IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

Shortest Path Algorithm for Spatial Query Processing in Road Network

Rambha Agrahari[1], Rakesh Kumar Singh[2], Toshika Dutta[3]

M.Tech Scholar, CSE, Kamla Nehru Institute of Technology, Sultanpur, U.P [1]

Associate Professor, CSE, Kamla Nehru Institute of Technology, Sultanpur, U.P[2]

M.Tech Scholar, CSE, Kamla Nehru Institute of Technology, Sultanpur, U.P [3]

Abstract: If any algorithm has n-shortest path then it has some problem like path similarity among the network expansion and determined path describing turn prohibition. GPS system is an application in the road network field. the meaning of path similarity that is alternative paths derived from k-shortest path algorithm which is connected to lots of links, hence heterogeneity cannot be represented .In road network computing optimal path routes is not easy it's like a showpiece of real world application algorithm. For computing the optimal route in real road network we can use classic solution from the graph theory which is describe by Dijkstra's algorithm. But for the large network this is not suitable algorithm. This works too slow for the large road network. This paper proposed link based shortest path algorithm which find out the different path for travelling for node to node network where turn prohibition exist. In this process there is no need of extra node and edge for expanding the network. Dijkstra's algorithm tested with some example networks then will be expanded to dynamic case**.**

Keywords: n-shortest path, turn prohibition, wireless network, road network, Routing algorithm, kNN query.

I. INTRODUCTION

In road network computing the shortest path between two places is an important problem that finds application in various map services and commercial navigation products. The solution for this problem is Dijkstra's algorithm [1], which, given a source s and a destination d in a road network N[2, 3-7, 8-10], traverses the vertices in N in ascending order of their distances to s. When d is completed the traversal then the shortest path computed and returned. This algorithm is easy and successful, but it is often inefficient for more number of road networks. In Dijkstra's algorithm the process is start from source to destination. In this algorithm first source node S is start the visiting each node (all vertices) which closer to the path from source S to destination d. The number of those vertices will be considering which is defining the shortest path from source to destination. The Dijkstra's algorithm can be easily represented by the graph. For solving this problem we can distinguish between several variants of this problem:

- Point-to-point: compute the shortest-path length from a given source node s to destination node d.
- Single-source: for a given source node is s.
- many-to-many
- All-pairs: a special case of the many-to-many variant with s: =d: =N.

This paper intend to do something a link-based shortest path algorithms for the travel facts in real road network where be found sprain prohibitions. When penalized turns are dealt with without explicitly more extensive the network, node-based existing shortest path algorithm is not suitable, because the Bellman's optimality condition has the property that no node may be approached from the points where something begins for less cost than the chosen preceding node which is also reached by a shortest cost path. But link-based shortest path algorithm makes possible to throw back without absorbing it all turns effectively because it holds Bellman's optimal condition in searching step.

II. PROBLEMS

Let's consider a road network G (i.e. degree restricted connected graph) with an edge set E and vertex set V of n-vertices. Let each edge e $\in$ E be make conceptual connection with a weight w(e),which we accept as true without proof (unless loss of the majority) to be the length of e. For ease of archaic public exposure, we think carefully about undirected graph in this paper.

We study two types of queries on graph i.e. to say shortest path queries and distance queries. Specified two vertices s, d $\in$ V, a shortest path query expect for a sequence of edges (e_1, e_2,…….. e_n) that connect source (s) to destination (d),extent that $\sum_{i=1}^{n}(w(ei))$ is minimized. On the other side, a distance query between s to d formily ask to do for something only the value $\sum_{i=1}^{n}w(ei)$,extent that e_1, e_2, e_3,…… $\in$ E constitute the shortest path from s to d [2,11-13]. Distance query is useful in the setting where the distance between two positions in place of the shortest path is the major effect. For example take that a person has a list of his favorite shopping malls and he wants to identify the shopping mall i.e. closest to his working place P.

In that case he may issue a distance query form P to each of the shopping mall to find the nearest one. For freedom from difficulty, we use dist (v1, v2) to indicate

DOI 10.17148/IJARCCE.2015.41264

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

the length of the shortest path between two vertices v1, v2 ∈ V.

III. ALGORITHMS

This category has five techniques, assessed in our experiments, that is to say:

i. The bidirectional Dijkstra's algorithm [1], a different version of Dijkstra's algorithm that we used as the starting point.
ii. CH [9] and TNR [4], two state-of-art vertex-importance-based methods.
iii. SILC [2, 11] and PCPD [13], two most advance spatial coherence based algorithm.

A. The bidirectional Dijkstra's algorithm.

Given two vertices source vertex s and destination vertex d belongs to V i.e. s, d ∈ V, the bidirectional Dijkstra's algorithm, incantations two particular cases of Dijkstra's algorithm occurring at the same time , extent that the first (respective the second) particular cases extent across the vertices in graph G in go up order of their distance to s (respect to d).And it also maintains the minimum spanning tree from source node S to destination node d. When the traversal is performed in Dijkstra's algorithm it is started from source node s. In first traversal if source node is connected to the different node (n1,n2) then we check which is nearest to source node then we take that node and the further Dijkstra's algorithm apply on the second node and so on.

Complexity of bidirectional Dijkstra's algorithm is O(nlogn) time ,but it is more efficient than Dijkstra's algorithm in practice. Because, the ability to understand something immediately, in two graph each graph traversal incantation by the bidirectional algorithm visit that vertices which are within roughly dist (s,d)/2 distance to s or d. The number of so great vertices is frequently smaller than the number of vertices which are within dist (s,d) distance to s, i.e. these vertices need to be traversed by Dijkstra's algorithm.

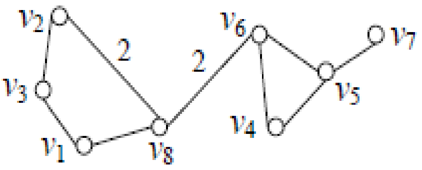

Figure 1: Road Network

B. Contraction Hierarchies

Contraction Hierarchies CH [9] is a graph indexing techniques i.e. network partition index techniques that complied with a total on the vertices in graph G in proportion to their relative importance.

It pre-calculates the distance in the middle of different properties vertices based on the total order, and it utilizes as pre-calculated distance to begin to move more quickly shortest path and distance queries. Processing step of CH is explained below:

Let us think carefully the road network G in fig.1. Which contains eight vertices $v_1, v_2, v_3,$ ………….. v_8 and nine edges . In individual the edge lengths of (v2, v8) and (v6, v8) equal 2, while the other edge length are 1.Without the loss of the quality, take that CH complied with a total order $v_1 < v_2 < v_3, <$…………..$< v_8$ on the vertices in graph G. In CH is pre-processing step test the knowledge the vertices following the total order. If three vertices we take v_i , v_j and v_k, first we check neighbor of v_i. If vi has two vertices v_j and v_k then we check the shortest path from v_j to v_k passes through vi. In the process of CH it inserts in graph G an artificial edge c which is called shortcut which are connected to v_j to v_k, extent that

$$W(c) = dist\ (v_j, v_k).$$

Define that with the shortcut added, v_j becomes a neighbor of v_k and v_k becomes a neighbor of v_j. Once all neighbor of vi are inspected, vi is removed from graph G. This process is called the Contraction of v_i [9].After all vertices are contracted processing step are terminated by CH, and the shortcuts that have been created in duration of contraction process are added to the earliest road network.

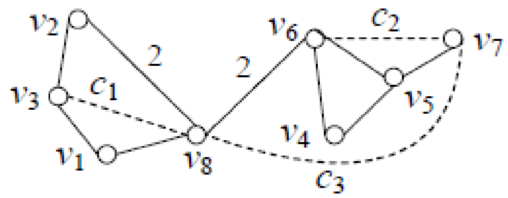

Figure 2: Contraction Hierarchies

The inferior ordering can load to $O(n^2)$ shortcuts and the complexity of CH is $O(n^2 logn)$.

C. Transit Node Routing

It is an indexing method which complied with a grid on road network. TNR [4] pre-calculate shortest path from each grid cell c. We define the TNR using the example in fig.3 that shows a grid complied with on the road network in fig.1. In the grid for each cell c, let us define the inner shell (respect to outer shell) of c as the boundary of 5×5 (resp. 9×9) square. TNR also calculates the pairwise distances in the middle of the access nodes of grid cell c_1 and grid cell c_2 i.e. dist (v_3, v_5) and dist (v8,v5).In general, two cells are given c_s and c_d extent that they are not contained each other outer shells, the distance between any verticex s in c_s and any vertex in c_d can be calculated as:

$$dist\ (s,d) = \min_{v_s \in As, vd \in Ad} dist\ (s,v_s) + dist\ (v_s, v_t) + dist\ (v_t, t) \ \ldots\ldots (1)$$

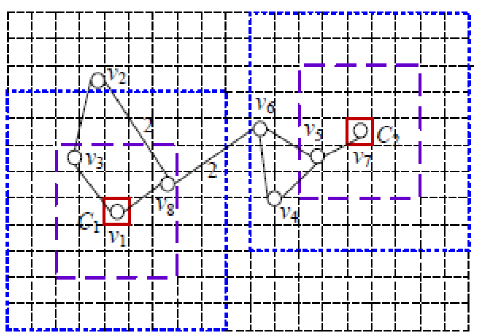

Figure 3: Transit Node Routing [4]

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

D. Spatially Induced Linkage Cognizance

SILC [2, 11] is a technique that defines the different given properties-

i. All-pairs shortest in the road network are pre-calculated.
ii. For efficient query processing it stores the shortest paths in a concise form. Example of road network is define in fig.1 in graph G.

In general, for each vertices v ∈ V the equivalent partition of V/{v} can be represented using O(√n) disjoint square[].The space complexity of SILC is O(n√n).For square region it can be done in O(log n) time[].Hence complexity of SILC for defining the shortest path algorithm is O(k log n) time.

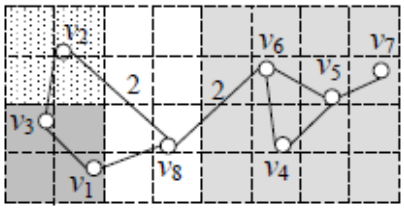

Figure.4: SILC [2, 11]

E. Path Coherent pairs decomposition

Path coherent pair decomposition is similar to SILC. A road network PCPD [13] pre-calculates a set s_{pcp} of path coherent pairs, extent that any two verticesv_1, v_2 ∈ V are covered by a unique path-coherent pair (x, y, ∅) ∈ s_{pcp} .It defines the shortest path from v_s to v_d in path-coherent.

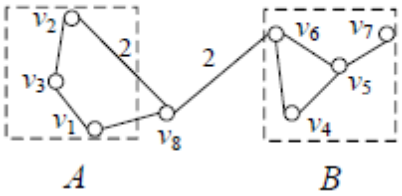

Figure 5: PCPD [13]

IV. OTHER IMPLEMENTATIONS OF DIJKSTRA'S ALGORITHM

Dijkstra's labeling method is a most important procedure in shortest path algorithms. In the labeling method the output is an out-tree from a source node s, to a set of node L. An out tree is a type of tree which is originating from source node to the other nodes which defines the shortest distance from the source node is known. This out tree is erected iteratively, ant the shortest path from source node s to destination node d is obtained termination of the technique. In labeling method each node i required three piece of information for constructing the shortest path tree

- The distance label d(i)
- The parent node P(i)
- The set of permanently labeled nodes L.

V. EVALUATION

In this paper, we represent covering a large area experiments to assess the competence of NPI[21] (Network Partition Index) for corroborate range queries, kNN [20] queries , and CNN[17,19] queries processing techniques are put into effect as a competitors, containing as part of whole being considered RNE and INE algorithm[14], NVD algorithm[15],and NGE algorithm[16]. All the algorithms which is used in this paper, all are implemented in C++, and competence evolution is simulated on a genuine Intel (R) 1.8 GHz PC with 3.00 GB RAM, running Microsoft windows 7 Ultimate. We first concise describe the experimental settings, and then present the experimental result.

Our evaluation look attentively at Dijkstra's algorithm, ArcFlag, Turning point, EB and NR. HiTi and SPQ are removed from consideration as their space a thing that is needed surpass our device's heap size even for the smallest of our networks, for further indications about their impracticality see Table 1 and discussion thereof.

TABLE 1: BROADCAST CYCLE LENGTH [18]

Method	Packets	Sec. (2Mbps)	Sec. (348Kbps)
Dijkstra (DJ)	14019	6.845	40.284
NR	14260	6.963	40.977
EB	15299	7.470	43.962
Landmark (LD)	21236	10.369	61.022
ArcFlag (AF)	29233	14.274	84.003
SPQ	52337	25.555	150.391
HiTi	58138	28.387	167.061

VI. EXPERIMENTAL SETUP

In this paper for the simulation we use two real road network datasets. There are two country can be taken first is Ukraine (UR) and second is Australia (AS). City of Ukraine road network contains 7105 nodes and 8324 edges, while city of Australia road network contains 20,034 nodes and 21,873 edges. A set of objects are randomly generated for each road network and uniformly distributed over the network. After this our frame work can handle objects of different sizes, for the sake of simplicity, we fix the size of an object at 64 bytes.

The evolution is run on a simulator which can be composed of a server, a client and a broadcast channel. The server pre calculates the distance bound matrix and the diameter of each cell. The pre-calculation information together with the road network data is broadcast on the channel repeatedly. For simplicity we run only one client in our simulation, because the performance of the broadcast system will not affect the number of clients.

If we take 300 randomly issued queries are evaluated for set of experiments and then average competence is reported. As previously define, we assume the query issuing points are always at the network nodes although our algorithm can be extended without effort to support cases where queries are issued along the edges. We assume the main performance is both the access latency and tuning time. We assume measure AT and TT in terms of number of bytes of data transferred and band-width is fixed in the wireless channel in place of actual clock time in the client side.

In this paper, we mainly consider the influence caused by four parameters. These parameters are

The number of k asked by kNN queries, the distance d of range queries, the object density, and the number of grid cells N.

Table.2 consists their values with the underlined values standing for the default settings. Define that D_N refers to the diameter of the road networks, object set s define as |S| and the number of nodes of road network define by |V|. Without loss of majority, in each set of experiments we vary the value of one parameter while the other set at their defaults three parameters.

TABLE 2: PARAMETER SETTINGS [19]

Parameter	Settings
k	1, 5, 10, 15
Query scope (d/D_N)	0.01, 0.05, 0.1, 0.2
Object density (|S|/|V|)	0.01, 0.05, 0.1, 0.2
Number of cells (N)	16, 64, 256

VII. FUTURE SCOPES

In the concluding remarks of this paper, we have listed possible additional development of existing work-some of referred projects have already commenced. In individual presentation, we have started to firmly decide better highway node sets, to computing the pre-processing the highway-node routing for a mobile device.

VIII. CONCLUSION

This paper presents a based on untested ideas equivalent of four state-of the-art a procedure that is effective in achieving the aim for answering shortest path and distance queries on road networks, namely, SILC, PCPD, CH, and TNR. We used the quality of real datasets with up to twenty million vertices, and we assessed each technique in terms of its preprocessing time, space overhead, and query efficiency. From our experimental results, we have the following actions. First, CH is the most space economic technique compared with TNR, SILC and PCPD, and nevertheless, it is the second most efficient technique in answering shortest path and distance queries. This makes CH a suitable choice when both space efficiency and time efficiency are major effects. Second, TNR can be combined with CH to achieve significant speedup.

For distance queries, especially when the source and destination vertices are far away from each other. However, it also involves considerable space overhead, and it is not as efficient as CH for shortest path queries. Third, SILC incurs significant preprocessing time and space consumption, but it offers superior efficiency for shortest path queries. Therefore, SILC is desirable for processing shortest path queries when time efficiency is decisive and space overhead is less concerned. Finally, although PCPD was proposed as a successor to SILC with an improved asymptotic space complexity, its practical performance (in terms of preprocessing time, space consumption, and query efficiency) is inferior to SILC.

In this paper, we propose the first study on shortest path Mathematical calculation in this model. We first make suitable for new purpose existing shortest path an initial proposals to the broadcast setting, and recognize their deficiencies. Then, we propose two novel methods fitted to the technical peculiarities of the model and of real-world handheld devices. Finally, we empirically demonstrate the efficiency of our techniques using real road networks and actual device specifications. A promising direction for future work is to consider on-air processing of spatial queries in road networks, e.g., range and nearest neighbor retrieval.

REFERENCES

[1] E. W. Dijkstra, "A Note on Two Problems in Connection With Graphs",Numerical Mathematics, 1:269–271, 1959.
[2] H. Samet, J. Sankaranarayanan, and H. Alborzi, "Scalable Network Distance Browsing Insspatial Databases", In SIGMOD, pages 43–54, 2008.
[3] I. Abraham, A. Fiat, A. V. Goldberg, and R. F. F., "Werneck. Highway Dimension, Shortest Paths, and Provably Efficient Algorithms", In SODA, pages 782–793, 2010.
[4] H. Bast, S. Funke, and D. Matijevic, "Transit: Ultrafast Shortest-Path Queries with Linear-Time Preprocessing", In Proc. of the 9th DIMACS Implementation Challenge, pages 175–192, 2006. http://www.mpi-inf.mpg.de/ ˜dmatijev/papers/DIMACS06.pdf.
[5] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes, "In transit to Constant Time Shortest-Path Queries in Road Networks", In ALENEX, 2007.
[6] H. Bast, S. Funke, P. Sanders, and D. Schultes, "Fast Routing in Road Networks with Transit Nodes", Science, 316(5824):566, 2007.
[7] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering Route Planning Algorithms", In Algorithmics of Large and Complex Networks, pages 117–139, 2009.
[8] J. Fakcharoenphol and S. Rao. "Planar graphs, Negative Weight Edges, Shortest Paths, and Near Linear Time", J. Comput. Syst. Sci., 72(5):868–889, 2006.
[9] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction hierarchies: Faster and Simpler Hierarchical Routing in Road Networks", In WEA, pages 319–333, 2008.
[10] A. V. Goldberg and C. Harrelson, "Computing the Shortest Path: A∗ Search Meets Graph Theory", In SODA, pages 156–165, 2005.
[11] J. Sankaranarayanan, H. Alborzi, and H. Samet, "Efficient Query Processing on Spatial Networks", In GIS, pages 200–209, 2005.
[12] J. Sankaranarayanan and H. Samet, "Query Processing Using Distance Oracles for Spatial Networks", IEEE Trans. Knowl. Data Eng., 22(8):1158–1175, 2010.
[13] J. Sankaranarayanan, H. Samet, and H. Alborzi, "Path Oracles for Spatial Networks", PVLDB, 2:1210–1221, 2009.
[14] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao, "Query Processing in Spatial Network Databases", in Proc. 29th Int. Conf. Very Large Data Bases, 2003, pp. 802–813.
[15] M. Kolahdouzan and C. Shahabi, "Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases", in Proc. 30th Int. Conf. Very Large Data Bases, 2004, pp. 840–851.
[16] H. Kriegel, P. Kroger, P. Kunath, M. Renz, and T. Schmidt, "Proximity queries in large traffic networks", in Proc. 15th Annu. ACM Int. Symp. Adv. Geographic Inf. Syst., 2007, pp. 21–28.
[17] Weiwei Sun, Chunan Chen, Bhaihua Zeng, Chong chne, Peng Liu, "Network Partition Index",IEEE Transactions On Knowledge And Data Engineering, Vol.27, No.2 pp.393-394, feb 2015
[18] B. Zheng, W. Lee, and D. Lee, "Spatial Queries in Wireless Broadcast Systems," Wireless Network, vol. 10, no. 6, pp. 723–736, 2004.
[19] H. Cho and C. Chung, "An Efficient and Scalable Approach to CNN Queries in a Road Network", in Proc. 31st Int. Conf. Very Large Data Bases, 2005, pp. 865–876.
[20] Weiwei Sun, Chunan Chen, Bhaihua Zeng, Chong chne, Peng Liu,"kNN Query", IEEE transactions on knowledge and data engineering , Vol.27, No.2 pp.392-393, feb 2015.
[21] Weiwei Sun, Chunan Chen, Bhaihua Zeng, Chong chne, Peng Liu, "CNN Query", IEEE transactions on knowledge and data engineering , Vol.27, No.2 pp.384-386, feb 2015