A Survey on Path Queries

Sithara M P[1], Varsha C[2]

[1]M.Tech in Computer science and Engineering, KMCT College of Engineering, Kerala, India

[2]Assistant Professor, Department of Computer Science and Engineering, KMCT College of Engineering, Kerala, India

Abstract: ***In large scale network shortest distance query is used to find the shortest path. When we consider road network, route search and optimal path queries are two important types of queries. A path query returns a path that is a set of points that connects the source and destination. The optimal path queries find the optimum path from set of information. In the case of road network users give some specification about the travelling with or without constraints. The optimal path queries optimize the possible paths and give the optimal path that satisfies all the constraints. The road network mainly deals with time dependent parameters and we use different models to represent the road network .This survey paper discusses about various path query processing algorithms in road network and the two models that is used to represent the road network.***

Keywords: Path Queries, Road Network, Optimum Path, Travel Time

1. Introduction

Standard algorithms for finding shortest path in graph assume that costs are deterministic. The most commonly used algorithm to find the shortest path is Dijkstra's algorithm [1]. The optimal route query processing is mainly used in the road network. A road networks is represented by a large graph in 2-dimentional space. The edge is considered as a road segment. Various queries have been proposed to find the optimal path in road network. The optimal path queries find the optimal path from the given set of information. Various techniques are used for the processing of path queries .Some of the techniques use travelling constraints which are either total order or partial order. Optimal path query processing finds the entire possible path and then refines the best path. Different queries use different algorithms. In this paper, we discuss about various techniques used in path queries. Queries on road network are different types. Some consider single point while others consider multiple points. In the case of trip planning multiple points are considered. Road network mainly deals with time dependent parameters and it can be modeled by considering the parameters. Different models are used to represent this.

The paper is organized as follows: section 2 presents various path query algorithms. The two models to represent the road network are presented in section 3.

2. Path Query Algorithms

2.1 Fast Approximation Algorithm

On Trip Planning Queries are the efficient and exact solutions for the general optimal route queries. A set of query points are given, where each point belongs to a specific category, a starting point S and a destination E. TPQ retrieves best trip that starts at S and passes through at least one point from each category and ends at E[4]. Four algorithms with various approximation ratios are used in terms of m and ρ, *where* m is total number of categories and ρ is maximum category cardinality. Two greedy algorithms with tight approximation ratios with respect m are Nearest Neighbor Algorithm and Minimum Distance Algorithm.

Nearest Neighbor Algorithm Iteratively visit the nearest neighbor of last vertex added to the trip from all vertices in categories that have not been visited yet. Minimum Distance Algorithm is better when compared with Nearest Neighbor Algorithm. Algorithm chooses set of vertices, one vertex per category and sum of cost per vertex is minimum cost among all vertices belong to respective category. Creates trip by traversing these vertices in nearest neighbor order.

2.2 New Dijkstra Based Algorithm

A new DIJKSTRA-based algorithm is used to find the optimal LTT with time complexity $O((n \log n + m)\alpha(T))$ and space complexity $O((n + m)\alpha(T))$, where n is the number of nodes, m is the number of edges, and $\alpha(T)$ is the cost required for each function operation. Propose algorithm by decoupling path selection and time refinement. DIJKSTRA-based algorithm is used for time-refinement and a linear-time algorithm for path selection. Dijkstra Based time refinement: time-refinement means to compute and refine the earliest arrival-time function $g_i(t)$ for every node v_i in V The earliest arrival time function is calculated for every node. Then refine the arrival time function, incrementally in the given starting time interval. Incrementally means refine the earliest arrival time function by extending the starting sub-interval to larger starting sub interval.

2.3 P*-A Best First Search Algorithm

P* algorithm, a best-first search algorithm for efficient probabilistic path query evaluation. P* carries the similar spirit as the A* algorithm. It iteratively visits the next vertex that is most likely to be an answer path using a heuristic evaluation function, and stops when the rest unexplored paths have no potential to satisfy the query. However, the two algorithms are critically different due to the different types of graphs and queries. A* is used to find the shortest path between two vertices u and v in a certain graph. Therefore, the heuristic evaluation function for each vertex v_i is simply the sum of the actual distance between *u* and *vi* and the estimated distance between *vi* and *v*. P* aims to find the paths between two vertices *u* and *v* that satisfy the weight threshold *l* and probability threshold *p* in a probabilistic graph with complex correlations among edge weights. Therefore, the heuristic evaluation functions for each vertex *vi* is the joint distribution on a set of correlated random variables. This posts serious challenges in designing heuristic evaluation functions and calculation. Three heuristic evaluation functions that can be used in the P* algorithm are Constant Estimate, Min-Value Estimate and Stochastic Estimate.

2.4 Priority First Search With Dominance Pruning

This is a modified version of shortest path algorithm. Here all the undominated paths are maintained .If one path to a node dominates other then stochastic consistency condition ensures that the second cannot be part of overall shortest path. A method called priority first search with dominance pruning, a variant of priority first search is used. Two data structures are maintained Priority Queue and Closed list. Priority Queue consists of path and path cost. Closed list associates nodes with undominated paths found to that node.

The general procedure is:
Step1: Algorithm first add origin to priority queue with path cost 0.
Step2: Then get highest-priority item from PQ. If this item has lower expected utility than a known path to the destination, then terminate and return the best path to the destination found so far.
Step3: Then add item to closed list. If there is already another path to that node with dominating priority then, go to step 2. Otherwise, add the path and its cost to the closed-list associated with this node.
Step4: Generate successors to this item. Construct new paths for each possible bus we could take from this node, and put the resulting items on PQ. Go to step 2.

2.5 Route Traversal And Link Traversal Search With Transitions

Route Traversal Search traverses nodes similar to DFS.When expanding the current search node, RTS consider all successor nodes for each route that includes this node. It employs a termination check, based on the reachability information within the routes. The principle depends on inverted file R-index on the route collection. Route Traversal Search with Transitions exploits information about the transitions among routes stored in T-index. It employs a stronger termination check based on the transitions between routes. In Link Traversal Search the search stops as soon as LTS visits a node (link) that lies on the same route with the target. Algorithm employs an augmented inverted file on the route collections, termed R-Index$^{+,}$ which associates a node with the routes that contain it and the immediately following link. Link Traversal Search with Transitions enforces a stronger termination check than LTS using the transition graph of the route collection. Search finishes when it reaches a node that is closer than two routes away from the target. It uses information from the T –Index.

2.6 Backward Search And Forward Search Solution

The backward search methodology computes the optimal routes in reverse order of its points. Two algorithms are developed based on BSS are Simple Backward Search (SBS) and Batch Backward Search [6]

SBS computes an upper bound of the optimal route length, using a greedy algorithm .Then, SBS retrieves the set CS of candidate points that may be part of the optimal route which are those that 1) belong to any category contained in the visit order graph , 2) fall within distance to the query start point q. This can be performed efficiently, e.g., by executing a circular range query on each R-tree that indexes a category of points relevant to the query.

The batch backward search (BBS) method, improves SBS by employing batch processing in the backward join operations. Specifically, both the candidate set CS and the route set is partitioned into clusters before participating in a backward join. The partitioning of CS first groups points by their category, and then for each group, the points are further partitioned into clusters based on their spatial proximity. The partitioning of route set follows a similar strategy, by first grouping routes based on the categories they cover, and then clustering each group according to the locations of their start points. The clustering module in BBS must be highly efficient, since it is called during query time.

The forward search approach traverses the search space in a depth-first manner, and incrementally improves the bound for optimal route length. As an additional benefit, forward search methods report results progressively, i.e., they first quickly produce one solution to the query, and then incrementally update it, until reaching the optimal one or being terminated by the user. Two algorithms developed based on FSS are Simple forward Search (SFS) and Batch Forward Search (BFS)

The simple forward search (SFS) method resembles Greedy in that it also extends the current path by adding the nearest point from an unvisited category. A major difference between the two is that SFS backtracks after it obtains a complete route. BFS follows the same depth-first search paradigm as SFS. However, instead of enumerating individual routes, BFS searches for sequences of clusters, which we call cluster paths.

Specifically, in a preprocessing step, BFS partitions the candidate set into clusters as in BBS, i.e., the points in each cluster belong to the same category, and are close to each other in space.

3. Models Used to Represent Road Network

3.1 Multi-Cost, Time-Dependent and Uncertain Graph (MTUG) Model

Construction of a multi-cost, time-dependent and uncertain graph (MTUG) model [8] of a road network is capable of capturing multiple time-varying and uncertain travel costs.

Specifically, each cost on a road segment is modeled as a vector of (*interval, random variable*) pairs. The techniques build an MTUG from a massive collection of GPS data collected from vehicles traveling in the road network.Multi-cost, Time-dependent, Uncertain Graph (MTUG) G=(V,E,MM,W) is a directed weighted graph. V and E are vertex and edge set, MM is a vector of functions, maintain minimum and maximum values of cost type for all edges is a vector of weight functions. Based on the MTUG, a cost of a route is defined, a *dominance* relationship between routes based on their costs, and a natural notion of an *optimal route* for a given source destination pair and a trip starting time. An optimal route is a "good" route with the property that no other route is better when considering all travel costs of interest.

A pruning strategy and an efficient method for checking stochastic dominance support efficient optimal route query processing. For query processing partially explored route and complete route are considered. Estimate the best possible travel costs to the destination for partially explored routes. If a partially explored route with its estimated travel cost is dominated by a complete route, the partially explored route can be disregarded.

Stochastic Dominance Checking is the most time consuming part of optimal route planning algorithm. To check whether route R1 dominates route R2, check the dominance relationship between the cumulative distribution functions of the routes for each travel cost random variable of interest. Three cases are there disjoint case, covered case and overlapping case.

3.2 Time Expanded Network(TEN)

Many tour planning problem focus on solving time dependent parameters. It is difficult to represent the time-dependent model using general static network models. Time-Expanded Network (TEN) is introduced to represent such model. It obtains the optimal tour route with appropriate departure place and time schedule. TEN deal with time-dependent parameters. It contains a copy to the set of nodes in the underlying static network for each discrete time step. The TEN-based methodology involves i) Identifying and quantifying switching values of parameters.ii) Creating an optimization model based on the concept of TEN. iii) Solving the model under probabilistic or manually generated scenarios for parameters to identify optimal designs.

TEN constructions consist of mainly three steps.

1) Expansion of static Network, that is make a copy of original static network for every time step from time zero to time horizon T.
2) Connecting nodes with direct edges according to corresponding time durations.
3) Set satisfaction values and cost with respect to each directed edge for every time step.

The disadvantage is that it is computationally difficult to solve large scale tour planning problem. TEN models formulate as network optimization model and include many constraints.

4. Conclusion

Path queries are used to find the optimal path in the road network. Various queries have been proposed. All methods give an optimal path. Some consider partial order constrains while others consider total order constraint. This paper made a detailed survey about various path query methods used in road networks and also discussed about two different models to represent the road network.

References

[1] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[2] M. Hua and J. Pei, "Probabilistic path queries in road networks: Traffic uncertainty aware path selection," in *Proc. 13th EDBT*, Lausanne, Switzerland, 2010, pp. 347–358.

[3] R. Jin, L. Liu, B. Ding, and H. Wang, "Distance-constraint reachability computation in uncertain graphs," in *Proc. VLDB*, Jun. 2011,pp. 551–562.

[4] F. Li, D. Cheng, M. Hadjieleftheriou, G. Kollios, and S.-H. Teng, "On trip planning queries in spatial databases," in *Proc. 9th Int.Conf. SSTD*, Angra dos Reis, Brazil, 2005.

[5] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios, "*K*-nearest neighbors in uncertain graphs," in *Proc. VLDB*, Singapore, Sep.2010.

[6] Li J, Yang Y.D, Mamoulis N(2013), "Optimal Route Queries with Arbitrary Order Constraints", IEEE Trans. Computers, vol. 25, no. 5, pp. 1097- 1110

[7] Li F, Cheng D, Hadjieleftheriou M, Kollios G, Teng S.H(2005), "On Trip Planning Queries in Spatial Databases", Proc. Ninth Int'l Conf. Advances in Spatial and Temporal Databases (SSTD)

[8] Bin Yang, Chenjuan Guo, Christian S. Jensen, Manohar Kaul, Shuo Shang," Multi-Cost Optimal Route Planning Under Time-Varying Uncertainty"