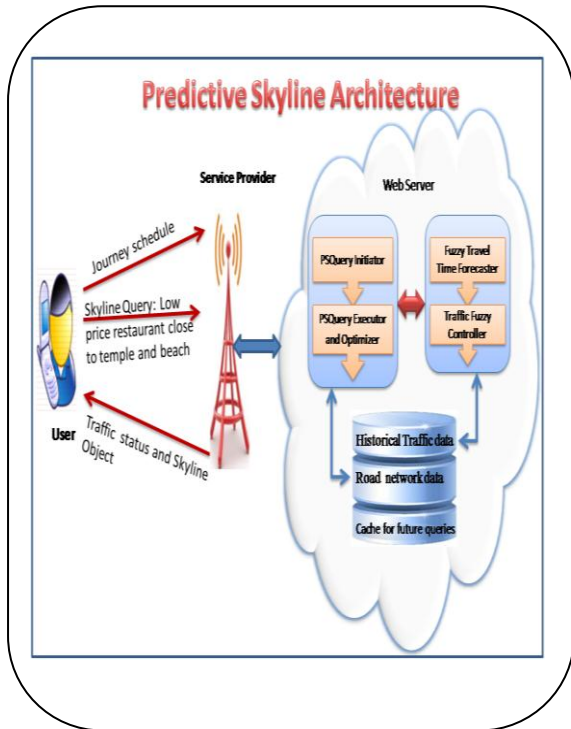


Figure 2: Architecture – Predictive Skyline

4. ALGORITHM

4.1 Phase I: PSQuery Initiator

The algorithm first finds the nearest vertex of the query origin. The user location is given by GPS / Wi-Fi or any location specified by the user. The GetVertex function returns the nearest neighboring road node “u” of the user location. Harvesian Formula is used in finding the distance between two locations. The nodes in database are sorted in descending order of their distances from user location “u” where the top most nodes gives the nearest neighbour of “u”.

4.2 Phase II: Fuzzy Travel Time Forecaster (FTF)

During this phase, the travel time for future journey date is predicted using fuzzy inference rules. The travel schedule includes both date of journey and time of journey. We know traffic depends on various factors like day of travel, time, weather condition etc. we assume travel time depends on day significance, time of journey only. We design a Fuzzy Travel Time Forecaster inference engine in MATALB. The Day

significance is categorized as HOLIDAY, MONDAY, MID-WEEKDAY, FRIDAY,

WEEKEND-SATURDAY, WEEKEND-SUNDAY. The journey time selection is based on following inference rules:

If traveltime is between 4hrs-7.20hrs then traffictype is EARLY MORNING HRS(EMH)

If traveltime is between 7hrs-8.10hrs then traffictype is MORNING SCHOOL HRS (MSH)

If traveltime is between 8hrs-11.30hrs then traffictype is MORNING OFFICE HRS (MOH)

If traveltime is between 11hrs-13.30hrs then traffictype is SHOPPING HRS (SPH)

If traveltime is between 13hrs-15.15hrs then traffictype is NOON OFF PEAK HRS (OPH)

If traveltime is between 15hrs-17.10hrs then traffictype is EVENING SCHOOL HRS (ESH)

If traveltime is between 17hrs-22.15hrs then traffictype is EVENING OFFICE HRS (EOH)

If traveltime is between 22hrs-4.15hrs then traffictype is LATE NIGHT HRS (LNH)

Figure 3: Fuzzy Travel Time Forecaster

The function DateSignificance returns the date significance(holiday or weekday or weekend etc). The function TraveltimeMap returns the slice of travel time from inference rules.

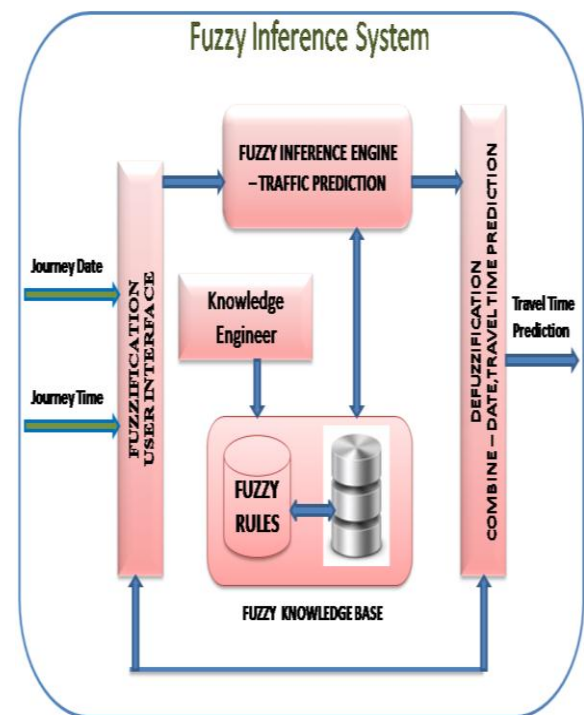


Figure 4: Fuzzy Inference System

Phase 4.3: PS Query executor and Optimizer

From the nearest node, all adjacent nodes are traversed to find the list of first user preferred query point. While searching the query points, the query filters out the unwanted data points that are out of range of user preferences (Filter phase). The algorithm proceeds by applying the other query preferences on the list of candidate data objects (Heap phase). All the candidate objects are refined by considering the minimum travel time and other factors (Skyline-Refine phase). At the end, the algorithm returns the goal directed skyline object relative to user location.

Optimization is achieved by computing the travel times to selected hotels which have the potential to participate in the final query answer instead of all hotels as shown in table 1 and 2. Query Q1 (list of all low cost hotels) before and after optimization is compared. This approach would enhance the query processing by avoiding redundant extensive computations. Aggregations for secondary query points are done before applying skyline retrieval. This helps in further reduction of query computation.

Phase 4.4: PS Query Traffic Fuzzy Controller

The TFC predicts the overall traffic congestion status to the user given spatial region.

4.5 Phase 5: PSQuery Recorder

Each time a query is received, the dynamic skyline cache (DSC) stores the query point. If the cache is exceeded in size, the least frequently used (LFU) query is evicted. This is done by incrementing a counter to each skyline w.r.t query point and also considering the traffic update. The cache page will also be replaced when there is a drastic change in traffic.

Algorithm 1 : Predictive Skyline

PSQ (uloc,pref[],A[],journeyschedule)
 /* upos: query origin (latitude,ongitude) , pref[], an array consisting of user query preferences Q1,Q2,...Qn and A[], an array of their corresponding attributes a1,a2,...an, journey schedule: date and travel time of journey */

1. Find the nearest vertex for the query origin uloc
 $u \leftarrow \text{Getvertex}(uloc/loc)$
2. $C[i]=\{0\}$ /*set the initial candidate objects as zero */
3. $\text{Predicted_doj} \leftarrow \text{fuzzymatlab_datepredictor}()$
4. $\text{Predicted_time} \leftarrow \text{fuzzymatlab_traveltimepredictor}()$
5. $C[i] \leftarrow \text{Generate}(Q1)$ /* retrieves the list of initial candidate set for the query point Q1*/
6. $\text{Heapset}[i] \leftarrow \text{addQToheap}(C[i])$
7. if (checkpredictiveskyline(Heapset[i])) then
8. $\text{Dispskylineobject}()$
 /* display skyline object */
9. $\text{Queryrecorder}(C[i], \text{Heapset}[i])$
10. /* store for future queries*/
11. end if

5. EXPERIMENTAL EVALUATION

5.1 Experimental Setup

We conducted experiments on California road network which contains 21,050 nodes and 21693 edges. The algorithm is implemented in Java and tested on Windows Platform with Intel Core 2 CPU and 80GB memory. The main metric we adopt is CPU time that reflects how much time the algorithm takes in processing a skyline query. The input map is extracted from Tiger/Line files that are publicly available [21].

Table 1: Query Q1 (list of all low cost hotels before optimization)

| MONTH | DAY | DAY SIGNIFICANCE | TIME SLICE | KEY MAP VALUE |
|-------|-----|------------------|-------------------------|---------------|
| 1 | 1 | HOLIDAY | EARLY MORNING HRS | TT_1 |
| 1 | 1 | HOLIDAY | MORNING SCHOOL HRS | TT_2 |
| 1 | 1 | HOLIDAY | MORNING OFFICE HRS | TT_3 |
| 1 | 1 | HOLIDAY | SHOPPING HRS | TT_4 |
| 1 | 1 | HOLIDAY | NOON OFF PEAK HRS | TT_5 |
| 1 | 1 | HOLIDAY | EVENING SCHOOL HRS | TT_6 |
| 1 | 1 | HOLIDAY | EVENING OFFICE HRS | TT_7 |
| 1 | 1 | HOLIDAY | LATE NIGHT OFF PEAK HRS | TT_8 |
| 1 | 2 | WEEKEND-SUNDAY | EARLY MORNING HRS | TT_9 |
| 1 | 2 | WEEKEND-SUNDAY | MORNING SCHOOL HRS | TT_10 |
| 1 | 2 | WEEKEND-SUNDAY | MORNING OFFICE HRS | TT_11 |
| 1 | 2 | WEEKEND-SUNDAY | SHOPPING HRS | TT_12 |
| 1 | 2 | WEEKEND-SUNDAY | NOON OFF PEAK HRS | TT_13 |
| 1 | 2 | WEEKEND-SUNDAY | EVENING SCHOOL HRS | TT_14 |
| 1 | 2 | WEEKEND-SUNDAY | EVENING OFFICE HRS | TT_15 |
| 1 | 2 | WEEKEND-SUNDAY | LATE NIGHT OFF PEAK HRS | TT_16 |

Table 2: Query Q1 (list of all low cost hotels after optimization)

| START NODE ID | END NODE ID | TT_1 | TT_2 |
|---------------|-------------|-----------|-----------|
| 0 | 1 | 0.0000992 | 0.0001002 |
| 0 | 6 | 0.0001026 | 0.0001058 |
| 9 | 10 | 0.0000491 | 0.0000503 |
| 93 | 94 | 0.000205 | 0.000235 |
| 978 | 1185 | 0.0002525 | 0.0002605 |
| 9787 | 9835 | 0.0001288 | 0.0001321 |
| 9788 | 9789 | 0.0001586 | 0.0001625 |

5.2 Results

5.2.1 Travel time prediction for Query route Q1

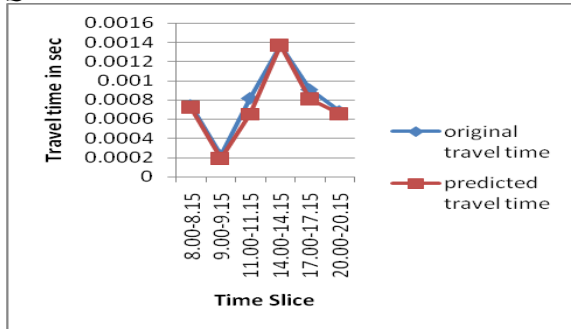


Figure 5: (a)

With this experiment in Fig 5(a), we show the travel time of original and predicted values for Query Q1.

5.2.2 Travel time prediction for Node n1

In Fig 5(b), we show the travel time taken for any one node during the varying time slices.

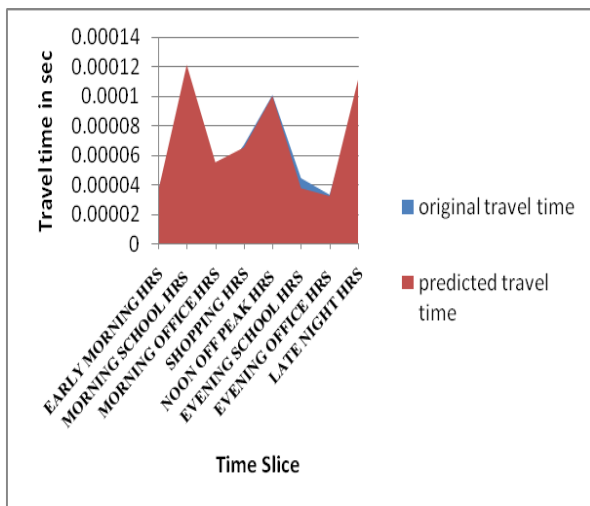


Figure 5: (b)

5.2.3 Traffic Status for Query route Q1

| Time Slice | Original Travel time | Predicted Travel time |
|--------------------|----------------------|-----------------------|
| EARLY MORNING HRS | NORMAL | LOW |
| MORNING SCHOOL HRS | LOW | PEAK |
| MORNING OFFICE HRS | PEAK | PEAK |
| SHOPPING HRS | PEAK | PEAK |
| NOON OFF PEAK HRS | LOW | WEEK |
| EVENING SCHOOL HRS | PEAK | PEAK |
| EVENING OFFICE HRS | PEAK | PEAK |
| LATE NIGHT HRS | LOW | WEEK |

Figure 5: (c)

Fig 5(c), we show the predicted travel time in all time slice.

5.2.4 Impact of cache in query computation

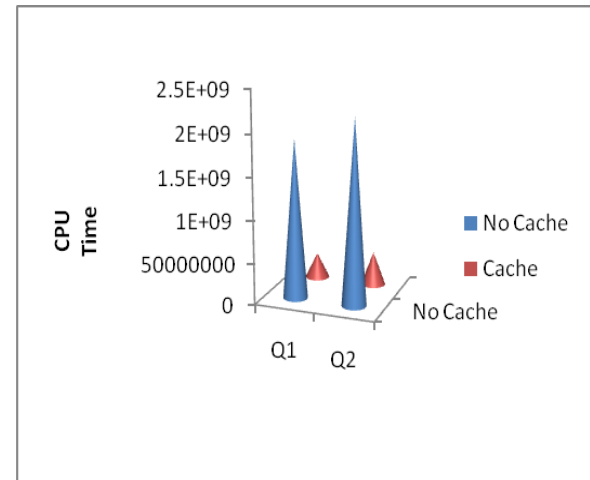


Figure 5: (d)

Fig 5(d), we show the impact of PSQuery Recorder technique in computing future queries.

5.2.5 Impact of query optimization on dominant objects

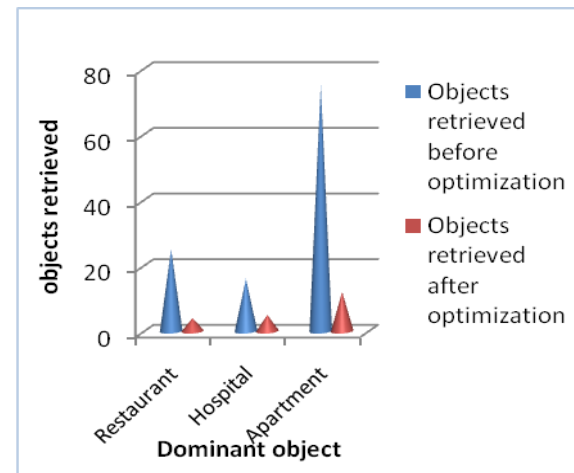


Figure 5: (e)

Fig 5(e), we show the impact of query optimization in object retrieval from database.

6. CONCLUSION

In this paper, we propose a Predictive Skyline queries on travel time dependent road networks. This algorithm efficiently searches and computes the data object to query origin. The travel time is predicted using historical data based on fuzzy inference rules. Traffic Fuzzy Controller predicts the traffic congestion on the day of travel. The algorithm also caches the skylines to reduce the computation cost for future queries. Additional future works includes skyline query for moving objects, different types of transportation mode, road types and taking weather conditions, day light for travel time prediction.

7. REFERENCES

- [1] B. Bartin and K. Ozbay. 2010. Determining the optimal configuration of highway routes for real-time traffic information: A case study. *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 225–231.
- [2] S. Borzsonyi, D. Kossmann, and K. Stoker. 2001. The skyline operator. *ICDE*.
- [3] K. Deng, X. Zhou, and H. T. Shen. 2007. Multi-source skyline query processing in road networks. In *Proc. 23th Int. Conf. on Data Engineering*, pages 796–805.
- [4] Hui Zhu Su, En Tzu Wang and Arbee L. P. Chen. 2011. Continuous probabilistic skyline queries over uncertain data streams. *Database and Expert Systems Applications*, Springer.
- [5] Khodayari A, Kazemi R, Ghaffari A, Brauningl R. 2011. Design of an improved fuzzy logic based model for prediction of car following behavior. *IEEE International Conference on Mechatronics (ICM)*, On Page(s): 200 – 205.
- [6] D. Kossmann, F. Ranmsak, and S. Rost. 2002. Shooting stars in the sky: An online algorithm for skyline queries. *VLDB*.
- [7] Kriegel, H.-P.; Renz, M.; Schubert, M. 2010. Route skyline queries: A multi-preference path planning approach. in *IEEE 26th International Conference on Data Engineering (ICDE)*.
- [8] X. Lin, Y. Yuan, W. Wang, and H. Lu. 2005. Stabbing the sky: Efficient skyline computation over sliding windows. *ICDE*.
- [9] E. Mazloumi, G. Currie, and G. Rose. 2010. Using GPS data to gain insight into public transport travel time variability,” *Journal of Transportation Engineering*, vol. 136, no. 7, pp. 623–631.
- [10] E. Mazloumi, G. Rose, G. Currie, and S. Moridpour. 2010. Prediction intervals to account for uncertainties in neural network predictions: Methodology and application in bus travel time prediction, *Engineering Applications of Artificial Intelligence*.
- [11] Mehdi Sharifzadeh, Cyrus Shahabi, and Leyla Kazemi. 2009. Processing spatial skyline queries in both vector spaces and spatial network databases, *ACM Transactions on Database Systems*, vol. 34, no. 3, pp. 1–43.
- [12] D. Papadias, Y. Tao, G. Fu, and B. Seeger. 2005. Progressive skyline computation in database systems. *ACM Trans. Database Syst.*, 30(1):41–82.
- [13] J. Pei, B. Jiang, X. Lin, and Y. Yuan. 2007. Probabilistic skylines on uncertain data. In *Proc. 33th Int. Conf. on Very Large Data Bases*, pages 15–26.
- [14] J. Pei, W. Jin, M. Ester, and Y. Tao. 2005. Catching the best views of skyline: a semantic approach based on decisive subspaces. In *Proc. 31th Int. Conf. on Very Large Data Bases*, pages 253–264.
- [15] M. Sharifzadeh and C. Shahabi. 2006. The spatial skyline queries. *VLDB*.
- [16] Simroth, A. Zähle, H. Travel Time Prediction Using Floating Car Data Applied to Logistics Planning. *IEEE Transactions on Intelligent Transportation Systems*, Volume: 12, Issue: 1, On Page(s): 243 – 253.
- [17] F. Soriguera F. Robusté. 2011. Highway travel time accurate measurement and short-term prediction using multiple data sources. *Transportmetrica*, Volume 7, Issue 1, pages 85–109.
- [18] Y. Tao, X. K. Xiao, and J. Pei. 2006. SUBSKY: Efficient computation of skylines in subspaces. In *Proc. 22th Int. Conf. on Data Engineering*, page 65.
- [19] M. Yang, Y. Liu, and Z. You. 2010. The reliability of travel time forecasting. *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 162–171.
- [20] Y. Yuan, X. Lin, Q., W. Wang, J. Xu Yu, and Q. Zhang. 2005. Efficient computation of the skyline cube. In *Proc. 31th Int. Conf. on Very Large Data Bases*.
- [21] Tiger/Line: www.census.gov/geo/www/tiger/.