

A Novel Approach for Finding Top-k Shortest Path with Diversity

Namrata Sayaji Patil¹, Rutuja Shinde¹, Mr. Viresh Vanarote²

B. E Student, Department of Computer Engineering, RMD College of Engineering, Pune, India¹

Asst. Professor, Department of Computer Engineering, RMD College of Engineering, Pune, India²

ABSTRACT: Finding shortest simple path in a directed graph is a fundamental problem in many engineering applications. The classical K-Shortest Paths (KSP) problem, which identifies the k shortest paths in a directed graph, plays an important role in many application domains, such as providing alternative paths for vehicle routing services. However, the returned k shortest paths may be highly similar, i.e., sharing significant amounts of edges, thus adversely affecting service qualities. In the previous paper, author formalizes the K Shortest Paths with Diversity (KSPD) problem that identifies top-k shortest paths such that the paths are dissimilar with each other and the total length of the paths is minimized. The core of the framework includes the use of two judiciously designed lower bounds, where one is dependent on and the other one is independent on the chosen path similarity metric, which effectively reduces the search space and significantly improves efficiency. In this paper we are using new algorithms to improve time complexity and efficiency.

KEYWORDS: k-shortest path, path diversity, path similarity, efficiency.

I. INTRODUCTION

The shortest-path distance between vertices in a network is a fundamental concept in graph theory and is widely applied in the AI and Web communities. Finding top K shortest paths (KSP) problem has been extensively studied as a generalization of the shortest path problem. Given a graph G with non-negative edge weights, a positive integer K , a source vertex s and a destination vertex t , KSP algorithm ranks the top-K shortest paths from s to t and lists them in increasing order of length. The KSP problem can be found in various real world applications including network routing, chip layout planning, wireless communications, traffic engineering and database applications. Several variants of KSP problems have been examined which can be classified into two classes including the unconstrained problem and the constrained problem. No constraints need to be considered on the path definition in the unconstrained problem, while some constraints should be satisfied in the constrained problem. For example, some constrained KSP problems require the shortest paths to be simple (loop less), i.e., duplicate vertex in a graph should be excluded in each of the shortest paths. The k shortest paths problem is to list the k paths connecting a given source-destination pair in the digraph with minimum total length. Our techniques also apply to the problem of listing all paths shorter than some given threshold length.

Due to the different purposes the travel of the passengers and its transport requirements are different. For example, for business travel, the passenger wants the time of travel as short as possible. Tourists prefer the cost of travel as minimum as possible. National City traffic advisory procedures can satisfy the passengers demand and provide two to three optimal decision rules according to the transport advisory for passengers. The procedures allow users to look up city information, choose the optimal decision rule and edit city information. At first, the users choose the starting station, the terminal station, the optimal decision rules and traffic tool. The output information is the most time-efficient or the cost-efficient decision, the cost of time and money. Users can also add the city information and edit the cost and the timetable of traffic tools.

Actually, the shortest path problems are closely related to our daily life. For example, we all have to travel within the campus when we attend different lectures, such as going from one building to other building to attend the lectures. In fact, we should wisely choose our path so we can travel with least amount of time in order to arrive on time, especially when the next lesson starts only 5 minutes later. This will be an example of shortest path problem and the weight will be the time cost. Certainly different routes will involve different buildings and pathways, which some are less time-consuming.

Motivation

To Get the Current Location of Searching user and Get Result from current Location and the idea of matching patterns is based on an observation that similar starting and destination nodes of two queries may result in similar shortest paths (known as the path coherence property).

II. RELATED WORK

Literature survey is the most important step in any kind of research. Before start developing we need to study the previous papers of our domain which we are working and on the basis of study we can predict or generate the drawback and start working with the reference of previous papers.

In this section, we briefly review the related work on Finding Shortest Path and their different techniques.

This paper considers the problem of finding a number of spatially dissimilar paths between an origin and a destination. A number of dissimilar paths can be useful in solving capacitated flow problems or in selecting routes for hazardous materials. A detailed analysis of three existing methods to generate a set of dissimilar paths is provided and computational experience with the methods is reported. It is concluded that each of these three methods has a number of drawbacks.[1]

The major approach to approximate distances is the landmark-based approach which pre-compute and store a number of shortest path trees rooted at landmarks. While these methods easily attain preferable scalability, some of them have critical precision problems for close pairs and the other methods with better precision have three orders of magnitude slower query time. Consequently, focus of the research community is shifting toward under mentioned exact methods, leading to recent large improvement on exact methods.[2]

In this paper the author propose an indexing scheme for top-k shortest path distance queries on graphs, which is useful in a wide range of important applications such as network aware searches and link prediction. We develop a new framework for top-k distance queries based on 2-hop cover and then present an efficient indexing algorithm based on the recently proposed pruned landmark labeling scheme. The drawback of this method is efficiency and robustness.[3]

In this paper author propose a sidetrack representation of path, with which a path can be stored in space. We then show how to categorize a candidate path as either partial or complete, and restrict the number of paths added to the queue. In addition, we provide an empirical equation that can predict the shortest path length, provided that a much smaller number of shortest paths have been found. The drawback of the proposed algorithm is the case where the number of paths that has to be stored may grow exponentially.[4]

In this paper we define a new problem, the aim of which is to find a set of k dissimilar solutions for a vehicle routing problem (VRP) on a single instance. This problem has several practical applications in the cash-in-transit sector and in the transportation of hazardous materials. A min-max mathematical formulation is proposed which requires a maximum similarity threshold between VRP solutions, and the number k of dissimilar VRP solutions that need to be generated.[5]

In this paper, the author uses different algorithms which are acceptable in terms of their overall performance in solving the shortest path problem. All of these algorithms produce only one solution. improved in finding the shortest path or distance between two places in a map that The computed time complexity for each of the algorithms show that these represents any types of networks. In addition, other artificial intelligence techniques such as fuzzy logic and neural networks can also be implemented in improving existing shortest path algorithms in order to make them more intelligent and more efficient.[6]

This paper presents a survey of different shortest-path algorithms used for different purposes. Wireless Sensor Networks have potential applications in environment monitoring, disaster warning systems, health care, defense reconnaissance, and surveillance systems. However, the main constraint of the WSNs is the limited power sources of the sensor nodes. Therefore, energy conservation of the sensor nodes is the most challenging issue for the long run operation of WSNs. So we need to go for shortest Path routing for conserving the battery power. So author proposes the Particle Swarm optimization algorithm to find the shortest path in a Wireless Sensor Network. We further would go for a performance evaluation in terms of energy consumption to compare our new algorithm with the other already developed algorithms of WSN.[7]

In this paper, we propose a system, namely, Path Planning by Caching (PPC), to answer a new path planning query in real time by efficiently caching and reusing historical queried-paths. Unlike the conventional cache based path planning systems, where a queried-path in cache is used only when it matches perfectly with the new query, PPC leverages the partially matched queries to answer part(s) of the new query.[8]

In this paper, we propose an improved MPS algorithm for ranking top K shortest simple paths in large graph/networks. Our algorithm overcomes the problem of excessive amount of memory consumption in the original MPS algorithm by



two novel design strategies. One is to only add the shortest path to the pseudo-tree in each iteration through a tree-pruning scheme. The other is to adopt reversed order in the pseudo-tree such that the storing and searching shortest paths would be much more efficient.[9]

A literature review on this technique with focus on transportation network would be quite helpful for any research related to dynamicRoute Guidance systems (RGS). Route guidance helps us in providing the path directions based on changing traffic conditions. Given a set of origin-destination (O/D) pairs, there could be many possible routes for a driver. A useful routing system should have the capability to support the driver effectively in deciding on an optimum route to his preference. The algorithm is suitable for finding not only the shortest route but also better routes. The shortest travel time is estimated by applying various shortest path algorithms to the traffic network that has deterministic or dynamic link travel times. Because it is difficult to evaluate these shortest path-finding algorithms in real traffic situations, most of them are evaluated in the virtual traffic networks.[10]

III. PROBLEM STATEMENT

Finding the shortest path is the most prominent problem of our daily busy lives. Finding the shortest path/paths in a network is a fundamental problem or sub-problem of many practical applications. However, solving the shortest path problem is non-trivial as there may exist a huge number of paths between a source and a destination and thus it may take prohibitively long time to enumerate all such paths. Thus, the framework is also able to improve the efficiency of the classical K-Shortest Path problem when no path similarity metric is specified.

IV. PROPOSED METHOD

Our aim to find the top-k diversified paths with minimal total length. The proposed system is used to and out the shortest path between source location and destination location by using Shortest path estimation algorithm and cache replacement policy is used for cache management.

- The query is given to the system as an input. The query can be Place name, Location, Address.
- The query contains source location and destination location. User gives query to the system server.
- It detects the set best matching patterns which match with the new query.
- The set of Patterns is given to the shortest path estimation algorithm. It computes the best shortest path among them.

In this we are using 2 modules i.e. User and Admin.

Module 1 - Administrator (Admin):- Admin Add City details and check user Details.

Module 2 - User (Customer):- Customer can Search location with keyword and Get the Shortest Path using Shortest path Estimated Algorithms.

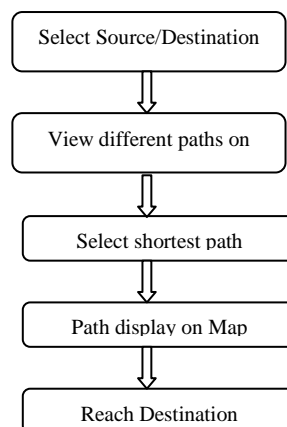


Fig.1 Flow diagram

V. ARCHITECTURE

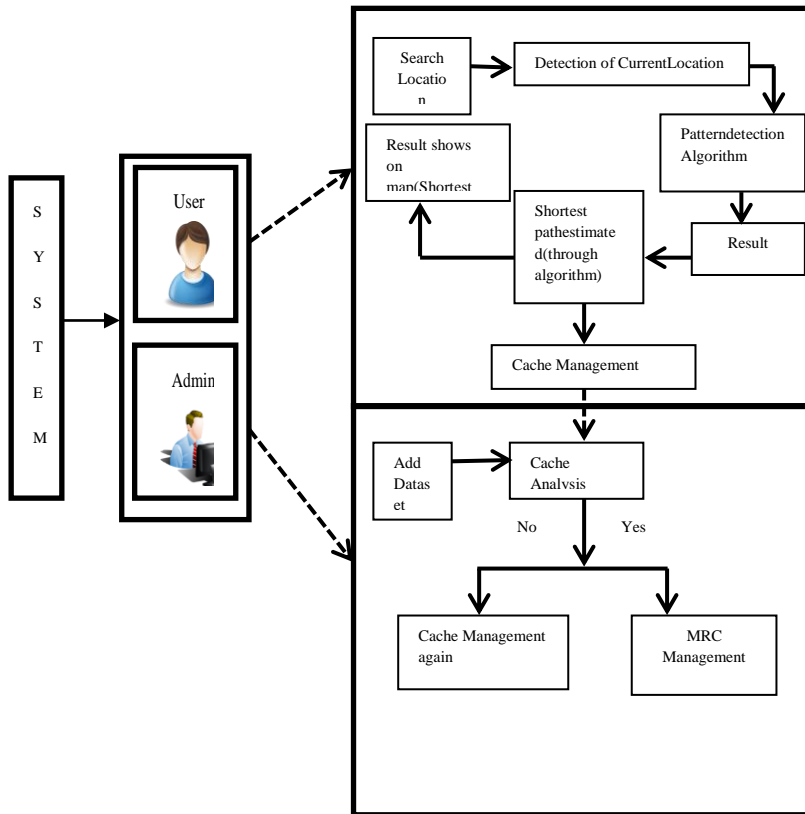


Fig.2 System Architecture

Algorithms:

A Cache algorithm is an algorithm used to manage a cache or group of data. When the cache is full, it decides which item should be deleted from the cache. The word hit rate describes how often a request can be served from the cache. The term latency describes for how long a cached item can be obtained. Cache algorithms are a trade-off between hit-rate and latency. The most efficient caching algorithm would be to always discard the information that will not be needed for the longest time in the future. A cache algorithm is a detailed list of instructions that directs which items should be discarded in a computing device's cache of information.

The most efficient caching algorithm would be to always discard the information that will not be needed for the longest time in the future. This optimal result is referred to as Cache algorithm. Since it is generally impossible to predict how far in the future information will be needed, this is generally not implementable in practice. The practical minimum can be calculated only after experimentation, and one can compare the effectiveness of the actually chosen cache algorithm.

1) Cache Construction and Update Algorithm

Input: a query, a cache C.

Output: Updated cache C with best shortest path.

Steps:

1. PT PPattern Detection
2. p Shortest Path Estimation from PT.
3. If C is Not full then 2
4. Insert p into C ; Return C
5. Else
6. Calculate usability for each cached path.
7. p* Path with the minimum usability
8. if p*. ; p. the
9. C Replace p* with
10. End



- 11. End
- 12. End

2) P Pattern Detection Algorithm

Input: qs;t:a query; DI:Distance Threshold ; C:a cache; Dg:grid cell size.

Output: All candidates PPatterns PT.

Steps

1. if $D(s; t) \leq DI$ then
2. Return $PT=0$
3. End If
4. Divided the Target space by Size Dg
5. Qs Logged queries whose paths pass gs .
6. Qt Logged queries whose paths pass gt .
7. $Q = \text{Intersect}(Qs;Qt)$.
8. PT Sub paths from Gs to Gt for each query in Q .
9. End
10. Return PT

Mathematical Model

Here , Denoting the n th node on the path p_i as v_n^i , Our System use $PT(p_i, p_j)$ to indicate whether path p_j is a k -P Pattern for path p_i in the Above equation.

Based on the above definition, we may construct the shortest distance path of a query q from a starting node s to a destination node t , denoted by $SDP(q)$ or $SDP(s, t)$, from one of its cached P Patterns $SDP(s, t)$ as follows:

Note that Eq. (1) is a general form of the constructed path, where $SDP(s, a)$ or $SDP(b, t)$ may not necessarily exist,

$$SPD(s',t') = SPD(s',a) \cup SPD(a,b) \cup SPD(b,t') \dots\dots\dots(1)$$

Note that Eq. (1) is a general form of the constructed path, where $SDP(s, a)$ or $SDP(b, t)$ may not necessarily exist, e.g., $s = a$ or $b = t$. The unshared segments $SDP(s, a)$ and $SDP(b,t)$ are to be computed by the server. The length of the path can thus be recomputed as follows

$$| SPD(s',t') | = | SPD(s',a) | + | SPD(a,b) | + | SPD(b,t') | \dots\dots\dots(2)$$

VI. RESULTS

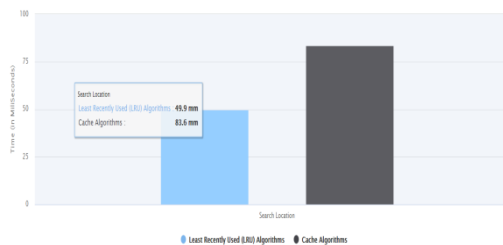


Fig.3 Comparison algorithm of cache management

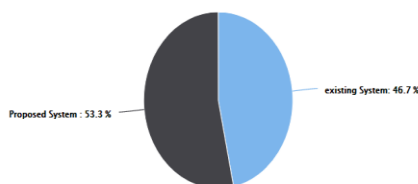


Fig.4 Compare Existing & Proposed system

VII. CONCLUSION

The shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized. In other words, when we have to find a path with minimum cost to go from a place to another place which there are a number of intermediate points in between to travel to with different costs, we are dealing with the shortest path problems. It should be noted that the phrase “shortest path” here does not necessarily mean physically shortest distance, but a path with minimum weight which can be measured in, say, time or monetary cost.

REFERENCES

1. V. Akgun, E. Erkut, and R. Batta, “On finding dissimilar paths”, *European Journal of Operational Research*, 121(2):232–246, 2000.
2. Y. Lim and S. Rhee, “An efficient dissimilar path searching method for evacuation routing”, *Ksce Journal of Civil Engineering*, 14(1):61–67, 2010.
3. T. Akiba, T. Hayashi, N. Nori, Y. Iwata, and Y. Yoshid, “Efficient top-k shortest-path distance queries on large networks by pruned landmark labeling”, In *Proc. AAAI*, pages 2–8, 2015.
4. Gang Feng, “Improving space efficiency with path length prediction for finding shortest simple paths”, *IEEE Transactions On Computers*, Vol. 63, No. 10, October 2014.
5. L. Talarico, K. Sørensen, and J. Springael, “The k-dissimilar vehicle routing problem”, *European Journal of Operational Research*, 244(1):129–140, 2015.
6. Thorat Surekha, Rahane Santosh, “Review of Shortest Path Algorithm”, *International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 08 Aug -2016*.
7. Abinash Mishra , Madhumita Panda, “A Survey of Shortest-Path Algorithms”, *International Journal of Applied Engineering & Research Volume 13*, 2018.
8. Ying Zhang, Yu-Ling Hsueh, Wang-Chien Lee, Yi-Hao Jhang, “Efficient Cache-Supported Path Planning on Roads”, 2017 IEEE 33rd International Conference on Data Engineering (ICDE).
9. Qingsong Wen, Ren Chen, Lifeng Nai, Li Zhou, Yinglong Xia, “Finding Top K Shortest Simple Paths with Improved Space Efficiency”, *Conference Paper May 2017*.
10. Ashok Kuppasamy , K.Yuvaraj, “A Literature review on finding the K Shortest Path using Dynamic Route Guidance Systems”, *International Journal of Advanced Research in Computer Science Volume 7*, November 2016.
11. L.Zammit, M. Attard, and K. Scerri, “Bayesian Hierarchical Modelling of Traffic Flow - With Application to Malta’s Road Network, ” in *International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 1376-1381.
12. H. Mahmud, A. M. Amin, M. E. Ali, and T. Hashem, Execution of Path Queries on Road Networks,” *Clinical Orthopaedics and Related Research*, vol. abs/1210.6746, 2012.
13. H. Mahmud, A. M. Amin, M. E. Ali, and T. Hashem, Execution of Path Queries on Road Networks,” *Clinical Orthopaedics and Related Research*, vol. abs/1210.6746, 2012.
14. R. Gutman, -Based Routing: A New Approach to Shortest Path Algorithms Optimized for Road Networks, in *Workshop on AlgorithmEngineering and Experiments*, 2004.
15. S. Jung and S. Pramanik, Efficient Path Computation Model for Hierarchically Structured Topographical Road Maps,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1029-1046, 2002.